Zaragoza - Del 8 al 10 de Noviembre de 2006

IV Jornadas en Tecnología del Habla

IV Jornadas en

Tecnología del Habla

Zaragoza • Del 8 al 10 de Noviembre de 2006

universidad de zaragoza

grupo de tecnologías de las comunicaciones

instituto de investigación en ingeniería de Aragón

Universidad de Zaragoza

IV Jornadas en Tecnología del Habla

Libro de actas de las
IV Jornadas en Tecnología del Habla

jth2006.unizar.es

Depósito Legal: Z-3138-2006
ISBN: 84-96214-82-6
Editado por Luis Buera, Eduardo Lleida, Antonio Miguel y Alfonso Ortega.

© Los derechos sobre cada uno de los trabajos pertenecen a los autores.

Organizan:

Red Temática en Tecnologías del Habla

instituto de investigación en ingeniería de Aragón

Universidad de Zaragoza

Patrocinan:

MINISTERIO DE EDUCACION Y CIENCIA

GOBIERNO DE ARAGON

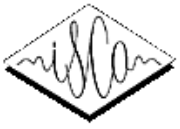
Departamento de Ciencia, Tecnología y Universidad

Telefónica

Cátedra Univ. Zaragoza

Apoyan y Colaboran:

AYUNTAMIENTO DE ZARAGOZA



CPS

Comité Organizador

Eduardo Lleida Solano *(Presidente)*

Alfonso Ortega Giménez

Enrique Masgrau Gómez

Luis Buera Rodríguez

Óscar Saz Torralba

Antonio Miguel Artiaga

Juan Diego Rosas Rodríguez

Jesús Villalba López

Luis Vicente Borruel

Comité Científico

José Miguel Benedí Ruiz, Universidad Politécnica de Valencia

Antonio Bonafonte Cávez, Universidad Politécnica de Cataluña

Francisco Casacuberta Nolla, Universidad Politécnica de Valencia

María José Castro Bleda, Universidad Politécnica de Valencia

Javier Ferreiros López, Universidad Politécnica de Madrid

Carmen García Mateo, Universidad de Vigo

Pedro Gómez Vilda, Universidad Politécnica de Madrid

Francisco Javier Hernando Pericás, Universidad Politécnica de Cataluña

Eduardo Lleida Solano, Universidad de Zaragoza

Joaquim Llisterri Boix, Universidad Autonoma de Barcelona

Teresa López Soto, Universidad de Sevilla

Ramón López-Cózar Delgado, Universidad de Granada

José B. Mariño Acebal, Universidad Politécnica de Cataluña

Climent Nadeu Camprubi, Universidad Politécnica de Cataluña

Eva Navas, Universidad del Pais Vasco

Alfonso Ortega Giménez, Universidad de Zaragoza

José M. Pardo Muñoz, Universidad Politécnica de Madrid

Antonio J. Rubio Ayuso, Universidad de Granada

Rubén San-Segundo Hernández, Universidad Politécnica de Madrid

Emilio Sanchís Arnal, Universidad Politécnica de Valencia

Encarna Segarra Soriano, Universidad Politécnica de Valencia

José Carlos Segura Luna, Universidad de Granada

Inés Torres Barañano, Universidad del País Vasco

Bienvenida del Comité Organizador

En el nombre del Comité Organizador, es un placer darles la bienvenida a la cuarta edición de las Jornadas en Tecnología del Habla promovidas por la Red Temática en Tecnologías del Habla.

Esta cuarta edición ha sido organizada por el I3A (Instituto de Investigación en Ingeniería de Aragón) de la Universidad de Zaragoza y apoyadas por ISCA (International Speech Communication Association).

Esta serie de Jornadas se han convertido en un punto de encuentro de los científicos y técnicos dedicados a la investigación y aplicación de las Tecnologías del Habla. Las Jornadas brindan una oportunidad excepcional para la presentación y divulgación de trabajos procedentes de diferentes campos de actividad relacionados, así como para el acercamiento del colectivo de investigadores, cada vez más numeroso, dedicados a esta disciplina. Al mismo tiempo, las Jornadas sirven de estímulo para la presentación de trabajos realizados por jóvenes investigadores que comienzan su actividad en el campo de las Tecnologías del Habla.

En esta edición, y a fin de fomentar y estimular la participación de jóvenes investigadores, la Red Temática en Tecnologías del Habla ha patrocinado una serie de premios y propuestas de evaluación de sistemas. Por un lado se van a otorgar 6 premios a los mejores artículos cuyo primer firmante sea un estudiante de una Universidad o Centro de Investigación Espaï¿½l. Por otro lado, se ha organizado una propuesta de evaluación de sistemas en las areas de Reconocimiento automático del Habla frente a habla patológica, Sínstesis del habla, Segmentación e Identificador de hablantes y Traducción de texto a lengua de signos.

Además del programa de sesiones orales y posters, se han organizado cuatro conferencias plenarias invitadas, que esperamos sean de gran interés para todos los participantes. El miércoles 8 de Noviembre Alex Acero, Microsoft Research, nos hablará sobre "Grand Challenges in Speech Recognition". El jueves 9 de Noviembre el profesor John H.L. Hansen, Universidad de texas en Dallas, nos hablará sobre "Spoken Document Retrieval for a National Gallery of the Spoken Word" y Roberto Pieraccini, Tell-Eureka Corporation, sobre "Speech 3.0, Spoken Dialog Systems of the Third Generation". Finalmente, el última día de la conferencia el profesor Richard C. Rose, McGill University, hablará sobre "Where are the Speech Production Models in Our Speech Processing Systems?"

Quisiera expresar mi agradecimiento en primer lugar a todos y cada uno de los miembros del Comité Organizador por su entusiasta colaboración en la organización de este evento. Agradezco igualmente su cooperación al Comité Científico y a las diferentes entidades que con su apoyo y colaboración han contribuido a una mejor organización de las Jornadas.

Sólo me resta desear a todos los participantes una estancia fructífera y agradable en la acogedora ciudad de Zaragoza, a lo que esperamos contribuya el conjunto de actividades de carácter social organizadas como complemento al programa técnico.

Zaragoza, Noviembre de 2006
Eduardo Lleida Solano
Presidente del Comité Organizador JTH2006

Índice de Sesiones

MIÉRCOLES, 8 de Noviembre de 2006

08:45 – 09:00	**Bienvenida** **Salón de Actos - Edificio Ada Byron** *Eduardo Lleida Solano, Presidente del Comité Organizador*

Sesión Oral: Procesado y Análisis de la Señal de Voz
De 09:00 a 10:40 - Salón de Actos - Edificio Ada Byron

09:00 – 09:15	**Acoustic Event Detection and Classification in Smart-Room Environments: Evaluation of Chil Project Systems** *Andrey Temko, Robert Malkin, Christian Zieger, Dušan Macho, Climent Nadeu, Maurizio Omologo*
09:15 – 09:30	**Time-Dependent Cross-Probability Model for Feature Vector Normalization** *Luis Buera, Eduardo Lleida, Antonio Miguel, Alfonso Ortega, Óscar Saz*
09:30 – 09:45	**Detección de Pitch en Condiciones Adversas** *Ibon Saratxaga, Íker Luengo, Eva Navas, Inmaculada Hernáez, Jon Sánchez, Iñaki Sainz*
09:45 – 10:00	**Systems for robust speech activity detection and their results with the RT05 and RT06 evaluation tests** *Dušan Macho, Andrey Temko, Climent Nadeu*
10:00 – 10:15	**Detección de Actividad de Voz Robusta al Ruido Basada en Modelos Acústicos** *Ángel de la Torre, Javier Ramírez, Carmen Benítez, José C. Segura, Luz García, Antonio J. Rubio*
10:15 – 10:30	**Voice activity detection using a contextual information and multiple hypothesis testing** *Javier Ramírez, José C. Segura, J. M. Górriz, Ángel de la Torre, Luz García, Carmen Benítez*
10:40 – 11:00	**Pausa para Café**

11:00 – 12:00	**Conferencia Invitada** **Salón de Actos - Edificio Ada Byron** **Grand Challenges in Speech Recognition** *Alex Acero, Microsoft Research, Redmond, WA USA*

Sesión de Pósteres: Traducción Automática y Procesado del Lenguaje
De 12:00 a 13:30 - Seminarios, Planta 2 - Edificio Ada Byron

A Good Gesture: Exploring Nonverbal Communication for Robust SLDSs
Beatriz López, Álvaro Hernández, David Díaz, Doroteo Torre, Luis A. Hernández, Eduardo López

A Multiple-Gaussian Classifier for Language Identification Using Acoustic Information and PPRLM Scores
R. Córdoba, L. F. D'Haro, R. San-Segundo, J. Macías-Guarasa, F. Fernández, J. C. Plaza

Basque-Spanish-English Machine Translation Using Finite-State Transducers
Alicia Pérez, M. Inés Torres, Víctor Guijarrubia, Francisco Casacuberta

Chinese-Spanish Statistical Machine Translation Experiments
Rafael E. Banchs, Josep M. Crego, Patrick Lambert, José B. Mariño

Diferentes Aproximaciones a la Codificación del Vocabulario en Modelado de Lenguaje Conexionista
Salvador Tortajada, María José Castro

Estimación de Modelos de Traducción de Secuencias de Palabras a Partir de Corpus Muy Grandes Mediante Thot
Daniel Ortiz-Martínez, Ismael García-Varea, Francisco Casacuberta

	Interfaz para el sistema de ayuda a la traducción del PRHLT *María-Teresa González, Francisco Casacuberta*
	Inventario de frecuencias fonémicas y silábicas del castellano espontáneo y escrito *Antonio Moreno, Doroteo Torre, Natalia Curto, Raúl de la Torre*
	Language modeling for verbatim translation task *Maxim Khalilov, José A. R. Fonollosa*
	Modelos de lenguaje basados en categorías semánticas en un sistema de diálogo de habla espontánea en castellano *Raquel Justo, M. Inés Torres, Lluís F. Hurtado*
	Una evaluación exhaustiva de SisHiTra, un paradigma híbrido en Traducción Automática *Jesús González, A. Giménez, Jorge González, A. L. Lagarda, J. R. Navarro, L. Eliodoro, V. Fénix, P. Peris, Francisco Casacuberta*
	N-Best Reordering in Statistical Machine Translation *Germán Sanchís, Francisco Casacuberta*
	Optimización del proceso de generación de gramáticas para RAH basadas en reglas y aplicación en un sistema de diálogo *Zoraida Callejas, Ramón López-Cózar*
	Perspectivas de la traducción automática castellano-gallego mediante técnicas estadísticas y por transferencia *Gonzalo Iglesias, Leandro Rodríguez, Eduardo R. Banga, Francisco Campillo, Francisco Méndez*
	Prototipo de traducción de voz a lengua de signos española *E. Ibáñez, A. Huerta, R. San-Segundo, L. F. D'Haro, F. Fernández, R. Barra*
13:30 – 15:00	**Pausa para Comer**

Sesión Oral: Sistemas de Diálogo y de Recuperación de la Información
De 15:00 a 16:40 - Salón de Actos - Edificio Ada Byron

15:00 – 15:15	**Automatic Vs. Manual Topic Segmentation and Indexation in Broadcast News** *R. Amaral, H. Meinedo, D. Caseiro, I. Trancoso, J. Neto*
15:15 – 15:30	**Evaluación de un corpus de diálogos mediante un gestor de diálogo estocástico** *David Griol, Lluís F. Hurtado, Emilio Sanchís, Encarna Segarra*
15:30 – 15:45	**DS-UCAT: Sistema de Diálogo Multimodal y Multilingüe para un Entorno Educativo** *Ramón López-Cózar, Zoraida Callejas, Germán Montoro, Pablo Haya*
15:45 – 16:00	**Entrenamiento de Modelos de Lenguaje Conexionistas con Grandes Vocabularios** *Francisco Zamora, Salvador España, Jorge Gorbe, María José Castro*
16:00 – 16:15	**Incorporating semantic knowledge to the language model in a speech understanding system** *Sergio Grau, Encarna Segarra, Emilio Sanchís, Fernando García, Lluís F. Hurtado*
16:15 – 16:30	**Experiments in Speech Driven Information Retrieval for Spanish Language** *César González-Ferreras, Valentín Cardeñoso-Payo*
16:40 – 17:00	**Pausa para Café**
17:00 – 17:45	**Evaluación de Sistemas ALBAYZIN-06** **Salón de Actos - Edificio Ada Byron**
17:45 – 18:30	**Reunión de la Red Temática** **Salón de Actos - Edificio Ada Byron**
19:00 – 20:00	**Recepción:** **Excmo. Ayuntamiento de Zaragoza**

JUEVES, 9 de Noviembre de 2006

Sesión Oral: Conversión Texto-Voz
De 09:00 a 10:40 - Salón de Actos - Edificio Ada Byron

09:00 – 09:15	**Sistema de Síntesis Armónico/Estocástico en modo Pitch-Asíncrono aplicado a Conversión de Voz** *Daniel Erro, Asunción Moreno*
09:15 – 09:30	**Síntesis de habla emocional por selección de unidades** *Ignasi Esquerra*
09:30 – 09:45	**A Method for Avoiding F0 Discontinuities in a Concatenative Intonation Model** *Francisco Campillo, Taniya Mishra, Jan van Santen, Eduardo R. Banga*
09:45 – 10:00	**Letter-to-sound conversion for galician TTS systems** *Daniela Braga, Luis Coelho*
10:00 – 10:15	**Maximum entropy motivated grapheme-to-phoneme, stress and syllable boundary prediction for portuguese text-to-speech** *María João Barros, Christian Weiss*
10:15 – 10:30	**Modelado y estimación de la prosodia mediante razonamiento basado en casos** *Ignasi Iriondo, J. Claudi, Lluís Formiga, Xavier Gonzalvo, Francesc Alías, Pere Miralles*
10:40 – 11:00	**Pausa para Café**

	Conferencia Invitada
11:00 – 12:00	**Salón de Actos - Edificio Ada Byron**
	Spoken Document Retrieval for a National Gallery of the Spoken Word *John H.L. Hansen, University of Texas at Dallas, U.S.A.*

Sesión de Pósteres: Miscelanea
De 12:00 a 13:30 - Seminarios, Planta 2 - Edificio Ada Byron

A Proposal of Metrics for Detailed Evaluation in Pronunciation Modeling
R. Barra, J. Macías-Guarasa, F. Fernández, L. F. D'Haro, J. M. Montero, J. Ferreiros

Bilingual Speech Recognition in Two Phonetically Similar Languages
Vicente Alabau, Carlos Martínez-Hinarejos

Cuantificación de la Calidad de la Voz para su Evaluación Clínica por Medio del Análisis Acústico
Jesús B. Alonso, Miguel A. Ferrer, José de León, Carlos M. Travieso

Estimación de Matrices de Transformación Mediante Funciones de Coste Basadas en el Mínimo Error de Bayes y su Aplicación a GMM
J. L. Navarro-Mesa, F. D. Lorenzo-García, A. G. Ravelo-García, S. I. Martín-González, P. J. Quintana-Morales, E. Hernández-Pérez

Estimación de probabilidades a posteriori en SVMs multiclase para reconocimiento de habla continua
Rubén Solera, Fernando Pérez-Cruz, Fernando Díaz de María

Estudio de Heurísticas para la implementación de A* en CTH basados en selección de unidades
Lluís Formiga, Francesc Alías

First Experiments on an HMM Based Double Layer Framework for Automatic Continuous Speech Recognition
Albino Nogueiras, Marta Casar, José A. R. Fonollosa, Mónica Caballero

Implementación de un reconocedor distribuido de voz en tiempo real sobre IP
Juan A. Morales, Timo Bauman, José L. Pérez Córdoba, Antonio M. Peinado, Ángel M. Gómez

La Conversión de Texto en Habla Multidominio: principios y portabilidad
Francesc Alías, J. Claudí

	Ponderación de las probabilidades de observación de los HMM para verificación de locuciones en sistemas de RAH *Marta Casar, José A. R. Fonollosa*
	Rendimiento Perceptual y Reconocimiento con Codificadores VoIP sobre Redes de Paquetes *José L. Carmona, Antonio M. Peinado, José L. Pérez, Ángel M. Gómez, Victoria Sánchez*
	Spanish keyword spotting system based on filler models, pseudo n-gram language model and a confidence measure *Javier Tejedor, José Colás*
	Speaker recognition experiments on a bilingual database *Marcos Faundez-Zanuy, Antonio Satué-Villar*
	Time course of Intonation Processing in European Portuguese: a Gating paradigm approach *Isabel Falé, Isabel Hub Faria*
	Multimodal Person Tracking in a Smart-Room Environment *C. Segura, A. Abad, Climent Nadeu, Javier Herrando*
	Iterative speaker adaptation using MLLR *J. R. Navarro, Carlos Martínez-Hinarejos, A. L. Lagarda, Luis Rodríguez*
13:30 – 14:45	**Pausa para Comer**
	Conferencia Invitada
14:45 – 15:45	**Salón de Actos - Edificio Ada Byron**
	Speech 3.0 Spoken Dialog Systems of the Third Generation *Roberto Pieraccini, CTO, Tell-Eureka Corporation*
15:45 – 16:00	**Pausa para Café**

Sesión de Proyectos/Demos
De 16:00 a 17:30 - Seminarios, Salón de Actos - Edificio Ada Byron

16:00 – 16:10	**AVIVAVOZ: Tecnologías para la traducción de voz** *José B. Mariño*
16:10 – 16:20	**EDECÁN: sistEma de Diálogo multidominio con adaptación al contExto aCústico y de AplicaciónN** *Eduardo Lleida, Encarna Segarra, M. Inés Torres, Javier Macías-Guarasa*
16:20 – 16:30	**A Virtual Butler Controlled by Speech** *A. Uría, Alfonso Ortega, M. Inés Torres, Antonio Miguel, Víctor Guijarrubia, Luis Buera, J. Garmendia, Eduardo Lleida, O. Aizpuru, Amparo Varona, E. Alonso, Óscar Saz*
16:30 – 16:40	**Control de un Robot "RoboSapien" Mediante Voz** *Vicent Tamarit, Carlos Martínez-Hinarejos*
16:40 – 16:50	**GeoVAQA: A Voice Activated Geographical Question Answering System** *Jordi Luque, Daniel Ferrés, Javier Herrando, José B. Mariño, Horacio Rodríguez*
16:50 – 17:00	**Prototipo de Sistema de Automatización de Encuestas Telefónicas** *Alejandro H. Toselli, Elsa Cubel, Alberto Sanchís*
17:00 – 17:10	**Vocaliza: An Application for Computer-Aided Speech Therapy in Spanish Language** *Carlos Vaquero, Óscar Saz, Eduardo Lleida, José Manuel Marcos, César Canalís*
17:30 – 21:00	**Acto Social**
21:00 – 00:00	**Cena de Gala - Entrega de Premios**

VIERNES, 10 de Noviembre de 2006

Sesión Oral: Reconocimiento de Locutor
De 09:00 a 10:40 - Salón de Actos - Edificio Ada Byron

09:00 – 09:15	**Multimodal person identification in a smart room** *Jordi Luque, R. Morros, J. Anguita, M. Farrus, Dušan Macho, F. Marqués, C. Martínez, V. Vilaplana, Javier Herrando*
09:15 – 09:30	**Effectiveness of Short-Term Prosodic Features for Speaker Verification** *Íker Luengo, Eva Navas, Inmaculada Hernáez, Jon Sánchez, Ibon Saratxaga, Iñaki Sainz*
09:30 – 09:45	**Descripción Biométrica de la Voz a Partir de la Estructura Biomecánica de la Cuerda Vocal** *P. Gómez, R. Fernández, A. Álvarez, L. M. Mazaira, V. Rodellar, J. I. Godino*
09:45 – 10:00	**Estudio del uso de información prosódica en reconocimiento de locutor en ámbito forense** *Rubén Fernández, Carlos Fombella, Doroteo Torre, Eduardo López, Luis A. Hernández*
10:00 – 10:15	**Selección y pesado de parámetros acústicos mediante algoritmos genéticos para el reconocimiento del locutor** *Maider Zamalloa, Germán Bordel, Luis Javier Rodríguez, Mikel Peñagarikano, Juan Pedro Uribe*
10:15 – 10:30	**On the use of high-level information in speaker and language recognition** *Alberto Montero-Asenjo, Javier González-Domínguez, Daniel Ramos-Castro, Ignacio López-Moreno, Doroteo Torre, Joaquín González-Rodríguez*
10:40 – 11:00	**Pausa para Café**

	Conferencia Invitada
11:00 – 12:00	**Salón de Actos - Edificio Ada Byron**
	Where are the Speech Production Models in Our Speech Processing Systems? *Richard Rose, Department of Electrical and Computer Engineering, McGill University, Montreal Quebec*

Sesión Oral: Reconocimiento Automático del Habla
De 12:00 a 13:45 - Salón de Actos - Edificio Ada Byron

12:00 – 12:15	**Joint Training of Codebooks and Acoustic Models in Automatic Speech Recognition Using Semi-continuous HMMs** *Albino Nogueiras, Mónica Caballero, José B. Mariño*
12:15 – 12:30	**Automatic Phonetic Segmentation and Labelling of Spontaneous Speech** *Luis Coelho, Daniela Braga*
12:30 – 12:45	**Diseño e Implementación de una Interfaz Genérica para la Estimación de Autómatas de Estados Finitos Ponderados** *Mikel Peñagarikano, Germán Bordel, Luis Javier Rodríguez, Maider Zamalloa*
12:45 – 13:00	**Reconocimiento automático del habla en euskera empleando información lingüística** *Víctor Guijarrubia, M. Inés Torres*
13:00 – 13:15	**Segmentación de fonemas no supervisada basada en métodos kernel de máximo margen.** *Yago Pereiro, Vicent Wan, Odette Scharenborg, Ascensión Gallardo*
13:15 – 13:30	**On the use of MLP-distance to estimate Posterior Probabilities by KNN for Speech Recongnition** *Ana I. García, Carmen Peláez, Hervé Bourlard*
13:30 – 13:45	**Study of maximum a posteriori speaker adaptation for automatic speech recognition of pathological speech** *Óscar Saz, Carlos Vaquero, Eduardo Lleida, José Manuel Marcos, César Canalís*

	Clausura
13:45 – 14:00	**Salón de Actos - Edificio Ada Byron**
	Eduardo Lleida Solano, Presidente del Comité Organizador

ACOUSTIC EVENT DETECTION AND CLASSIFICATION IN SMART-ROOM ENVIRONMENTS: EVALUATION OF CHIL PROJECT SYSTEMS

Andrey Temko[1], Robert Malkin[2], Christian Zieger[3], Dusan Macho[1], Climent Nadeu[1], Maurizio Omologo[3]

[1]TALP Research Center, UPC, Barcelona, Spain
[2]interACT, CMU, Pittsburgh, USA
[3]ITC-irst, Povo (TN), Italy
[1]{temko, dusan, climent}@talp.upc.es, [2]rgmalkin@cs.cmu.edu, [3]{zieger, omologo}@itc.it

ABSTRACT

The identification of acoustic events that are produced in a meeting room environment may help to detect and describe the human and social activity that takes place in the room. In the framework of the CHIL project, three different sites have developed and tested several preliminary systems for acoustic event classification (AEC) and acoustic event detection (AED). A primary AED evaluation task with the testing portions of the isolated sound databases and the seminar recordings produced in CHIL was designed, and it was carried out in February 2006. Additionally, a secondary AEC evaluation task was carried out using only the isolated sound databases. In this paper, a short description of the systems is presented, and the evaluation setup and results, for both AED and AEC, are reported and discussed.

1. INTRODUCTION

Although speech is certainly the most informative acoustic event, other kind of sounds may also carry useful information in a meeting room environment. In fact, in that environment the human activity is reflected in a rich variety of acoustic events, either produced by the human body or by objects handled by humans. Consequently, detection or classification of acoustic events may help to detect and describe the human and social activity that takes place in the room. For example: clapping or laughter inside a speech discourse, a strong yawn in the middle of a lecture, a chair moving or door noise when the meeting has just started, etc Additionally, the robustness of automatic speech recognition systems may be increased by a previous detection of the non-speech sounds lying in the captured signals.

Acoustic Event Detection/Classification (AED/C) is a recent sub-area of computational auditory scene analysis [1] that deals with processing acoustic signals and converting them into symbolic descriptions corresponding to a listener's perception of the different sound events that are present in the signals and their sources. While acoustic event classification deals with events that have already been isolated from its temporal context, acoustic event detection refers to both identification and localization in time of events in continuous audio streams.

In this paper, we present the results of the AED/C CLEAR evaluations carried out in February 2006 by the three participant partners from the CHIL project [2] which sign this paper (UPC, CMU and ITC). The primary evaluation task was AED of the testing portions of the two isolated sound databases (from ITC and UPC) and 4 UPC's seminar recordings produced in CHIL. Additionally, a secondary AEC evaluation task was designed using only the isolated sound databases, and it is also included in this report. All the partners agreed the set of acoustic classes a priori before recording the databases. A common metric was also developed at the UPC and agreed with the other partners. ELDA was in charge of the scoring task. In this paper, the three participant sites present their own preliminary systems for the tasks of AED and AEC. Two of them are based on the classical Hidden Markov Model (HMM) [3] approach used in continuous speech recognition, and the other uses Support Vector Machine (SVM) [4] as the basic classifier. Since the evaluation procedure was not strictly defined, there are some differences between the degrees of fitting of the systems to the testing data: two partners developed specific systems for each room, but not the third; one partner uses a system trained differently for seminars and isolated event databases, etc. If those differences are neglected, it is observed that the system closest to the usual speech recognition approach offers better average AED results.

The paper is organized as follows: Section 2 gives the experimental setup. Specifically, the databases used in the evaluations are described in Subsection 2.1, while the evaluation scenario and metrics are given in Subsection 2.2 and 2.3, respectively. Section 3 reviews the systems used by each of the AED/C evaluation participants. The results obtained by the detection and classification systems in the CLEAR evaluations are shown and discussed in Section 4. Conclusions are presented in Section 5.

2. EVALUATION SETUP

The conducted experiments were carried out on 2 different kinds of databases, namely: 2 databases of isolated acoustic events recorded at the UPC and IRST, and 5 interactive seminars recorded at the UPC.

The two former databases contain a set of isolated acoustic events that occur in a meeting room environment and were recorded specially for the CHIL AED/C task. The recorded sounds do not have temporal overlapping and no interfering noises were present in the room.

The UPC database of isolated acoustic events [5] was recorded using 84 microphones, namely, Mark III (array of 64 microphones), three T-shape clusters (4 mics per cluster), 4 tabletop directional and 4 omni-directional microphones. The database consists of 13 semantic classes plus "unknown". Approximately 60 sounds per each of the sound classes were recorded as shown in Table 1. Ten people participated in recordings: 5 men and 5 women. There are 3 sessions per each participant. At each session, the participant took a different place in the room out of 7 fixed different positions.

The ITC database of isolated acoustic events [6] was recorded with 32 microphones. They were mounted in 7 T-shaped arrays (composed by 4 microphones each one) plus there were 4 table microphones. The database contains 16 semantic classes of events. Approximately 50 sounds per almost each of the sound classes were recorded as shown in Table 1. 9 people participated at the recordings. For each experiment 4 positions in the room were located. People swapped their positions after every session. During each session every person reproduced a complete set of acoustic events.

Additionally, the AED techniques were applied to the database of the interactive seminars [7] recorded at the UPC. 5 interactive seminars have been collected. The difference with two previous databases of isolated acoustic events is that seminars consist of real environment events that may have temporal overlapping with speech and/or other acoustic events. Each seminar consists of a 10-20 minutes presentation to a group of 3-5 attendees in a meeting room. During and after the presentation there are questions from the attendees with answers from the presenter. There is also activity in terms of people entering/leaving the room, opening and closing the door, standing up and going to the screen, some discussion among the attendees, coffee breaks, etc. The databases was recorded using 88 different sensors that include 3 4-microphoneT-shaped arrays, 1 64-microphone Mark III array, 4 omni-directional table-top microphones, 4 directional table-top microphones, and 4 close-talk microphones. The number of events of one of the seminars is summarized in Table 1.

The AED/C evaluation is done on 12 semantic classes that are defined as:

- Knock (door, table) [kn]
- Door slam [ds]
- Steps [st]
- Chair moving [cm]
- Spoon (cup jingle) [cl]
- Paper wrapping [pw]
- Key jingle [kj]
- Keyboard typing [kt]
- Phone ringing/Music [pr]
- Applause [ap]
- Cough [co]
- Laugh [la]

Table 1. *Number of events for the UPC and ITC databases of isolated acoustic events, and the UPC interactive seminar.*

Event type	Number of events		
	UPC -iso	ITC-iso	UPC-sem
Door knock	50	47	4
Door open	60	49	7
Door slam	61	51	7
Steps	73	50	43
Chair moving	76	47	26
Spoon/cup	64	48	15
Paper work	84	48	21
Key jingle	65	48	2
Keyboard	66	48	14
Phone ring	116	89	6
Applause	60	12	2
Cough	65	48	5
Laugh	64	48	8
Unknown	126		12
Mimo pen buzz		48	
Falling object		48	
Phone vibr.		13	
Speech			169

Also there are two other possible events that are present but are not evaluated

- Speech [sp]
- Unknown [un]

Actually, the databases of isolated acoustic events contain more semantic classes than the above-proposed list as shown in Table 1. For that reason, the classes that are out of the scope of the current AED/C evaluation were marked as "unknown".

Two main series of experiments are performed: AED and AEC. AED was done in both isolated and real environment conditions. For the task of AEC and isolated AED the databases of isolated acoustic events were split into training and testing parts, namely, for the UPC database sessions 1 and 2 were used for training and session 3 for testing; for the ITC database sessions 1-3 were used for training and session 4 for testing. For the task of AED in real environment all databases of isolated acoustic events and one of five seminars were allowed to use for training and developing, while for testing a 5-minute extract from each of the remaining 4 seminars was proposed forming in total 4 five-minute segments. The selection of extracted parts was done by ELDA.

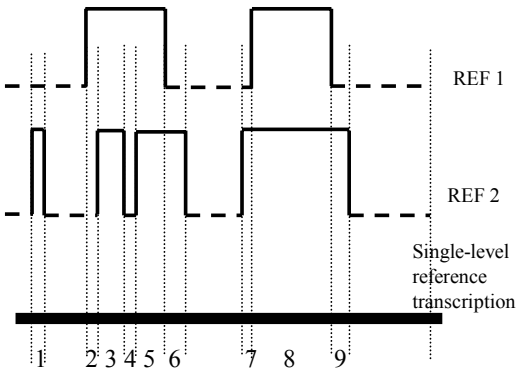


Figure 1. *From reference transcription with overlapping of level 2 to reference single-level transcription.*

Table 2. *Obtained single-level reference transcription and a list of events to detect*

Single-level reference *transcription*	*List of* ***events to detect****:*
1 – co1	1 – cough1
2 – la1	2 – laugh1
3 – la1_ds1	3 – ds1
4 – la1	4 – spoon1
5 – la1_cl1	5 – laugh2
6 – cl1	6 – keyboard1
7 – la2	
8 – kt1_la2	
9 – la2	

The primary evaluation task was defined as AED evaluated on both the isolated databases and the seminars.
As it was mentioned above, the acoustic events that happen in real environment may have temporal overlapping. The appropriate metric was developed to score the system outputs. It consists of two steps: projecting all levels of overlapping events into a single-level reference transcription and comparing a hypothesized transcription with the single level reference transcription.

For instance, suppose we have a reference that contain overlapping of level 2 and can be represented as shown in Figure 1 where REF_1: _la_kt_ and REF_2: _co_ds_cl_la_ model two overlapping acoustic event sequences. Then we can form the single-level reference transcription and a list of events to detect as shown in Table 2.

Following definitions are needed to compute the metric:

- An event is correctly detected when the hypothesized temporal center is situated in the appropriate *single-level reference* interval and the hypothesized label is a constituent or a full name of this interval *single-level reference* label. After an event is claimed to be correctly detected, it is marked as detected in the list of *events to detect.*
- Empty intervals are the reference intervals that contain speech, silence or events belonging to the "unknown" class.
- A substitution error occurs when the temporal center of the hypothesized event is situated in the appropriate *single-level reference* interval and the label of the hypothesized event is *not* constituent or the full name of the label of that *single-level reference* interval.
- A deletion error occurs when there is an event in the list of *events to detect* that is not marked as detected.
- An insertion error occurs when the temporal center of the hypothesized event is *not* situated in any of the *single-level reference* intervals (i.e. are situated in *empty intervals*)

Finally, Acoustic Event Error Rate (AEER) is computed as

AEER= (D+I+S)/N * 100

where N is the number of *events to detect*, D – *deletions*, I – *insertions*, and S – *substitutions*.

3. ACOUSTIC EVENT DETECTION AND CLASSIFICATION SYSTEMS

3.1. UPC AEC/D systems

A system based on SVM was used at the UPC for the task of AED/C. A DAG [8] multi-classification scheme was chosen to extend the SVM binary classifier to the multi-classification problem. 5-fold cross-validation [4] on the training data was applied to find the optimal SVM hyper parameters that were σ for the chosen Gaussian kernel, and C, a parameter that controls the amount of data allowed to be misclassified during the training procedure. In all the experiments the third channel of the Mark III microphone array was used.

Firstly, the sound is downsampled from the initial 44kHz sampling rate to 22 kHz, and framed (frame length=25ms, overlapping 50%, Hamming window). For each frame, the set of spectral parameters that showed the best results in [9] was extracted. It consists of the concatenation of two types of parameters: 1) 16 Frequency-Filtered (FF) log filter-bank energies [10] taken from ASR, and 2) a set of other perceptual parameters: zero-crossing rate, short time energy, 4 subband energies, spectral flux calculated for each of the defined subbands, and pitch. The first and second time derivatives were also calculated for the FF parameters. In total, a vector of 59 components is build to represent each frame.

3.1.1. AEC system

The mean, standard deviation, entropy and autocorrelation coefficient of the parameter vectors were computed along the whole event signal thus forming one vector per audio

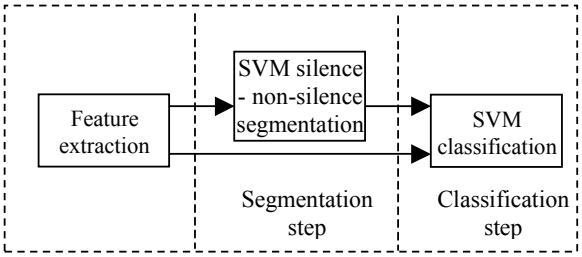


Figure 2. *UPC acoustic event detection system*

event with 4x59 elements. Then, that vector of statistical features was used to feed the SVM classifier, which was trained on the training set of the two databases of isolated acoustic events. The resulting system, herewith named "UPC-C", was used to test both UPC and ITC databases of isolated acoustic events, so neither feature nor system adaptation related to a specific database was applied.

3.1.2. AED system

The scheme of the AED system herewith named "UPC-D" is shown in Figure 2. Using a sliding window of one second with a 100ms shift, a vector of 4x59 statistical features was extracted like in the AEC system described in the last sub-section for each position of the window (every 100ms).

The statistical feature vector is then fed to an SVM-based silence/non-silence classifier trained on silence and non-silence segments of the two isolated acoustic events databases. At the output, a binary sequence of decisions is obtained. A median-filter of size 17 is applied to eliminate too short silences or non-silences.

Then, the SVM-based event classifier is applied to each detected non-silence segment. The event classifier was trained on a parameters extracted from a sliding window with 100ms shift applied to each event in the way that the first and the last windows still include more than 50% of the event content. The event classifier is trained on both isolated acoustic events and seminar databases to classify a set of 12 defined acoustical classes, plus classes "speech" and "unknown". A sequence of decisions made on a 1-second window every 100ms is obtained within the non-silence segment. That sequence is smoothed by assigning to the current decision point the label that is most frequent in a string of five decision points around the current one. Also, a confidence measure is calculated for each point as the quotient between the number of times that the chosen label appears in the string and the number of labels in the string (5).

The sequence of decisions from the non-silence segment is then processed again to get the detected events. In that step, only the events that have their length equal or larger than the average event length are kept, and the number of events kept in the non-silence segment is forced to be lower than a number which is proportional to the length of the

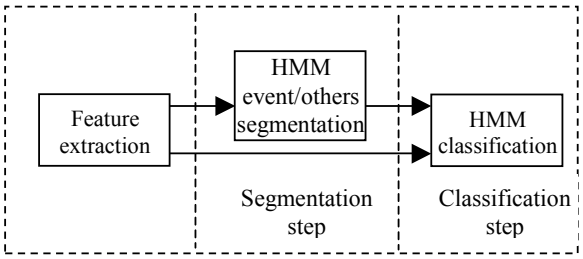


Figure 3. *CMU acoustic event detection system*

segment. The average length of the events is estimated from the training and development databases. Finally, if the average of the above mentioned computed confidences in a detected event is less than a threshold, the hypothesized event is marked as "unknown"; otherwise, it maintains the assigned label.

3.2. CMU AEC/D systems

The CMU acoustic event classification and detection systems were based on continuous density HMMs. We first downsampled the input signal from a single microphone to 16kHz, 2-byte quality. From this signal, we extracted 15 Mel-Frequency Cepstral Coefficients (MFCCs) at a rate of 100 frames per second. We additionally normalized these MFCCs to zero mean and unity variance using means and variances specific to each site. We used custom HMM topologies for each sound class; these topologies were induced using the k-variable k-means algorithm due to Reyes-Gomez and Ellis [11]. The k-variable k-means algorithm is a greedy approach to topology induction based on the leader-follower clustering paradigm; it uses a threshold to control the tendency to add new states to a class HMM.

We trained five complete sets of class HMMs using all available data from the isolated databases. After training these five complete HMM sets, we further trained site-specific feature space adaptation matrices that are reflected in systems "CMU-C1" and "CMU-C2". We used the maximum likelihood approach suggested by Leggetter and Woodland [12] and Gales [13]. Finally, as suggested by Reyes-Gomez and Ellis, we explored the combination of scores of HMMs trained with different thresholds on a per-site basis. We found that by combining three models for the ITC data and two for the UPC data, we were able to achieve a combined misclassification rate of less than 6% for acoustic event classification task.

For the acoustic event detection task, we wished to explore the possibility of pre-segmenting the data with a simple HMM before applying our more complex classification HMMs which used more than one Viterbi path to assign a final score. The scheme of the system is presented in Figure 3. Hence, we trained segmentation HMMs which included three classes: speech, CHIL event, and other. To train these HMMs, we used the same approach as for the

classification systems above, except that we added the UPC seminar data for training. The detection systems herewith will be named "CMU-D1" and "CMU-D2". We chose the optimal HMMs for segmentation on a per-site basis. Further, since we also needed to control the rate at which these HMMs created segments in the data, we optimized separate insertion penalties for the ITC isolated database, the UPC isolated database, and the UPC seminar database. This approach yielded poor results on the isolated condition, and very poor results for the seminar condition.

3.3. ITC AEC/D systems

The AED/C system that was studied at the ITC-irst is based on continuous density HMM. The scheme of the system is presented in Figure 4.

A signal acquired by a single microphone belonging to a T-shaped array was used in experiments. The front-end processing is based on 12 Mel-Frequency Cepstral Coefficients (MFCCs) [3] and log-energy of the signal. The analysis step is 10 ms with a Hamming window of 20 ms. The resulting parameters together with their first and second order time derivatives are arranged into a single observation vector of 39 components.

Each event is described by a 3 state HMM model. All of the HMMs have a left-to-right topology and use output probability densities represented by means of 32 Gaussian components with diagonal covariance matrices. HMM training was accomplished through the standard Baum-Welch training procedure.

For AEC task two different sets of models were created to fit the ITC and UPC rooms; the corresponding systems will herewith be named "ITC-C1" and "ITC-C2". The first one is trained on the ITC isolated acoustic events database and the other is trained on the UPC isolated acoustic events database. The selected training data refers to the recordings of a single microphone belonging to a T-shaped array.

For the AED task the same models adopted in the AEC task were used, but also the models of speech and silence were added; the corresponding systems will herewith be named "ITC-D1" and "ITC-D2". To train the model of speech, recordings of meetings in the ITC room were used, while for the model of silence the database of isolated acoustic events was used exploiting the silence moments between each event.

To overcome the detection of events that are overlapped with speech that occur in the interactive seminars, the strategy based on the contamination of the events with speech in the training procedure was exploited; that is reflected in the system "ITC-D3". An artificial database was created by adding speech to the isolated events imposing different SNR values, from 0 to 15 dB. At this moment the system is not trained to detect events that overlap with other events except speech.

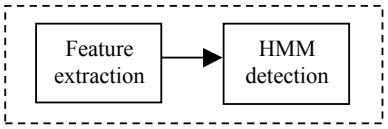


Figure 4. ITC acoustic event detection system

4. RESULTS AND DISCUSSION

Table 3 shows classification error rates obtained using different classification systems described previously. Since the evaluation procedure was not strictly defined, there are some differences in the degree of fitting of the systems to the two testing databases (ITC and UPC isolated DB): both CMU and ITC systems use two sets of models, one for each testing database, while UPC system uses one set of models for the both testing databases. We can observe that the system based on SVM obtained the same or better results than the systems based on the HMM technology, despite the fact that database-specific systems were used in the case of HMM.

In the detection task, as explained in the previous sections, participants took two different approaches: a) First performing segmentation and then classification (UPC and CMU systems) b) Merging the segmentation and classification in one step as performed by the Viterbi search in the state-of-the-art ASR systems (ITC systems) Table 4 shows detection error rates for the two isolated event databases and the interactive seminar database. The lowest detection error rates are obtained by the ITC systems, which are based on the approach b). Notice that both CMU and UPC systems achieved better results than the ITC systems in the classification task (Table 3), however they both rely on a previous segmentation step (the approach a)). If we add up the results obtained for the detection task for both isolated and seminar conditions neglecting the test-specificities of the CMU and ITC systems, we obtain the following error rates: UPC: 69.6%, CMU: 80.5%, ITC: 46.8%. Although there might be a number of reasons to explain the differences

Table 3. *Error rates (in %) for AE classification task of the systems explained in Section 3*

Systems / Databases	UPC C	CMU C1	CMU C2	ITC C1	ITC C2
ITC iso	4.1	7.5	----	12.3	----
UPC iso	5.8	----	5.8	----	6.2

Table 4. *Error rates (in %) for AE detection task of the systems explained in Section 3*

Sys / DBs	UPC D	CMU D1	CMU D2	ITC D1	ITC D2	ITC D3
ITC iso	64.6	45.2	----	23.6	----	----
UPC iso	58.9	----	52.5	----	33.7	----
UPC sem	97.1	----	177.3	----	----	99.3

across the systems, we conjecture that the initial segmentation step included in both UPC and CMU systems, but not in the ITC systems, is the main cause of the lower overall detection performance of these systems. Further investigation is needed in the direction of the approach a) to see whether it can outperform the well-established scheme b).

Besides, it can be seen from the Table 4, that the error rates increase significantly for the UPC seminar database. One of possible reasons of such a bad performance is that it is difficult to detect low-energy acoustic classes that overlap with speech, such as e.g. “chair moving”, “steps”, “keyboard typing”, and “paper work”. Actually, these classes cover the majority of the events in the UPC seminars and probably they are the cause of the bad results we obtained in the seminar task. A usage of multiple microphones might be helpful in this case.

5. CONCLUSIONS

The presented work focused on the CLEAR evaluation tasks concerning the detection and classification of acoustic events that may happen in a lecture/meeting room environment. In this context, we evaluated two different tasks, acoustic event classification (AEC) and acoustic event detection (AED), AED being the primary objective of the evaluation. Two kinds of databases were used, two databases of isolated acoustic events and a database of interactive seminars containing a significant number of acoustic events of interest.

Preliminary detection and classification systems from three different participants were presented, which allowed an evaluation of different approaches for both classification and detection. The UPC system is based on the Support Vector Machine (SVM) discriminative approach and uses Frequency Filtering features and four kinds of perceptual features. Both the CMU and ITC systems are based on the Hidden Markov Model (HMM) generative approach and they use MFCC features.

In the classification task, the UPC SVM-based system showed better performance than the two systems based on HMM. In the detection task, we could see two different approaches: a) first performing segmentation and then classification (UPC and CMU systems), and b) merging the segmentation and classification in one step as performed by the Viterbi search in the state-of-the-art Automatic Speech Recognition (ASR) systems (ITC systems). In the presented results, the approach b) showed better performance than the approach a). Notice however that the b) approach (and actually the ITC systems) is a well-established ASR approach developed for many years and thus can be considered as a challenging reference for the other presented approaches/systems in the acoustic event detection task.

6. ACKNOWLEDGEMENTS

This work has been partially sponsored by the EC-funded project CHIL (IST-2002-506909). Authors wish to thank Djamel Mostefa and Nicolas Moreau from ELDA for their role in the transcription of the seminar data and in the scoring task. UPC authors have been partially sponsored by the Spanish Government-funded project ACESCA (TIN2005-08852).

7. REFERENCES

[1] D. Wang, G. Brown, Computational Auditory Scene Analysis: Principles, Algorithms and Applications, Wiley-IEEE Press, 2006

[2] CHIL - Computers In the Human Interaction Loop, http://chil.server.de/

[3] L. Rabiner, B. Juang, Fundamentals of Speech Recognition, Prentice Hall, 1993

[4] B. Schölkopf, A. Smola, Learning with Kernels, MIT Press, Cambridge, MA, 2002

[5] A. Temko, D. Macho, C. Nadeu, C. Segura, “UPC-TALP Database of Isolated Acoustic Events”, Internal UPC report, 2005

[6] C. Zieger, M. Omologo, “Acoustic Event Detection - ITC-irst AED database”, Internal ITC report, 2005

[7] J. Casas, R. Stiefelhagen, et al, “Multi-camera/multi-microphone system design for continuous room monitoring,” CHIL-WP4-D4.1-V2.1-2004-07-08-CO, CHIL Consortium Deliverable D4.1, July 2004

[8] J. Platt et al., “Large Margin DAGs for Multiclass Classification”, Proc. Advances in Neural Information Processing Systems 12, pp. 547-553, 2000

[9] A. Temko, C. Nadeu, “Classification of meeting-room acoustic events with Support Vector Machines and Confusion-based Clustering”, Proc. ICASSP’05, pp. 505-508, 2005

[10] C. Nadeu et al., “On the decorrelation of filter-bank energies in speech recognition”, Proc. Eurospeech’95, pp. 1381-1384, 1995

[11] M. Reyes-Gomez and D. Ellis, “Selection, Parameter Estimation, and Discriminative Training of Hidden Markov Models for General Audio Modeling”, Proc. ICME’03, 2003

[12] C. Leggetter and P. Woodland, “Speaker Adaptation of Continuous Density HMMs using Multivariate Regression”, Proc. ICSLP’94, 1994

[13] M. Gales, “Maximum Likelihood Linear Transformations for HMM-based Speech Recognition”, Computer Speech and Language, 1998

TIME-DEPENDENT CROSS-PROBABILITY MODEL FOR FEATURE VECTOR NORMALIZATION

Luis Buera, Eduardo Lleida, Antonio Miguel, Alfonso Ortega, Óscar Saz

Communication Technologies Group (GTC)
I3A, University of Zaragoza, Spain
{lbuera,lleida,amiguel,ortega,oskarsaz}@unizar.es

ABSTRACT

In previous works, Multi-Environment Model based LInear Normalization, MEMLIN, and Phoneme-Dependent MEMLIN, PD-MEMLIN, were presented and they were proved to be effective to compensate environment mismatch. Both are empirical feature vector normalization techniques which model clean and noisy spaces with Gaussian Mixture Models, GMMs, and the probability of the clean model Gaussian, given the noisy model one and the noisy feature vector (cross-probability model) is a critical point in both algorithms. In the previous works the cross-model probability was approximated as time-independent. However, in this paper, a time-dependent estimation based on GMM is proposed for MEMLIN and PD-MEMLIN. Some experiments with SpeechDat Car database were carried out in order to study the performance of the proposed estimation of the cross-probability model in a real acoustic environment, obtaining important improvements: 78.48% and 76.76% of mean improvement in Word Error Rate, WER, for MEMLIN and PD-MEMLIN, respectively (70.21% and 75.44% if time-independent cross-probability model is applied).

1. INTRODUCCIÓN

When training and testing acoustic conditions differ, the accuracy of speech recognition systems rapidly degrades. To compensate for this mismatch, robustness techniques have been developed along the following two main lines of research: acoustic model adaptation methods, and feature vector adaptation/normalization methods. Also, some of the techniques can be combined to generate hybrid solutions, which are effective under certain conditions [1]. In general, acoustic model adaptation methods produce the best results [2] because they can model the uncertainty caused by the noise statistics. However, these methods require more data and computing time than do feature vector adaptation/normalization methods, which do not produce as good results but provide more on line solutions. So, the choice of a robustness technique depends on the characteristics of the application in each situation.

Feature vector adaptation/normalization methods fall into one of three main classes [3]: high-pass filtering, which contains very simple methods such Cepstral Mean Normalization, CMN, model-based techniques, which assumes a structural model of environmental degradation, and empirical compensation, which uses direct cepstral comparisons. In any case, and independently of the class, some algorithms assume a prior probability density function (pdf) for the estimation variable. In those cases, a Bayesian estimator can be used to estimate the clean feature vector. The most commonly used criterion is to minimize the Mean Square Error (MSE), and the optimal estimator for this criterion, Minimum Mean Square Error (MMSE), is the mean of the posterior pdf. Methods, such as Stereo-based Piecewise Linear Compensation for Environments (SPLICE) [4], Multi-Environment Model-based LInear Normalization (MEMLIN) [5], or Phoneme-Dependent MEMLIN [6], use the MMSE estimator to compute the estimated clean feature vector.

The previous works [5] [6] show that MEMLIN and PD-MEMLIN are effective to compensate the effects of dynamic and adverse car conditions. MEMLIN is an empirical feature vector normalization technique based on stereo data and the MMSE estimator. MEMLIN splits the noisy space into several basic environments and each of them and clean feature space are modelled using GMMs. Therefore, a bias vector transformation is associated with each pair of Gaussians from the clean and the noisy basic environment spaces. On the other hand, the main difference of PD-MEMLIN concerning MEMLIN consists on splitting the clean and noisy basic environments into phonemes that are modelled using GMMs.

A critical point in MEMLIN and PD-MEMLIN is the estimation of the cross-probability model: the probability of the clean model Gaussian, given the noisy model one, the noisy feature vector and the phoneme (only in PD-MEMLIN). In [5] [6], a time-independent solution is considered. This work focuses on this term and it is proposed a time-dependent solution, modelling the noisy feature vectors associated to each pair of Gaussians from the clean and the noisy basic environment spaces (and the phoneme in PD-MEMLIN) with a GMM.

This paper is organized as follows: In Section 2, an overview of PD-MEMLIN is detailed. In Section 3, some experiments are presented to show the importance of the cross-probability model estimation. The new proposed cross-probability model based on GMM is explained in Section 4. The results with Spanish SpeechDat Car database [7] are included in Section 5. Finally, the conclusions are presented in Section 6.

2. PD-MEMLIN OVERVIEW

Phoneme Dependent Multi-Environment Models based LInear Normalization is an empirical feature vector normalization technique which uses stereo data in order to estimate the different compensation linear transformations in a previous training process. The clean feature space is modelled as a mixture of Gaussians for each phoneme. The noisy space is split in several basic acoustic environments and each environment is modelled as a mixture of Gaussians for each phoneme. The transformations are estimated for all basic environments between a clean phoneme Gaussian and a noisy Gaussian of the same phoneme. This can be shown in Fig. 1 for one environment, where s_x^{ph} is the clean model Gaussian associated to the phoneme ph, $s_y^{e,ph}$ is the noisy model Gaussian associated to the basic environment

This work has been supported by the national project TIN 2005-08660-C04-01.

e and the phoneme ph, and finally, $r_{s_x^{ph},s_y^{e,ph}}$, which is the bias vector transformation between s_x^{ph} and $s_y^{e,ph}$.

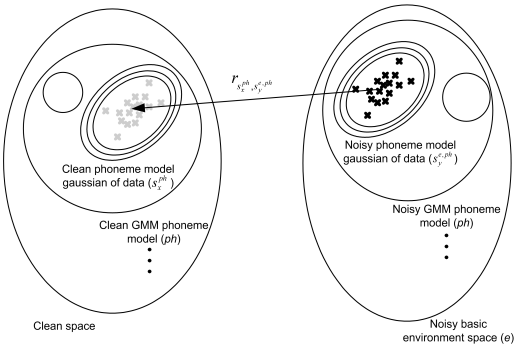


Figure 1. Scheme of PD-MEMLIN transformations for one environment.

2.1. PD-MEMLIN approximations

• Clean feature vectors, $\mathbf{x}_t$, are modelled using a GMM of C components for each phoneme, ph (assuming that all the phonemes are modelled with the same number of components)

$$p_{ph}(\mathbf{x}_t) = \sum_{s_x^{ph}=1}^{C} p(\mathbf{x}_t|s_x^{ph})p(s_x^{ph}), \quad (1)$$

$$p(\mathbf{x}_t|s_x^{ph}) = N(\mathbf{x}_t; \mu_{s_x^{ph}}, \mathbf{\Sigma}_{s_x^{ph}}), \quad (2)$$

where $\mu_{s_x^{ph}}$, $\mathbf{\Sigma}_{s_x^{ph}}$, and $p(s_x^{ph})$ are the mean vector, the diagonal covariance matrix, and the a priori probability associated with the clean model Gaussian s_x^{ph} of the ph phoneme.

• Noisy space is split into several basic environments, e, and the noisy feature vectors, $\mathbf{y}_t$, are modeled as a GMM of C' components for each basic environment and phoneme (assuming that all the phonemes of the all the basic environments are modelled with the same number of components)

$$p_{e,ph}(\mathbf{y}_t) = \sum_{s_y^{e,ph}=1}^{C'} p(\mathbf{y}_t|s_y^{e,ph})p(s_y^{e,ph}), \quad (3)$$

$$p(\mathbf{y}_t|s_y^{e,ph}) = N(\mathbf{y}_t; \mu_{s_y^{e,ph}}, \mathbf{\Sigma}_{s_y^{e,ph}}), \quad (4)$$

where $s_y^{e,ph}$ denotes the corresponding Gaussian of the noisy model for the e basic environment and the ph phoneme; $\mu_{s_y^{e,ph}}$, $\mathbf{\Sigma}_{s_y^{e,ph}}$, and $p(s_y^{e,ph})$ are the mean vector, the diagonal covariance matrix, and the a priori probability associated with $s_y^{e,ph}$.

• Clean feature vectors can be approximated as a linear function, Ψ, of the noisy feature vector which depends on the basic environments, the phonemes and the clean and noisy model Gaussians: $\mathbf{x} \approx \Psi(\mathbf{y}_t, s_x^{ph}, s_y^{e,ph}) = \mathbf{y}_t - \mathbf{r}_{s_x^{ph},s_y^{e,ph}}$, where $\mathbf{r}_{s_x^{ph},s_y^{e,ph}}$ is the bias vector transformation between noisy and clean feature vectors for each pair of Gaussians, s_x^{ph} and $s_y^{e,ph}$.

2.2. PD-MEMLIN enhancement

With those approximations, PD-MEMLIN transforms the MMSE estimation expression, $\hat{\mathbf{x}}_t = E[\mathbf{x}|\mathbf{y}_t]$, into (5), where $p(e|\mathbf{y}_t)$ is the a posteriori probability of the basic environment; $p(ph|\mathbf{y}_t, e)$ is the a posteriori probability of the phoneme, given the noisy feature vector and the environment; $p(s_y^{e,ph}|\mathbf{y}_t, e, ph)$ is the a posteriori probability of the noisy model Gaussian, $s_y^{e,ph}$, given the feature vector, $\mathbf{y}_t$, the basic environment, e, and the phoneme, ph. To estimate those terms: $p(e|\mathbf{y}_t)$, $p(ph|\mathbf{y}_t, e)$ and $p(s_y^{e,ph}|\mathbf{y}_t, e, ph)$, equations (3) and (4) are applied as described in [6]. Finally, the cross-probability model, $p(s_x^{ph}|\mathbf{y}_t, e, ph, s_y^{e,ph})$, is the probability of the clean model Gaussian, s_x^{ph}, given the feature vector, $\mathbf{y}_t$, the basic environment, e, the phoneme, ph, and the noisy model Gaussian, $s_y^{e,ph}$. The cross-probability model is estimated in a training phase using stereo data for each basic environment and phoneme $(\mathbf{X}_{e,ph}, \mathbf{Y}_{e,ph}) = (\mathbf{x}_1^{e,ph}, \mathbf{y}_1^{e,ph}); ...; (\mathbf{x}_{t_{e,ph}}^{e,ph}, \mathbf{y}_{t_{e,ph}}^{e,ph}); ...; (\mathbf{x}_{T_{e,ph}}^{e,ph}, \mathbf{y}_{T_{e,ph}}^{e,ph})$, with $t_{e,ph} \in [1, T_{e,ph}]$ [6]. The cross-probability model is computed avoiding the time dependence given by the noisy feature vector as (6) (time-independent cross-probability model). On the other hand, the bias vector transformation, $\mathbf{r}_{s_x^{ph},s_y^{e,ph}}$, is also computed using the stereo data in the previous training phase [6].

The expressions for MEMLIN can be obtained directly from the PD-MEMLIN ones if only one phoneme is considered [5].

3. CROSS-PROBABILITY MODEL PERFORMANCE

To study the performance of the cross-probability model in a qualitative way, the histograms and scattegrams between the first Mel Frequency Cepstral Coefficients (MFCCs) in non-silence frames for different signals are depicted in Fig. 2.

Figure 2.a, which represents clean and noisy in real car conditions feature vectors, shows the effects of car noise. The pdf of clean first MFCCs is clearly affected (Fig.2.a.1), and the uncertainty is increased (Fig.2.a.2).

Since we only want to observe the importance of the cross-probability model, and PD-MEMLIN performance is highly dependent of the probability of the phoneme, given the environment and the noisy feature vector, $p(ph|\mathbf{y}_t, e)$, we present results with MEMLIN in Fig. 2.b and 2.c. So, clean and normalized coefficients with MEMLIN are represented in the Fig. 2.b. MEMLIN is applied with 128 Gaussians. The pdf of normalized first MFCCs has been approximated to the clean signal one (Fig. 2.b.1), and the uncertainty has been reduced (Fig. 2.b.2). The peak that appears in Fig. 2.b.1 is due to the transformation of noisy feature vectors towards the clean silence.

Finally, Fig. 2.c represents clean and normalized with MEMLIN feature vectors where the cross-probability model is computed with the corresponding clean feature vector as (7). MEMLIN is also applied with 128 Gaussians. In this case the pdf of the normalized signal is almost the same that the clean one (Fig. 2.c.1) and the uncertainty is drastically reduced (Fig. 2.c.2). Furthermore, the WER results in this case are almost the same that we would obtain with clean signal. These results verify the importance of a good estimation of the cross-probability model in MEMLIN algorithm. Similar experiments were carried out with PD-MEMLIN, obtain a similar satisfactory performance.

$$p(s_x|\mathbf{y}_t, e, s_y^e) \simeq \frac{p(s_x)p(\mathbf{x}_t|s_x)}{\sum_{s_x} p(s_x)p(\mathbf{x}_t|s_x)}. \quad (7)$$

4. CROSS-PROBABILITY MODEL BASED ON GMM

To improve the time-independent cross-probability model, we propose to model the noisy feature vectors associated to a pair of Gaussians (s_x and s_y) with a GMM of C'' components (as-

$$\hat{\mathbf{x}}_t = \mathbf{y}_t - \sum_e \sum_{ph} \sum_{s_y^{e,ph}} \sum_{s_x^{ph}} \mathbf{r}_{s_x^{ph}, s_y^{e,ph}} p(e|\mathbf{y}_t) p(ph|\mathbf{y}_t, e) p(s_y^e|\mathbf{y}_t, e, ph) p(s_x^{ph}|\mathbf{y}_t, e, ph, s_y^e). \quad (5)$$

$$p(s_x^{ph}|\mathbf{y}_t, e, ph, s_y^{e,ph}) \simeq p(s_x^{ph}|s_y^{e,ph}) = \frac{\sum_{t_{e,ph}} p(\mathbf{x}_{t_{e,ph}}^{e,ph}|s_x^{ph}) p(\mathbf{y}_{t_{e,ph}}^{e,ph}|s_y^{e,ph}) p(s_x^{ph}) p(s_y^{e,ph})}{\sum_{t_{e,ph}} \sum_{s_x^{ph}} p(\mathbf{x}_{t_{e,ph}}^{e,ph}|s_x^{ph}) p(\mathbf{y}_{t_{e,ph}}^{e,ph}|s_y^{e,ph}) p(s_x^{ph}) p(s_y^{e,ph})}. \quad (6)$$

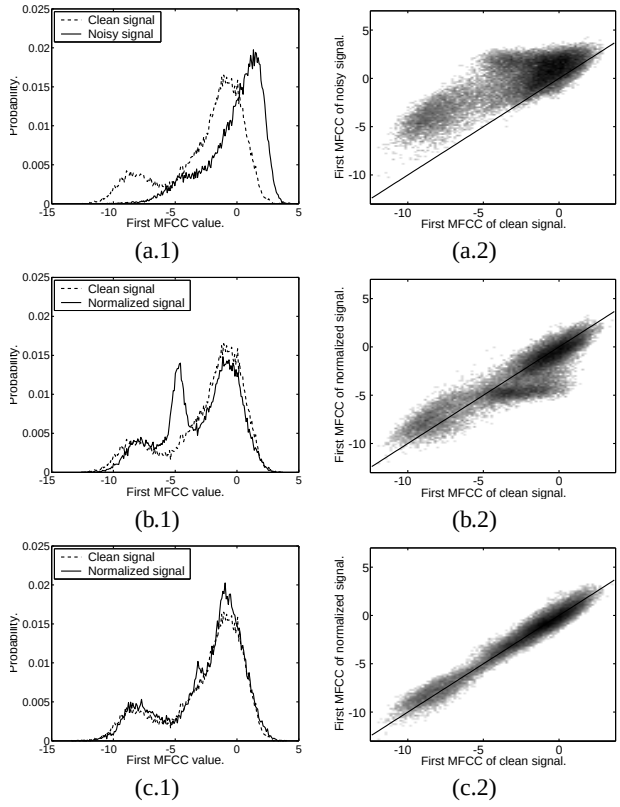


Figure 2. Scattegrams and histograms between the first MFCC in non-silence frames for different signals. The line in the scattergrams represents the function $x = y$.

suming that all the pair of Gaussians, s_x and s_y, are modelled with the same number of Gaussians

$$p(\mathbf{y}_t|s_x, s_y) = \sum_{s_y'=1}^{C''} p(\mathbf{y}_t|s_x, s_y, s_y') p(s_y'|s_x, s_y), \quad (8)$$

$$p(\mathbf{y}_t|s_x, s_y, s_y') = N(\mathbf{y}_t; \mu_{s_x,s_y,s_y'}, \mathbf{\Sigma}_{s_x,s_y,s_y'}), \quad (9)$$

where $\mu_{s_x,s_y,s_y'}$, $\mathbf{\Sigma}_{s_x,s_y,s_y'}$, and $p(s_y'|s_x, s_y)$ are the mean, the diagonal covariance matrix, and the a priori probability associated with s_y' Gaussian of the cross-probability GMM associated with s_x and s_y. To train these three parameters, the EM algorithm [8] is applied. The basic environments and the phonemes are not indexed for clarity, but they can be considered independently.

Let a set of clean and noisy stereo data available to learn the corresponding cross-probability GMM parameters $(\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}_1, \mathbf{y}_1), ...(\mathbf{x}_n, \mathbf{y}_n)..., (\mathbf{x}_N, \mathbf{y}_N)\}$. Each $\mathbf{y}_n$ can be seen as an incomplete component-labelled frame, which is completed by two indicator vectors. The first one is $\mathbf{w}_n \in \{0, 1\}^{C'}$, with 1 in the position corresponding to the s_y Gaussian generating $\mathbf{y}_n$ and zeros elsewhere ($\mathbf{W} = \{\mathbf{w}_1, ..., \mathbf{w}_N\}$). The second indicator vector is $\mathbf{z_n} \in \{0, 1\}^{C''}$, with 1 in the position corresponding to the s_y' Gaussian of the cross-probability GMM generating $\mathbf{y}_n$ and zeros elsewhere ($\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_N\}$). Each $\mathbf{x}_n$ can be seen also as an incomplete component-labelled frame, which is completed by one indicator vector: $\mathbf{v}_n \in \{0, 1\}^C$, with 1 in the position corresponding to the s_x Gaussian generating $\mathbf{x}_n$ and zeros elsewhere ($\mathbf{V} = \{\mathbf{v}_1, ..., \mathbf{v}_N\}$). The indicator vectors are called missing data, too. So, the complete data pdf is

$$\begin{aligned} p(\mathbf{x}, \mathbf{y}, \mathbf{v}, \mathbf{w}, \mathbf{z}) \quad &\simeq \quad p(\mathbf{v}, \mathbf{w}) p(\mathbf{x}|\mathbf{v}, \mathbf{w}) \times \\ &\quad p(\mathbf{v}, \mathbf{w}, \mathbf{z}) p(\mathbf{y}|\mathbf{v}, \mathbf{w}, \mathbf{z}), \end{aligned} \quad (10)$$

where it is assumed that $\mathbf{x}$ is independent of $\mathbf{y}$ and $\mathbf{z}$. Since the indicator vectors are Multinomial, the complete data pdf can be expressed as (11), where v_{s_x}, w_{s_y} and $z_{s_y'}$ are the components of $\mathbf{v}$, $\mathbf{x}$ and $\mathbf{z}$ associated to the Gaussians s_x, s_y and s_y', respectively.

The EM algorithm is applied iteratively in two steps. The Expectation (E) step, which estimates the expected values of the missing data, and the Maximization (M) step, which obtains the parameters of the cross-probability GMM using the estimated missing data.

4.1. The E step

To evaluate the E step, the function $Q(\mathbf{\Theta}|\mathbf{\Theta}^{(k)})$ is defined as $Q(\mathbf{\Theta}|\mathbf{\Theta}^{(k)}) = E[log(p(\mathbf{X}, \mathbf{Y}, \mathbf{V}, \mathbf{W}, \mathbf{Z}|\mathbf{\Theta}))|\mathbf{X}, \mathbf{Y}, \mathbf{\Theta}^{(k)}]$, where $E[\bullet]$ is the expected value, k is the iteration index and $\mathbf{\Theta}$ includes the unknown parameters of the cross-probability GMM. It is expressed as (12), where

$$(v_{s_x} w_{s_y})^{(k)} \simeq E[v_{s_x}|\mathbf{x}_n] E[w_{s_y}|\mathbf{y}_n], \quad (13)$$

$$(v_{s_x} w_{s_y} z_{s_y'})^{(k)} \simeq (v_{s_x} w_{s_y})^{(k)} E[z_{s_y'}|\mathbf{y}_n, v_{s_x}, w_{s_y}, \mathbf{\Theta}^{(k)}], \quad (14)$$

where it is assumed that v_{s_x} and w_{s_y} are independent, $E[v_{s_x}|\mathbf{x}_n, \mathbf{y}_n, \mathbf{\Theta}^{(k)}] \simeq E[v_{s_x}|\mathbf{x}_n]$ and $E[w_{s_y}|\mathbf{x}_n, \mathbf{y}_n, \mathbf{\Theta}^{(k)}] \simeq E[w_{s_y}|\mathbf{y}_n]$. $E[z_{s_y'}|\mathbf{y}_n, v_{s_x}, w_{s_y}, \mathbf{\Theta}^{(k)}]$ is estimated with (8) and (9) as (15), and $E[v_{s_x}|\mathbf{x}_n]$ and $E[w_{s_y}|\mathbf{y}_n]$ are computed in a similar way with (1) and (2), and with (3) and (4), respectively, assuming that there is only one phoneme. Although, in this work, to simplify, $E[v_{s_x}|\mathbf{x}_n]$ and $E[w_{s_y}|\mathbf{y}_n]$ values are 1, if the corresponding Gaussians are the most probable ones, and 0 in any other case (hard Gaussian estimation approach).

4.2. The M step

To obtain the maximum likelihood estimates for the parameters of the cross-probability GMM, $Q(\mathbf{\Theta}|\mathbf{\Theta}^{(k)})$ is maximized with respect to them. So, the corresponding expressions for the $(k+1)$th iteration are

$$p(\mathbf{x},\mathbf{y},\mathbf{v},\mathbf{w},\mathbf{z}) \simeq \prod_{s_x}\prod_{s_y}[p(v_{s_x}=1,w_{s_y}=1)p(\mathbf{x}|v_{s_x}=1,w_{s_y}=1)]^{v_{s_x}w_{s_y}} \times \prod_{s_x}\prod_{s_y}\prod_{s'_y}[p(v_{s_x}=1,w_{s_y}=1,z_{s'_y}=1)p(\mathbf{y}|v_{s_x}=1,w_{s_y}=1,z_{s'_y}=1)]^{v_{s_x}w_{s_y}z_{s'_y}}. \quad (11)$$

$$Q(\mathbf{\Theta}|\mathbf{\Theta}^{(k)}) = \sum_n\sum_{s_x}\sum_{s_y}(v_{s_x}w_{s_y})^{(k)}[log(p(s_x)p(s_y)) + log(p(\mathbf{x}_n|v_{s_x}=1,w_{s_y}=1))] + \sum_n\sum_{s_x}\sum_{s_y}\sum_{s'_y}(v_{s_x}w_{s_y}z_{s'_y})^{(k)}[log(p(s_x)p(s_y)p(s'_y|s_x,s_y)) + log(p(y_n|v_{s_x}=1,w_{s_y}=1,z_{s'_y}=1))]. \quad (12)$$

$$E[z_{s'_y}|\mathbf{y}_n,v_{s_x},w_{s_y},\mathbf{\Theta}^{(k)}] = \frac{p(s'_y|s_x,s_y)^{(k)}N(\mathbf{y}_n|\mu^{(k)}_{s_x,s_y,s'_y},\mathbf{\Sigma}^{(k)}_{s_x,s_y,s'_y})}{\sum_{s'_y}p(s'_y|s_x,s_y)^{(k)}N(\mathbf{y}_n|\mu^{(k)}_{s_x,s_y,s'_y},\mathbf{\Sigma}^{(k)}_{s_x,s_y,s'_y})}. \quad (15)$$

Train	Test	E1	E2	E3	E4	E5	E6	E7	MWER (%)
CLK	CLK	1.90	2.64	1.81	1.75	1.62	0.64	0.35	1.75
CLK	HF	5.91	14.49	14.55	20.17	21.07	16.19	35.71	16.21
HF	HF	6.67	14.24	12.73	12.91	14.97	9.68	8.50	11.81
†HF	HF	2.86	7.12	4.34	4.39	7.63	4.60	4.76	5.30

Table 1. WER baseline results, in %, from the different basic environments (E1,..., E7).

$$p(s'_y|s_x,s_y)^{(k+1)} = \frac{\sum_n(v_{s_x}w_{s_y}z_{s'_y})^{(k)}}{\sum_n\sum_{s'_y}(v_{s_x}w_{s_y}z_{s'_y})^{(k)}}. \quad (16)$$

$$\mu^{(k+1)}_{s_x,s_y,s'_y} = \frac{\sum_n(v_{s_x}w_{s_y}z_{s'_y})^{(k)}\mathbf{y}_n}{\sum_n(v_{s_x}w_{s_y}z_{s'_y})^{(k)}}. \quad (17)$$

$$\mathbf{\Sigma}^{(k+1)}_{s_x,s_y,s'_y} = \frac{1}{\sum_n(v_{s_x}w_{s_y}z_{s'_y})^{(k)}} \times \sum_n(v_{s_x}w_{s_y}z_{s'_y})^{(k)}(\mathbf{y}_n-\mu^{(k)}_{s_x,s_y,s'_y})(\mathbf{y}_n-\mu^{(k)}_{s_x,s_y,s'_y})^t. \quad (18)$$

Once the cross-probability GMM parameters are estimated for each basic environment, $p(s_x|\mathbf{y}_t,e,s^e_y)$ can be obtained for MEMLIN with (8) as (19). Note that the time-independent assumption has been avoided.

$$p(s_x|\mathbf{y}_t,e,s^e_y) = \frac{p(\mathbf{y}_t|s_x,s^e_y)}{\sum_{s_x}p(\mathbf{y}_t|s_x,s^e_y)}. \quad (19)$$

For PD-MEMLIN, the cross-probability GMM parameters are estimated for each basic environment and phoneme independently, and $p(s^{ph}_x|\mathbf{y}_t,e,ph,s^{e,ph}_y)$ is computed in a similar way as (19).

5. RESULTS

To observe the performance of the cross-probability GMM proposed in a real, dynamic, and complex environment, a set of experiments were carried out using the Spanish SpeechDat Car database [7]. Seven basic environments were defined: car stopped, motor running (E1), town traffic, windows close and climatizer off (silent conditions) (E2), town traffic and noisy conditions: windows open and/or climatizer on (E3), low speed, rough road, and silent conditions (E4), low speed, rough road, and noisy conditions (E5), high speed, good road, and silent conditions (E6), and high speed, good road, and noisy conditions (E7).

The clean signals are recorded with a CLose talK (CLK) microphone (Shure SM-10A), and the noisy ones are recorded by a Hands-Free (HF) microphone placed on the ceiling in front of the driver (Peiker ME15/V520-1). The SNR range for CLK signals goes from 20 to 30 dB, and for HF ones goes from 5 to 20 dB.

For speech recognition, the feature vectors are composed of the 12 MFCCs, first and second derivatives and the delta energy, giving a final feature vector of 37 coefficients computed every 10 ms using a 25 ms Hamming window. On the other hand, in this work, the feature vector normalization methods are applied only to the 12 MFCCs and energy, whereas the derivatives are computed over the normalized static coefficients

The recognition task is isolated and continuous digits recognition. Three-state 16 Gaussian continuous density HMM to model the 25 Spanish phonemes and 2 silence models for long and interword silences are used in this task.

The Word Error Rate (WER) baseline results for each basic environment are presented in Table 1, where MWER is the Mean WER computed proportionally to the number of words in each basic environment. Cepstral mean normalization is applied to testing and training data. "Train" column refers to the signals used to obtain the corresponding acoustic HMMs: CLK if they are trained with all clean training utterances, and HF and if they are trained with all noisy ones. HF† indicates that specific acoustic HMMs for each basic environment are applied in the recognition task (environment match condition). "Test" column indicates which signals are used for recognition: clean, CLK, or noisy, HF.

Table 1 shows the effect of real car conditions, which increases the WER in all of the basic environments, (Train CLK, Test HF), concerning the rates for clean conditions, (Train CLK, Test CLK). When acoustic models are retrained using all basic environment signals, (Train HF) MWER decreases. Finally, 5.30% of MWER is obtained for environment match condition.

Figure 3 shows the mean improvement in WER (MIMP) in % for MEMLIN with Time-Independent cross-probability model (MEMLIN TI) and with Time-Dependent cross-probability GMM (MEMLIN TD). Also the results with SPLICE with Environmental Model Selection (SPLICE EMS) [4] are included. A 100% MIMP would be achieved when MWER equals the same of clean conditions. The cross-probability GMMs are composed by 2 Gaussians. It can be observed the important improvement of MEMLIN TD concern-

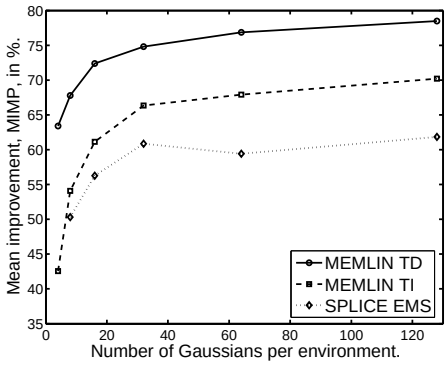


Figure 3. Mean improvement in WER, MIMP, in % for MEMLIN with Time-Dependent cross-probability model, MEMLIN TD, MEMLIN with Time-Dependent GMM cross-probability model, MEMLIN TI, and SPLICE with Environmental Model Selection, SPLICE EMS.

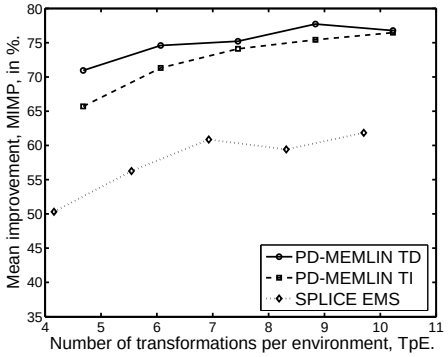


Figure 4. Mean improvement in WER, MIMP, in % for PD-MEMLIN with Time-Dependent cross-probability model, PD-MEMLIN TD, PD-MEMLIN with Time-Dependent GMM cross-probability model, PD-MEMLIN TI, and SPLICE with Environmental Model Selection, SPLICE EMS.

ing MEMLIN TI: from 42.55% to 63.38% with 4 Gaussians per basic environment and from 70.58% to 78.47% with 128 Gaussians.

On the other hand, the mean improvement in WER for PD-MEMLIN with Time-Independent cross-probability model (PD-MEMLIN TI) and with Time-Dependent cross-probability GMM (PD-MEMLIN TD) are depicted in Figure 4. The results obtained with SPLICE with Environmental Model Selection (SPLICE EMS) are also included to compare. The cross-probability GMMs are composed by 2 Gaussians for each pair of clean and noisy Gaussians of the same phoneme. To make a fair comparison between the methods, the results have been plotted as a function of the number of Transformations per basic Environment, TpE, which each method has to compute for each frame in normalization, in log_{10}

$$TpE = log_{10}(n_{s_y^{ph}} n_{s_x^{ph}} n_{ph}), \quad (20)$$

where $n_{s_y^{ph}}$ and $n_{s_x^{ph}}$ are the number of noisy and clean model Gaussians for ph phoneme, respectively, and n_{ph} is the number of phonemes ($n_{ph} = 1$, for SPLICE EMS).

It can be observed a slight improvement of PD-MEMLIN TD concerning PD-MEMLIN TI when TpE is higher: from 65.71% to 70.94% with 2 Gaussians per phoneme, and a bigger improvement when the phonemes are modelled with few Gaussians (lower TeP): from 75.43% to 77.72% with 16 Gaussians. Note that the critical point in PD-MEMLIN is not only the cross-probability model but also the probability of the phoneme, given the noisy feature vector and the environment, $p(ph|\mathbf{y}_t, e)$. So, there is still one term that needs to be improved in PD-MEMLIN.

Although the number of Gaussians to model the basic environments could be the same for MEMLIN TI and MEMLIN TD or for PD-MEMLIN TI and PD-MEMLIN TD, the computing time is not the same. To reduce it, only the cross-probability GMMs of the most probable pairs of Gaussians can be computed in normalization. In this case, for each noisy feature vector, the most probable PHonemes and Noisy model Gaussians (♯PH and ♯NG) can be obtained with (3) and (4), and for each one, the corresponding most probable Clean model Gaussians (♯CG) are obtained with (6) (No phoneme dependence for MEMLIN).

Table 2 shows the results for MEMLIN TD for different ♯NG and ♯CG. In all cases the cross-probability GMMs are composed by 2 Gaussians.

	♯NG	♯CG	MWER	MIMP
MEMLIN TD 4-4	4	4	7.04	63.40
MEMLIN TD 8-8	4	4	6.87	64.40
MEMLIN TD 16-16	8	8	5.67	72.87
MEMLIN TD 32-32	8	8	5.62	73.23
MEMLIN TD 64-64	16	16	5.44	74.46
MEMLIN TD 128-128	32	32	5.11	76.77

Table 2. Mean WER (MWER) and mean improvement in WER (MIMP) in % for MEMLIN TD when different Gaussians of cross-probability GMM are computed.

Table 3 shows the results for PD-MEMLIN TD for different ♯PH, ♯NG and ♯CG. In all cases the cross-probability GMMs are composed by 2 Gaussians. It can be observed that in MEMLIN TD and PD-MEMLIN TD it is not necessary to compute the cross-probability for all the Gaussians to obtain satisfactory results due to all of them has not the same importance.

6. CONCLUSIONS

In this paper we have presented an approach of MEMLIN and PD-MEMLIN where the cross-probability model is estimated by modelling the noisy feature vectors associated to each pair of Gaussians from the clean and the noisy basic environment spaces with a GMM. MEMLIN obtains an improvement in WER of 70.21% with 128 Gaussians per environment, whereas MEMLIN with cross-probability GMM reaches 78.47% for the same number of Gaussians to model each basic environment. On the other hand, PD-MEMLIN with 16 Gaussians per phoneme obtains an improvement in WER of 75.43%, whereas PD-MEMLIN with cross-probability GMM reaches 77.72%. Since the computing cost for the proposed approach is higher, an alternative is considered: only the cross-probability GMM of the most probable pair of Gaussians are computed. So, only with the 1024 most probable pair of Gaussians, an improvement of 76.77% is obtained, when 128 Gaussians per basic environment are used in MEMLIN; and if the cross-probability model GMM is computed over the 8125 most probable pair of Gaussians, an improvement of 77.36% is obtained in PD-MEMLIN when each phoneme is modelled wih 32 Gaussians.

	♯PH	♯NG	♯CG	MWER	MIMP
PD-MEMLIN TD 2-2	8	2	2	6.00	70.58
PD-MEMLIN TD 4-4	8	4	4	5.58	73.49
PD-MEMLIN TD 8-8	8	6	6	5.39	74.82
PD-MEMLIN TD 16-16	13	12	12	5.04	77.25
PD-MEMLIN TD 32-32	13	25	25	5.02	77.36

Table 3. Mean WER (MWER) and mean improvement in WER (MIMP) in % for PD-MEMLIN when different phonemes and Gaussians of cross-probability GMM are computed.

7. BIBLIOGRAPHY

[1] A. Sankar and C. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," vol. 4, pp. 190– 202, May 1996.

[2] Leonardo Neumeyer and Mitchel Weintraub, "Robust Speech Recognition in Noise Using Adaptation and Mapping Techniques," Detroit, USA, May 1995, vol. 1, pp. 141–144.

[3] Richard M. Stern, Bhiksha Raj, and Pedro J. Moreno, "Compensation for environmental degradation in automatic speech recognition," in *ESCA Tutorial and Research Workshop on Robust Speech Recognition for Unknown Communication Channels*. Pont-au-Mousson, France, Apr. 1997, pp. 33–42.

[4] J. Droppo, L. Deng, and A. Acero, "Evaluation of the SPLICE algorithm on the AURORA2 database," in *Proceedings of Eurospeech*. Aalborg, Denmark, 2001, pp. 217–220.

[5] L. Buera, E. Lleida, A. Miguel, and A. Ortega, "Multi-environment models based linear normalization for robust speech recognition in car conditions," in *Proceedings of ICASSP*. Montreal, Canada, May 2004, vol. 1, pp. 1013–1016.

[6] L. Buera, E. Lleida, A. Miguel, and A. Ortega, "Robust speech recognition in cars using phoneme dependent multi-environment linear normalization," in *Proceedings of Interspeech*. Lisbon, Portugal, 2005, pp. 381–384.

[7] Asuncion Moreno, Borge Lindberg, Christoph Draxler, Gael Richard, Khalid Choukri, Stephan Euler, and Jeffrey Allen, "Speechdat-car. a large speech database for automotive environments," in *Proceedings of LREC*. Athens, Greece, 2000, vol. 2, pp. 895–900.

[8] A. P. Dempster, N.P. Laird, and D.B. Rubin, "Maximum likelihood from imcomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 9, no. 1, pp. 1–37, 1977.

DETECCIÓN DE PITCH EN CONDICIONES ADVERSAS

Ibon Saratxaga, Iker Luengo, Eva Navas, Inmaculada Hernáez, Jon Sánchez, Iñaki Sainz

Aholab – Escuela Técnica Superior de Ingeniería.
Universidad del País Vasco – Euskal Herriko Unibertsitatea
Urkijo zum. z/g 48013 Bilbo
ibon.saratxaga@ehu.es, ikerl@bips.bi.ehu.es, eva.navas@ehu.es, inma.hernaez@ehu.es, jon.sanchez@ehu.es, inaki@bips.bi.ehu.es

RESUMEN

La necesidad de herramientas de detección del pitch lo suficientemente robustas para funcionar en entornos ruidosos se ha visto acrecentada en los últimos tiempos debido a la aparición de nuevos sistemas de codificación de voz, reconocimiento, conversión etc. En este trabajo se presenta un algoritmo de detección basado en los coeficientes cepstrales combinados con un algoritmo de Viterbi, que mantiene cierta robustez en condiciones ruidosas. Este algoritmo ha sido evaluado mediante una base de datos preparada al efecto, y se detallan los resultados obtenidos comparándolos con los de otros algoritmos de uso general.

1. INTRODUCCIÓN

La detección y el marcado de pitch han sido necesidades importantes desde los inicios de la investigación en el área de voz. Si tradicionalmente la detección de pitch había sido esencial para el estudio y modelado de la entonación, más recientemente han surgido nuevas aplicaciones como el reconocimiento de emociones [1], el reconocimiento de lenguajes tonales [2], la conversión de voz [3], el reconocimiento [4] o la verificación de locutor [5] que han reavivado los requerimientos de sistemas de detección y marcado de pitch robustos que sean capaces de trabajar con señales obtenidas fuera de los estudios de grabación, por canales más ruidosos.

Con estos antecedentes, un grupo de trabajo enmarcado en el European Center of Excellence on Speech Synthesis (ECESS) (www.ecess.eu), promovió entre sus miembros, una campaña de evaluación de algoritmos de detección de pitch, tanto para señales de buena calidad, como para aquellas con una relación SNR peor. Para ello se preparó una base de datos de voz de referencia, marcada y revisada manualmente, para su utilización en la evaluación de algoritmos de marcado y detección de pitch [6]. Nuestro grupo de trabajo participó en esta evaluación con una versión mejorada de uno de los algoritmos de detección de pitch de que disponíamos. Los resultados se presentaron en la reunión del ECESS en Maribor (Eslovenia) el 5 de julio de 2006.

Este módulo de detección de pitch es el que se describe en el presente artículo. Los diferentes bloques que componen el mismo se detallan en el siguiente apartado. Después se describen las características de la evaluación y se da cuenta de los resultados obtenidos, incluyendo una somera comparación de éstos con otros resultados publicados. El documento finaliza con unas breves conclusiones y posibles mejoras.

2. MÓDULO DE DETECCIÓN DE PITCH

El módulo de detección de pitch (MDP) se encarga de determinar los valores del pitch de la señal de voz en cada momento, generando así los puntos de la curva de la evolución del pitch en el tiempo. Para obtener esta curva se utiliza un algoritmo de determinación de pitch basado en los valores de los coeficientes cepstrales, con selección de los mismos mediante el algoritmo de Viterbi. La curva así obtenida es postprocesada junto con la información sobre el carácter sordo o sonoro de cada trama, en un bloque de suavizado.

La estructura del módulo de detección de pitch se esquematiza en la figura 1. La única entrada que el módulo requiere es la señal de voz. Esto es importante, puesto que permite obtener las curvas de pitch directamente, sin necesidad de procesado previo para obtener alguna otra información (como podría ser la segmentación fonética por ejemplo), lo que posibilita calcular valores de pitch de cualquier señal desconocida.

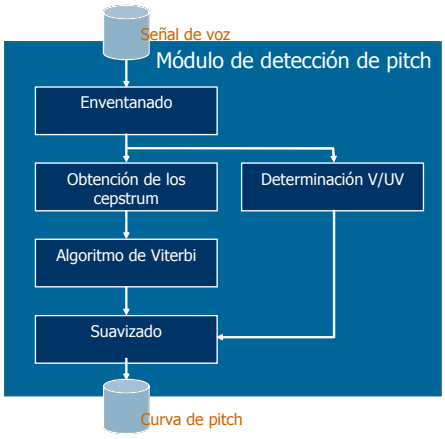


Figura 1. *Diagrama de bloques del MDP.*

La salida del algoritmo es un fichero con los valores de pitch a la frecuencia de muestreo establecida por el usuario, en formato PCM. Se detallan a continuación los pasos que se siguen para la determinación de estos valores.

2.1. Enventanado

El análisis de la periodicidad o aperiodicidad de la señal en cada momento debe realizarse sobre un fragmento de señal significativo. Por ello es necesario enventanar la señal tomando un fragmento lo suficientemente largo como para que la periodicidad sea detectable, pero lo suficientemente corto como para que no se pierda resolución temporal.

Para que la detección de periodicidad sea correcta se requiere que el enventanado contemple al menos dos periodos de pitch. Con esta idea, el tamaño de la ventana que se empleará queda fijado por un parámetro que es el pitch mínimo esperable en la señal. La ventana que se tomará será constante para todo el análisis y su longitud será de dos veces el periodo de pitch mínimo.

La ventana es de Hamming con lo que en realidad, debido a la atenuación de la misma en los bordes, la longitud efectiva de la trama será algo menor que dos periodos del pitch mínimo.

2.2. Determinación Sonoro/Sordo

La determinación de si una trama es sorda o sonora se hace evaluando la razón entre la potencia de la misma y su tasa de cruces por cero (Zero Crossing Rate, ZCR) también normalizada a la longitud de la trama. Para ello se calcula por un lado la potencia de la trama, como:

$$P_f = \frac{1}{N}\sum_{i=0}^{N-1}\left|x[i]\right|^2$$

Donde x[i] son cada una de las N muestras de la trama (incluye el efecto del enventanado por lo tanto).

La ZCR normalizada se basa en contar el número de veces que la señal cruza por cero en toda la longitud de la trama y dividir también entre el número de muestras de la trama.

Con este ratio, se trata de reforzar el efecto de dos medidas independientes relacionadas con el carácter sordo o sonoro de la señal de voz. En efecto, la tasa de cruces por cero es en general mayor para los sonidos sordos, que suelen tener componentes a más altas frecuencias. La energía o la potencia, por el contrario tiende a ser menor para los sonidos sordos. Por ello, calcular el ratio energía entre ZCR hace que ambos efectos se refuercen: para tramas sordas el valor será pequeño, y será mucho mayor (4 órdenes de magnitud aprox.) cuando la trama sea sonora.

2.3. Cálculo de los coeficientes cepstrales

El primer paso para determinar el valor de pitch de la trama es calcular sus coeficientes cepstrales. Recordemos que los cepstrum se calculan tomando logaritmos sobre la transformada de fourier de la señal y luego aplicando su transformada inversa:

$$\hat{x}[n] = TF^{-1}\{\log(X(\Omega))\}$$
$$\text{con } X(\Omega) = TF\{x[n]\}$$

Los valores de pitch están limitados fisiológicamente, por lo cual no es necesario examinar todos los coeficientes cepstrum. Por ello en primer lugar se calcula un intervalo limitado por las frecuencias mínima f_{min} y máxima f_{max} entre las que se considera que se moverá el pitch y sólo se buscarán los máximos entre los coeficientes de ese intervalo.

El siguiente paso, una vez definido el conjunto de coeficientes en los que se centrará la búsqueda, es encontrar cuáles son los máximos. Para ello, se normalizan todos los coeficientes al valor medio del conjunto de los coeficientes de la trama:

$$c'_i = \frac{c_i}{\bar{c}} \quad / \quad \bar{c} = \frac{1}{i_{\max} - i_{\min} + 1}\sum_{i=i_{\min}}^{i_{\max}} c_i$$

El uso de cepstrum normalizados permite independizar sus valores de factores como la energía de la señal en cada punto, produciendo una escala coherente a lo largo de toda la señal. Esto permitirá que se pueda definir un umbral de tal forma que si un coeficiente cepstrum es n veces mayor que la media de la trama, entonces se considerará que la trama es sonora, mientras que por el contrario, si el valor no supera ese umbral, entonces se considerará que la trama es sorda.

En cualquier caso, en este bloque lo único que se hace es calcular los cepstrum, normalizarlos respecto de la media de la trama y seleccionar los M coeficientes mayores, futuros candidatos para el algoritmo de Viterbi. La decisión final sobre si el tramo es sordo o sonoro se tomará en la etapa de Viterbi. Además de los M mayores coeficientes se añade otro más, la "no-frecuencia", correspondiente a la decisión de que la trama fuera sorda.

2.4. Algoritmo de Viterbi

Una vez que se han calculado el conjunto de M+1 candidatos para todas las tramas que forman la señal, se deben elegir aquéllos que formen una mejor curva de pitch. Para ello, como ya se ha comentado, se utiliza el algoritmo de Viterbi, que se basa en escoger aquellos valores que hagan que se minimice la suma de dos funciones de coste: el coste local (el coste de seleccionar un candidato por sí mismo, independientemente de los candidatos adyacentes); y el coste de transición (el coste de seleccionar un candidato teniendo en cuenta el candidato seleccionado para la trama anterior).

Estos costes se definen de forma que reflejen las características de la curva de pitch. El coste local se define mediante dos términos. Para el primero, se tiene en cuenta que el candidato más probable para la frecuencia fundamental de una trama es aquél cuyo coeficiente cepstrum asociado tiene el máximo valor.

Por ello, en términos de coste se define una función inversamente proporcional al valor del coeficiente cepstral.

El segundo término del coste local se fundamenta en que, para considerar la trama como sonora, el valor de los cepstrum debe superar un umbral respecto al valor medio de los cepstrum de su trama. Si no es así, probablemente la trama será sorda y el máximo será aleatorio. Así, se define un coste fijo en el caso de que se viole este criterio, es decir, que se considere una trama sonora a pesar de que no supere el umbral, o a la inversa, sorda aún superándolo.

El coste de transición se forma también mediante dos términos. El primero de ellos se basa en el criterio de que la parte sonora de la curva de pitch debe ser continua y sin saltos bruscos, pues estos se deberían probablemente a que se ha seleccionado un armónico o sub-armónico del valor correcto de pitch. Así pues se define un coste proporcional a la relación entre las frecuencias de pitch de una trama respecto a la anterior. Cuanto mayor sea esta diferencia mayor será el coste.

Finalmente, se define un segundo término de coste de transición, para recoger el criterio de que es muy poco probable que ocurran transiciones rápidas entre regiones sordas y sonoras. Es decir, es poco probable que pasemos de una región sorda a una sonora y rápidamente a otra sorda. En este caso, es muy probable que una región sonora (o sorda) suficientemente pequeña sea en realidad un error de decisión sordo-sonoro. Así se define un coste fijo que penaliza las transiciones sordo-sonoro y sonoro-sordo.

Con estas funciones el algoritmo de Viterbi encuentra la curva que menor coste acumulado arroja. Sin embargo, esta curva podría contener valores espurios de pitch y además no se ha tenido en cuenta la información proveniente del detector sordo/sonoro. De esto se encarga el último bloque.

2.5. Suavizado

La curva a la salida del algoritmo de Viterbi puede contener valores erróneos sobre todo en puntos en los que la periodicidad es débil. En estos puntos los cepstrum habrán estado cerca del umbral de detección y puede que se haya tomado un máximo erróneo. Para estos casos el algoritmo de detección sordo o sonoro proporciona información no relacionada con los cepstrum, que puede ser útil para dilucidar el carácter de la trama.

Por otro lado, es conocido [7] que el pitch de un locutor puede ser modelado por una distribución log-normal. Aquellos valores que se aparten en demasía de la media de esta distribución pueden considerarse como errores en la detección de pitch. Cuanto más alejados estén los valores de pitch de la media, más seguro será que son erróneos.

Por ello, existe un parámetro N que indica el umbral a partir del cual se considerará que un valor está fuera de rango. Un valor de la curva de pitch se considera fuera de rango si está alejado del valor medio N desviaciones típicas de la distribución log-normal.

Para aplicar este criterio, lo primero que se debe hacer es calcular la media y la desviación típica de la distribución log-normal que modela el pitch de la curva obtenida del Viterbi. Para ello se calcula el logaritmo de todos los valores de pitch de la curva y con estos valores escalados se obtiene su media y su desviación típica.

El bloque de suavizado comprueba en primer lugar si el valor está o no fuera de rango, y si está fuera de rango lo elimina, marcando la trama como sorda. En un paso posterior, se coteja el carácter sordo o sonoro de la trama según sale del Viterbi tras eliminar los valores fuera de rango con el resultado del detector sordo-sonoro. Así si el detector sordo sonoro indica que una trama es sonora y Viterbi dice que es sorda, se da preferencia al detector de sordo sonoro y se marca como sonora interpolando un valor de pitch para la misma, tomando para ello las muestras de pitch más cercanas e interpolando linealmente.

Si el detector sordo-sonoro dice que una trama es sorda y el algoritmo de Viterbi ha propuesto un valor de pitch y éste está dentro del rango, entonces tiene preferencia el algoritmo de Viterbi.

3. EVALUACIÓN

La evaluación de la eficiencia de estas herramientas requiere comparar sus resultados para un conjunto de señales en las que se conozca sus valores de pitch. El consorcio ECESS promovió una campaña de evaluación de herramientas para detección y marcado de pitch, y proporcionó una base de datos marcada manualmente. A continuación describiremos las características de esa base de datos para pasar luego a detallar los resultados de las evaluaciones.

3.1. Base de datos de evaluación

La base de datos que se ha utilizado para la evaluación consiste en un subconjunto de la base de datos SPEECON Spanish. Las bases de datos SPEECON fueron grabadas siguiendo las especificaciones del proyecto SPEECON de la Comisión Europea [8], y buscaba obtener grabaciones de señales de voz en diferentes entornos acústicos para su uso en reconocimiento, básicamente.

Las señales se adquirieron simultáneamente por cuatro canales utilizando diferentes micrófonos, en diferentes ambientes acústicos (coches, oficinas, lugares públicos, etc.). Así, el primero de los canales (C0) se grabó con un micrófono acoplado a unos auriculares, el canal C1 se adquirió con un micrófono Lavalier o de corbata, el C2 con un micrófono direccional situado a 1 metro del locutor y el C3 se grabó con uno omnidireccional situado a 2-3 metros del locutor.

En estas condiciones las relaciones de señal a ruido de los diferentes canales son muy diferentes. Así, el canal C0 tiene SNR's de unos 30 dB, y en el otro

extremo el canal C3, con un micro omnidireccional, tiene relaciones mucho más bajas, del orden de 0 dB's.

Para preparar la base de datos de referencia [6] se escogieron frases de 60 locutores, 30 hombres y 30 mujeres, desde 19 a 79 años, con grabaciones de 1 minuto por locutor, lo que hace un total de 60 minutos por cada canal. Las frases provienen de diferentes tipos de corpus desde el punto de vista de su dominio semántico.

Las señales del canal C0, las más libres de ruido, se marcaron a periodo de pitch mediante un sistema automático. Las marcas del sistema automático fueron revisadas manualmente en su totalidad, corrigiéndose cuando fue necesario. De estas marcas corregidas se derivaron los valores del pitch también para el canal 0, obteniéndose valores cada milisegundo.

Dado que las grabaciones se realizaron simultáneamente para todos los canales, las marcas y los valores de pitch serán idénticos para todos ellos, con la única diferencia que estarán más o menos retardados debido a las diferentes distancias que tiene que recorrer el sonido para llegar a cada micrófono. Con el fin de utilizar las mismas referencias para todos los canales, se compensaron estos retardos realineando las grabaciones, mediante correlación cruzada de cada canal con el C0.

3.2. Criterios de evaluación

Una vez descritas las características de la base de datos de referencia, veremos en este apartado cuáles son las magnitudes que nos van a permitir estudiar el rendimiento de nuestro módulo de detección de pitch. Para ello se definieron algunas medidas habituales en la literatura sobre estos temas [9].

- Valores erróneos por exceso y por defecto: Estas dos tasas de error miden el porcentaje de los valores de pitch correspondientes a segmentos sonoros cuyo valor supera, o queda por debajo, en más de un 20% el valor de pitch correcto. A estas medidas se les llama en inglés Gross Error High (GEH) y Gross Error Low (GEL), respectivamente, y se suelen visualizar acumulados, para dar una idea de la tasa de error total. Nótese que las tramas sordas o silenciosas (que no tienen ningún valor de pitch) no computan para este error.
- Porcentaje de error de tramas sonoras y de tramas sordas: El porcentaje de error de tramas sonoras (voiced error o VE) mide el porcentaje de las tramas sonoras que han sido erróneamente clasificadas como sordas, respecto del total de tramas sonoras de la BD. El porcentaje de error de tramas sordas (unvoiced error o UE) es, análogamente, el porcentaje de las tramas sordas, erróneamente clasificadas como sonoras, respecto al total de tramas sordas.
- Diferencias en la media y la desviación estándar: La última pareja de magnitudes que mediremos será la diferencia entre la media y la desviación estándar de todos los valores de pitch estimados y las de todos los datos de referencia.

3.3. Resultados

Los resultados que se muestran a continuación, buscan un equilibrio que dé resultados aceptables en todas las categorías y canales, lo que significa que a lo largo de las pruebas se vieron configuraciones de parámetros que hubieran permitido mejorar un canal o criterio específico. Por ello, los valores que aquí se proponen para los parámetros pueden mejorarse para adaptarse a las características de las señales que se vayan a analizar en un problema concreto.

Los resultados así obtenidos son los mostrados en la tabla 1:

	C0	C1	C2	C3
VE(%)	9,94	22,62	28,41	35,62
UE(%)	7,44	7,35	6,93	7,35
GEH(%)	0,65	0,31	0,57	0,97
GEL(%)	1,99	2,38	2,45	2,40
AbsMeanDiff(Hz)	0,54	1,87	7,57	11,92
AbsStdDiff(Hz)	1,45	4,46	4,86	6,03

Tabla 1. *Resultados finales*

Para analizar estos resultados los dividiremos en tres grupos. Por un lado tenemos los dos indicadores de error de trama sorda/sonora, UE y VE (figura 2). Como era previsible, el bloque específico de detección por Pot/ZCR era muy sensible al ruido, por lo que se deshabilitó para los canales ruidosos. De esta forma la discriminación sordo/sonoro para la trama se obtiene de los cepstrum en C1, C2 y C3. El resto de los ajustes de parámetros son comunes para todos los experimentos y canales.

En cualquier caso, la precisión de la clasificación se degrada rápidamente con el ruido del canal, y los resultados no son muy espectaculares, aunque, como se verá en los datos comparativos, la clasificación obtenida es de las mejores de entre los algoritmos comparados.

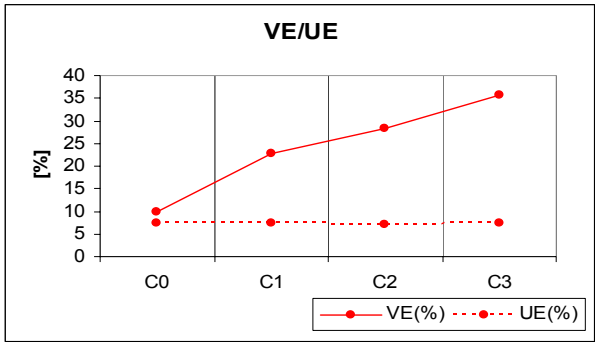


Figura 2 *Evolución VE/UE respecto del canal*

Es necesario hacer notar que la alta frecuencia del análisis, con tramas de 1 ms. hace que surjan una serie de cuestiones sobre los criterios para clasificar una trama como sonora. Debe tenerse en cuenta que 1ms. es un periodo entre 4 y 10 veces menor a un periodo de

pitch, por lo que cabe preguntarse al principio o al final de un fragmento periódico en qué punto del primer periodo de pitch que aparece (o en el último en que se desvanece), se debe comenzar a (o dejar de) considerar periódica la trama.

La siguiente pareja de datos es la formada por los errores en los valores de pitch por exceso (GEH) y por defecto (GEL). Los resultados en estas dos magnitudes son buenos, y lo son para todos los canales, empeorando sólo ligeramente al aumentar el ruido: su suma se mantiene siempre alrededor del 3% (ver figura 3).

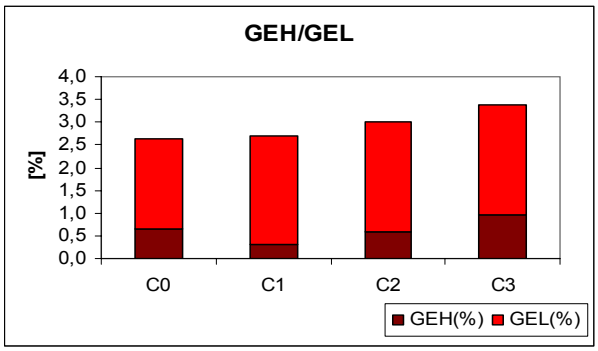


Figura 3. *Evolución GEH/GEL respecto del canal*

Las dos últimas medidas, la diferencia entre las medias y las desviaciones típicas de los valores de pitch de referencia y estimados, han resultado más afectadas por el ruido del canal, como se puede ver en la figura 4.

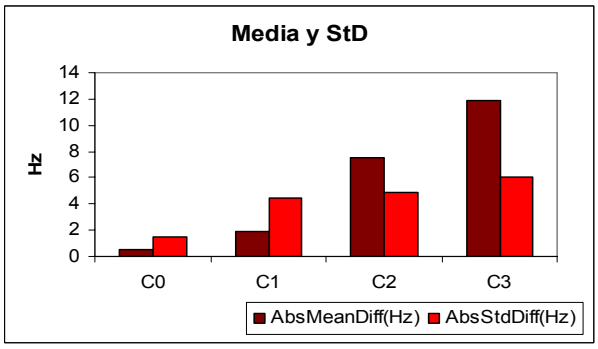


Figura 4. *Diferencias de media y desviación típica por canal*

3.4. Resultados comparativos

Para tener elementos de comparación contra los que validar la bondad de los resultados que estábamos obteniendo durante los experimentos, llevamos a cabo los mismos experimentos utilizando el algoritmo de autocorrelaciones del Praat [10], en su configuración por defecto. Este método consiste en calcular la autocorrelación de una trama de la señal con más de un periodo de pitch, lograda mediante enventanado. En la versión implementada en el Praat la autocorrelación está dividida por la autocorrelación de la ventana, para evitar el efecto distorsionador que ésta pudiera tener, según se describe en [11]. Además en [6] se habían publicado los resultados de otros dos algoritmos: el de los autores (KOT), basado en el cálculo de la transformada Hilbert de los residuos LPC; y el de Goncharoff [12], (GON), basado en la detección de la periodicidad en la energía de la señal a corto plazo, con programación dinámica para seleccionar los valores correctos.

Se muestran a continuación los resultados obtenidos por nuestro módulo comparándolo con el resto de los valores publicados.

3.4.1. Errores acumulados de trama sorda y sonora

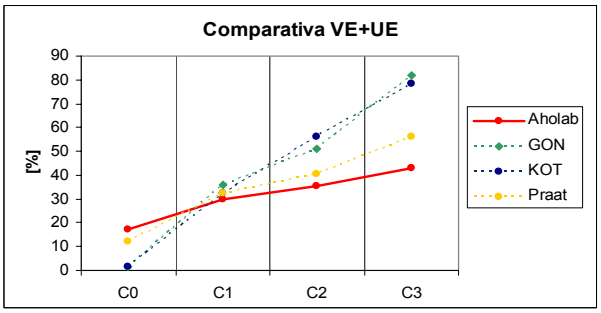


Figura 5. *Comparativa VE+UE, todos los canales*

Se ha comentado ya la relativa debilidad de los resultados de estos errores para el canal 0, sin ruido. Los algoritmos específicos en el dominio temporal, para la detección sordo-sonoro dan mejores resultados para este canal. Sin embargo, para los canales más ruidosos, los algoritmos temporales basados en ZCR empeoran rápidamente, mientras que la detección basada en cepstrum es considerablemente mejor (figura 5).

3.4.2. Errores acumulados de valor de pitch por defecto y por exceso

Los resultados de los errores en el valor del pitch estimado son en todos los casos los mejores de la comparativa (figura 6). Como hemos comentado son además buenos para todos los canales, por lo que su diferencia respecto al resto de los métodos considerados aumenta a medida que se incrementa el ruido del canal.

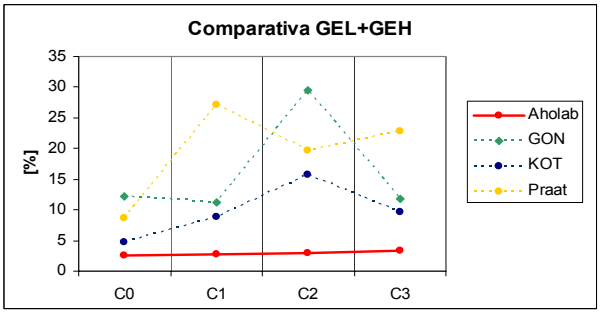


Figura 6. *GEL+GEH, todos los canales*

3.4.3. Diferencias en los estadísticos

Las diferencias en las medias comienzan siendo prácticamente despreciables para el canal 0 y van incrementándose con el ruido (figura 7). En nuestro caso el incremento es, sin embargo, menor que en los restantes algoritmos.

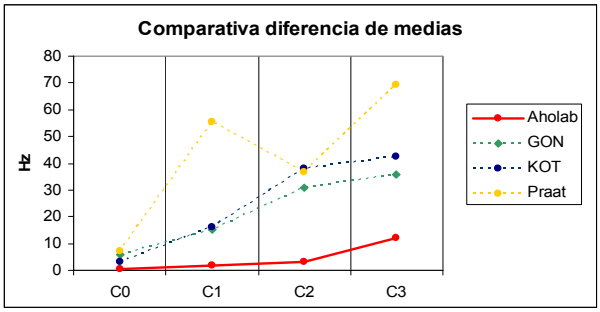


Figura 7. *Diferencia medias, todos los canales*

Respecto a la desviación estándar (figura 8) los valores se van incrementando en todos los casos a medida que sube el ruido, aunque es de resaltar la suave progresión que tiene en nuestro caso, lo que hace que obtengamos mejores resultados de nuevo para los canales ruidosos.

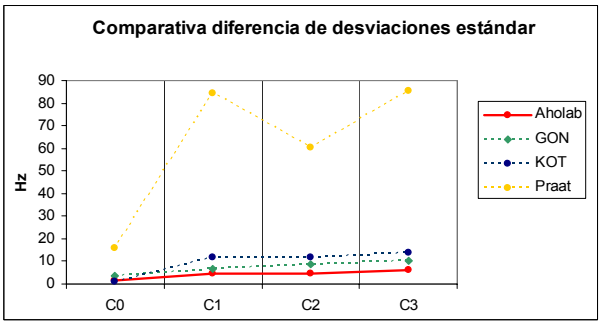


Figura 8. *Dif. desviación estándar, todos los canales*

Queda finalmente comentar en estos dos criterios de evaluación los erráticos resultados del algoritmo de Praat para los canales con ruido. Sin duda su aplicación para este tipo de señales requerirá ajustes en los parámetros por defecto, que no olvidemos que son los que se han utilizado en esta comparativa.

4. CONCLUSIONES

Las conclusiones de la evaluación del algoritmo de detección propuesto son muy prometedoras. El algoritmo ofrece unos resultados buenos en condiciones de bajo nivel de ruido; y es notoriamente robusto en condiciones ruidosas, ofreciendo el mejor rendimiento entre los algoritmos analizados.

Además el algoritmo participó en la campaña de evaluación del ECESS, con muy buenos resultados. Nuestra herramienta de detección de pitch obtuvo los mejores resultados en 19 de los 32 puntos de evaluación considerando todos los canales. Fue la mejor de las presentadas en la exactitud de los valores de pitch, para todos los canales; y obtuvo los mejores resultados en casi todas las medidas en los canales ruidosos.

Como continuación a estos trabajos, se pretenden desarrollar algunas ideas sobre posibles mejoras:

- Estudio de técnicas de detección del carácter sordo o sonoro de las tramas, que es la parte más débil del algoritmo: autocorrelación, filtrado paso bajo de ruido,…
- Inclusión de la información del bloque de detección sordo/sonoro en las funciones de coste del algoritmo de Viterbi.
- Comparación con otros algoritmos de detección y evaluación con otras bases de datos estándar.

5. BIBLIOGRAFÍA

[1] E. Navas, I. Hernáez, I. Luengo, J. Sánchez, I. Saratxaga. "Analysis of the Suitability of Common Corpora for Emotional Speech Modeling in Standard Basque". Lecture Notes in Artificial Intelligence, LNAI 3658, pp. 265-272, 2005.

[2] H.C.H. Huang, F. Seide. "Pitch tracking and tone features for Mandarin speech recognition". Procs. ICASSP 2000. Estambul. pp. 1523 - 1526 vol.3. Junio 2000.

[3] H. Ney, D. Suendermann, A. Bonafonte, H. Hoege. "A first step towards text-independent voice conversion". Procs. INTERSPEECH 2004, Jeju, Corea. pp. 1173-1176. Octubre 2004.

[4] S. Kim, T. Eriksson, H.G. Kang, D.H. Youn. "Pitch Synchronous Feature Extraction Method for Speaker Recognition". Procs. ICASSP 2004. Montreal. pp. 405-408. May 2004.

[5] I. Luengo, E. Navas, I. Hernáez. "Effectiveness of Short-Term Prosodic Features for Speaker Verification". Procs. The Fundamentals of Verbal and Non-verbal Communication and the Biometrical Issue. Vietri sul Mare, Italia. Septiembre 2006

[6] B. Kotnik, H. Höge, Z. Kacic. "Evaluation of Pitch Detection Algorithms in Adverse Conditions". Procs. 3rd International Conference on Speech Prosody, Dresden, Alemania, pp. 149-152, Mayo 2006.

[7] M. K. Sonmez, L. Heck, M. Weintraub, E. Shriberg. "A lognormal tied mixture model of pitch for prosody-based speaker recognition". Procs. EUROSPEECH '97. Rodas, Grecia. vol. 3, pp. 1391-1394. Septiembre 1997.

[8] D.J. Iskra et al. "SPEECON - Speech Databases for Consumer Devices: Database Specification and Validation". Procs. LREC'2002. Las Palmas de Gran Canaria. pp. 329-333. Junio 2002.

[9] X. Sun. "Pitch Determination and Voice Quality Analysis Using Subharmonic-To-Harmonic Ratio". Procs. ICASSP 2002. Orlando, EEUU. pp. 333-336. Mayo 2002.

[10] P. Boersma, D. Weenink. Praat: doing phonetics by computer (Version 4.3) [Computer program]. Retrieved from http://www.praat.org/

[11] P. Boersma. "Accurate short term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound". Procs. Institute of Phonetic Sciences 17. Univ. Amsterdam. pp. 97-110. 1993.

[12] V. Goncharoff, P. Gries. "An Algorithm for Accurately Marking Pitch Pulses in Speech Signals". IASTED International conference SIP '98. Nevada, USA. 1998.

SYSTEMS FOR ROBUST SPEECH ACTIVITY DETECTION AND THEIR RESULTS WITH THE RT05 AND RT06 EVALUATION TESTS

Dušan Macho, Andrey Temko, and Climent Nadeu

TALP Research Center, UPC, Barcelona, Spain
{dusan,temko,climent}@talp.upc.edu

ABSTRACT

Robust Speech Activity Detection (SAD) systems are required in smart-room environments due to the presence of noises and reverberation. In this work, a previous SAD system, based on LDA-extracted features and a decision tree classifier, has been modified in terms of both feature extraction and classification to significantly improve its performance. New features based on the low- and high-frequency energy dynamics, and classifiers based on SVM and GMM have been investigated. In particular, a specific training process has been developed for the SVM case to cope with the problems of that classifier in our application. The resulting SAD systems have been trained with a subset of the SPEECON database. Tested in realistic conditions with the meeting databases from the NIST RT05 and RT06 evaluations, they have shown large improvements in speech detection performance.

1. INTRODUCTION

Detecting the presence of speech is a key objective in speech-related technologies. In fact, Speech Activity Detection (SAD) usually allows an increase of recognition rate in automatic speech or speaker recognition, and it is also required in both speech/speaker recognition and speech coding to save computational resources (and batteries) in the devices where the processing of non-speech events is not needed. Also, as many speech enhancement techniques require a proper estimate of noise characteristics, the reliable detection of non-speech portions of signal is needed. On the other hand, SAD may boost the performance measures of other technologies belonging to audio scene analysis, like speaker localization or acoustic event detection. Last but not least in perceptive interface technologies, the determination of speech activity in a room environment may be used to infer the type of activity that takes place in the room, or at a specific position of the room, given the coordinates of the microphones whose signals show the presence of speech utterances.

Our work, inserted in the CHIL (Computers in the Human Interaction Loop) project framework, assumes a meeting room environment, where audio acquisition is done in an unobtrusive way by a network of far-field microphones. In such a challenging environment, a high robustness of the SAD algorithm against environmental noises and reverberation is extremely important. On the other hand, the working scenarios require online implementations that can operate in real time and only a given maximum latency is accepted. Consequently, segmentation algorithms that use the entire recorded file must be avoided.

In a previous work at our laboratory, we proposed a SAD algorithm [1] that assumed this kind of environment and working conditions. It was compared with other previously reported techniques using a subset of the SPEECON database [2]. The speech detection system was based on speech features that had already shown good robustness properties in automatic speech recognition: the Frequency-Filtered (FF) log spectral energies. The fact that these features are also used for speech recognition avoids the need to re-compute them for SAD when both tasks are being performed at the same time. The FF parameters were further processed by Linear Discriminant Analysis (LDA) to select only one feature per frame, and a Decision Tree (DT) classifier used a time sequence of these features to make the Speech/Non-Speech decision.

In this paper, further work is presented along that line. The already existing algorithm has been tested in more real conditions involving interactions of several persons in meetings and it has been modified to significantly improve its performance. We trained our SAD system with the previous subset of SPEECON and, without any additional tuning, we have used it to carry out tests with the meeting databases from the NIST Rich Transcription 2005 (RT05) evaluation. Both the usual NIST metrics and the ones used in CHIL for SAD have been used to compare performances. In order to improve the SAD results, we have considered two additional features which are measures of energy dynamics at low and high frequencies, respectively. Besides that, two alternative classifiers have been tested, which are based on Support Vector Machines (SVM) [7] and Gaussian Mixture Models (GMM) [9], respectively. The usual training algorithm of the SVM-based classifier has been improved in order to cope with two problems of that classifier in our application: the very large amount of training data and the particular characteristics of the NIST metric. Also, a variant of the GMM-based SAD system was used in the NIST RT06 evaluation campaign, and its results are reported in this paper.

The databases for training and testing are presented in Section 2. Section 3 describes the features and Section 4 is dedicated to the classifier training procedures. Experiments

and results are presented in Section 5, along with the improvements in the training of the SVM-based classifier.

2. DATABASES

For the classifier training, we used a portion of the office environment recordings from the *Spanish* language SPEECON database [2]. In total, 90 minutes of signal recorded by a far-field omni-directional microphone placed 2-3 meters in front of the speaker was used. The training material was well balanced in terms of the two classes of interest; it contained 49% of Speech and 51% of Non-Speech. The database sampling frequency was 16 kHz and the sample representation is 16 bits. Across all recordings, the audio signal uses about 50% of the available 16-bit dynamic range.

For the classifier test/development, we used the single distant microphone evaluation data from the NIST Rich Transcription 2005 (RT05) "conference room" meeting task [3]. It contains 10 extracts from 10 *English* language meetings recorded at 5 different sites. Each extract is about 12 minutes long. The proportion of Speech / Non-Speech is highly unbalanced, approximately 90% of all signal is Speech. The sampling frequency and sampling representation are the same as in the training data, 16 kHz and 16 bits, respectively. Some extracts, however, use only a small portion of the available dynamic range (less than 20%).

The Rich Transcription 2006 (RT06) test data set consists of two kinds of data, "confmtg" and "lectmtg". The confmtg data set is similar to the previously described RT05 data. The lectmtg data were collected from lectures and interactive seminars across the smart-rooms of different CHIL project partners.

The training and development/testing data are similar in a way that they are recorded in a closed environment using a far-field microphone, thus the recordings have a relatively low SNR due to the reverberation and the environmental noise. However, there are some differences that should be mentioned: different language (Spanish vs. English), different setup of the acquisition hardware, different Speech and Non-Speech proportion. Also, it is worth to mention that the main task, and thus the main attention, of the speaker in the training database was the recording itself, while in the test meeting/lecture database, the recording was secondary. As a consequence, the test/lecture database is more spontaneous, speakers speak not necessarily heading the microphone, and the data contain overlapped speech.

3. FEATURES

We investigate two kinds of features. The first feature set, based on Linear Discriminant Analysis (LDA) [9] of parameters that model spectra, extracts the information about the spectral shape of the acoustic signal from a short interval (approx. 70 ms). The second feature set focuses more on the dynamics of the signal along the time observing low- and high-frequency spectral components along a bit longer time interval (approx. 130 ms).

3.1. LDA Measure

The LDA measure, *ldam*, is based on Frequency Filtering (FF) features – a speech representation originally designed for ASR that showed higher robustness in noisy ASR tests than the usual mel-frequency cepstrum (MFCC) features (see e.g. [4]). The robustness issue is very important due to the low SNR of the recordings in our task.

The FF feature extraction scheme used in this work consists in calculating a log filter-bank energy vector of 16 bands for each signal frame (with frame length/shift = 30/10ms) and then applying a FIR filter with impulse response $h(k)=\{1, 0, -1\}$ on this vector along the frequency axis. The obtained static FF feature vector is accompanied with a short-time dynamic representation in form of delta (50ms) and delta-delta (70ms) features. In addition, the delta of the frame energy is also appended. The size of the FF representation (16+16+16+1=49) is reduced to a single scalar measure by applying LDA, a data-driven linear transformation designed to extract the principal components of the input data using a discriminative criterion. That single LDA measure, *ldam*, is computed by multiplying the FF feature vector and the LDA eigenvector corresponding to the largest LDA eigenvalue as calculated from the training set. More details on the LDA FF features can be found in [1], where it is also shown that the FF+LDA measure is more discriminative than the MFCC+LDA measure.

3.2. Low-Frequency and High-Frequency Energy Dynamics Feature

In addition to the LDA measures, we experimented with two sub-band energy based features, low-frequency and high-frequency energy dynamics feature (*lfed* and *hfed*, respectively). *lfed* is calculated as follows:

$$E_l(t) = \log\left(\sum_k S(k,t)\right) \text{ where } 13 \le k \le 38 \quad (1)$$

$$dE_l(t) = \frac{1}{60}\sum_{i=-4}^{4} i \cdot E_l(t+i) \quad (2)$$

$$lfed(t) = \frac{1}{5}\sum_{i=-2}^{2} abs\left(dE_l(t+i)\right) \quad (3)$$

where $S(k,t)$ is the k-th bin of the FFT-512 power spectrum at the frame index t. *lfed* involves approximately a frequency range from 400 Hz to 1200 Hz comprising most of the interval of high energy concentration of the voiced speech sounds (sampling frequency of 16 kHz is assumed). *hfed* is

calculated in the same way but $144 \le k \le 208$, which correspond to the interval from 4500 Hz to 6500 Hz and this feature focuses on fricative sounds. The frequency intervals of both features are based on general knowledge and were not tuned to the application. A similar feature as *lfed* and *hfed* was proposed in [5] but that feature was calculated over the entire frequency range and it included spectral autocorrelation to emphasize the speech harmonic structure.

Notice that in the final signal representation, the contextual information is involved in several ways. First, before applying the LDA transform, the current delta and delta-delta feature involve an interval of 50 and 70 ms, respectively, in their calculation. Next, for the representation of the current frame, eight LDA measures are selected from a time window spanning the interval of 310 ms around the current frame. Finally, lfed and hfed involve a smoothed derivative calculation that in total uses an interval of 130 ms.

We use the SAD system on-line in our smart-room, so we avoid using techniques that would cause an algorithmic delay larger than a given acceptable value (set to 160 ms in our case). In addition, the designed SAD feature extraction saves computational resources since most of the calculation dedicated to the feature extraction is performed anyway for the ASR system due to the fact that SAD features are based on ASR features.

4. CLASSIFIER TRAINING

In this section, we explain the training procedures used for the three classifiers used in this work. We use the Decision Tree (DT) as our baseline classifier and we contrast its performance with another discriminative classifier based on Support Vector Machine (SVM) [7] approach and a generative classifier based on Gaussian Mixture Model (GMM) [9].

4.1. Features

For each frame at a time index *t*, one LDA measure *ldam*(*t*) is available. To include information from a time span larger than just 70ms into the representation, the eight most important LDA measures are selected from the interval $t-15 \le t \le t+15$. As a criterion for this selection, we used the entropy based information gain criterion used in the DT training algorithm (see [1] for more details on this selection process). LDA measures were concatenated to form the final representation vector.

ldam(t-15) ldam(t-10) ldam(t-6) ldam(t-3) ldam(t) ldam(t+3) ldam(t+6) ldam(t+10)

Using these features, we defined the following four different feature sets (in parenthesis is the feature vector size):

A: Eight ldam features selected using the C4.5 DT training (8 features)
B: A + lfed (9 features)
C: A + hfed (9 features)
D: B + hfed (10 features)

The A feature set is considered a baseline in our tests (a six-feature version of this set was already used in [1]). The feature sets B, C, and D allow us to observe the contribution of the low and high frequency dynamics features when added individually to the feature set A, as well as when both of them are added to A.

The 90 minute SPEECON training data, processed on frame-by-frame basis using a frame shift of 10 ms, results in over 500 thousand training examples with their corresponding Speech / Non-Speech labels. The Speech / Non-Speech labeling was performed by applying a forced Viterbi alignment on the training files using our speech recognition system.

4.2. Decision Tree (DT) Classifier

For DT training we used the C4.5 algorithm [6] which is an improvement of Quinlan's original ID3 DT training algorithm. During training, for each node of the decision tree, the best feature element from the feature vector is selected and the best threshold is set for this element. Using the SPEECON training example set and setting the pruning confidence level to 25% resulted in decision trees with the following number of nodes, depending on the feature set: A 1031, B 1475, C 1571, and D 2117.

4.3. Support Vector Machine (SVM) Classifier

A set of 500 thousand of examples is an enormous number of feature vectors to be used for the usual SVM training approach and hardly makes such training process feasible in practice. Alternative methods should be used; we tested the so-called cascade learning [8], however our implementation did not achieve a satisfactory performance. A better result was obtained by imposing a hard data reduction by randomly selecting 20 thousand examples where the two classes of interest are equally represented. The training data were firstly normalized anisotropicly to be in the range from –1 to 1, and the obtained normalizing template was then applied also to the testing data set. We used the Gaussian kernel with gamma parameter equal 5.0 and the C parameter (controlling the training error) equal 10.0. To train the system we use the publicly available SVMlight software package [10]. That preliminar SVM system was posteriorly modified as explained in the subsection 5.3.

4.4. Gaussian Mixture Model (GMM) Classifier

We used the well known Expectation-Maximization (EM) [11] algorithm for Gaussian mixture model training with the K-means algorithm for the model parameter initialization. The number of mixtures was set to 32 for both Speech and Non-Speech classes and diagonal covariance matrices were used. 20 iterations of the EM algorithm were performed.

5. EXPERIMENTS

5.1. Metrics

We present results using several metrics; as a primary metric we use the one defined for the SAD task in the NIST Rich Transcription evaluation. It is defined as the ratio of the duration of incorrect decisions to the duration of all speech segments in reference. We denote this metric as NIST in our results.

Notice that the NIST metric depends strongly on the prior distribution of Speech and Non-Speech in the test database. For example, a system that achieves a 5% error rate at Speech portions and a 5% error rate at Non-Speech portions, would result in very different NIST error rates for test databases with different proportion of Speech and Non-Speech segments; in the case of 90-to-10% ratio of Speech-to-Non-Speech the NIST error rate is 5.6%, while in the case of 50-to-50% ratio it is 10%. Due to this fact we report three metrics from CHIL: Mismatch Rate (MR), Speech Detection Error Rate (SDER), and Non-Speech Detection Error Rate (NDER) defined as:

- MR = Duration of Incorrect Decisions / Duration of All Utterances
- SDER = Duration of Incorrect Decisions at Speech Segments / Duration of Speech Segments
- NDER = Duration of Incorrect Decisions at Non-Speech Segments / Duration of Non-Speech Segments

5.2. Results on RT05 testing data

For the RT05 test database, a post-processing was applied to each SAD output consisting of marking the non-speech intervals shorter than 0.3 seconds as speech. This non-speech gap smoothing was used to mimic the same post-processing that was applied to the original human labels used as the reference.

Notice also that the test database contains much more Speech intervals than Non-Speech intervals (approx. 90% Speech). The NIST metric is inversely proportional to the amount of Speech in the reference labels. Thus, assuming the same amount of incorrect decisions in both a testing data with 50% Speech content and a testing data with 90% Speech content, a lower NIST error will be reported for the later testing data. This is the reason why some error rates reported for the unbalanced test database (RT05) may be lower that those reported for the more balanced train database (SPEECON).

Table 1 shows the results we obtained on the test database. Neither features nor classifiers were tuned to these test data. Most of the observations about the lfed and hfed features from the previous experiments with the training data hold also in this case, which is quite encouraging. An exception is the feature set D in the GMM classifier, where adding hfed does not improve the performance of the feature set B for the same classifier. In general, significant improvements can be seen when adding the lfed and hfed features to the LDA vector; for example in the case of SVM, the usage of both features reduces the original error by 52%. Among the classifiers, GMM achieves substantially lower error rates than the two discriminative classifiers (note that GMM was the worst performing on the training data). It seems in our case that the GMM classifier generalizes better the knowledge from the training data than the two other classifiers. The performance of DT and SVM is very similar, especially in the D feature set case. The best overall performance, NIST error of 8.47%, was obtained using the GMM classifier with the feature set B. It represents a 59% error rate reduction with respect to the performance of the baseline system consisting of the DT classifier and the feature set A (20.69%). Notice the NDER scores are high in comparison to the SDER scores in these tests. This is caused mostly by the combined effect of the Non-Speech gap post-processing and the low amount of the Non-Speech testing material which was mentioned above.

Table 1. Error rates obtained for the RT05 test data

	NIST MR / SDER / NDER			
Feat set	A	B	C	D
DT	20.69 18.77 / 18.21 / 24.32	12.37 11.20 / 9.24 / 30.51	14.76 13.37 / 11.65 / 30.27	11.54 10.43 / 8.10 / 33.42
SVM	23.88 21.71 / 21.46 / 24.22	14.70 13.36 / 11.87 / 28.19	15.69 14.26 / 12.51 / 31.76	11.45 10.41 / 7.99 / 34.56
GMM	12.25 11.13 / 8.86 / 33.81	8.47 7.69 / 4.61 / 38.42	10.02 9.11 / 5.24 / 47.70	8.66 7.88 / 3.75 / 49.00

5.3. Improvement of the SVM performance on the RT05 testing data

As it has been mentioned in subsection 4.3, a hard data reduction method has been applied that randomly selects 20 thousand examples from the SPEECON database where the two classes of interest are equally represented. The experiments in this subsection aim to improve the

performance of the SVM classifier doing two modifications in the training process.

The first one aims at an efficient sample selection. Instead of using a random selection of a reasonable number of samples, a two-step approach is chosen. Firstly, the whole training database is decomposed into chunks of 1000 samples. Then, on each chunk, a Proximal Support Vector Machine (PSVM) [12] has been trained. Unlike conventional SVM, PSVM solves a single square system of linear equations and thus it is very quick to train. In the nonlinear case (we use a Gaussian kernel) of PSVM the concept of support vector disappears as the separating hyperplane depends on all chunk data. In that way, all training data must be preserved for testing. While training, PSVMs 5-fold cross validation (CV) was applied to obtain optimal C and gamma parameters. After training, a threshold was applied on the CV accuracies of all chunks to select a given number of them that show the highest CV accuracy (we select the same number of data as in subsection 4.3, i.e. 20 chunks = 20 thousand samples). In the second step, a conventional SVM with the setting described in subsection 4.3 is trained on the chosen data.

The second modification makes use of the knowledge of the specific NIST metrics during the training phase. As it has been mentioned in subsection 5.1, NIST metrics depends on the prior distribution of Speech and Non-Speech in the test database. For this reason, if we want to improve the NIST scores for the RT05 evaluation, we should penalize the Speech class more than the Non-Speech class. That is possible for a discriminative classifier as SVM, by introducing different costs for the two classes (but not for a GMM-based classifier).

Table 2 shows results obtained on the RT05 evaluation with the modified SVM system and feature set D, along with the ones obtained with the best SVM and GMM systems from Table 1.

Table 2. Error rates obtained for the RT05 evaluation with the modified SVM system

	NIST MR / SDER / NDER
GMM	8.47 7.69 / 4.61 / 38.42
SVM	11.45 10.41 / 7.99 / 34.56
SVM modified	8.03 7.30 / 2.51 / 55.07

From Table 2 we observe that, as it can be expected after the second modification, the NDER score has increased but the SDER score, which has the major influence on the NIST measure, has strongly decreased. In consequence, after both modifications, the NIST error for the SVM-based system decreases from 11.45% to 8.03%, slightly outperforming the best GMM system.

5.4. Results on RT06 testing data

In this subsection we present the results achieved in the RT06 evaluation campaign. For that evaluation we chose a GMM classifier and, as features, we used the feature set D augmented by a cross-frequency energy dynamic feature, *xfed*, which is obtained as a combination of *lfed* and *hfed* and it is calculated as follows:

$$xfed(t) = \frac{\sqrt{hfed(t-9)\cdot lfed(t+9)}}{2} + \frac{\sqrt{hfed(t+9)\cdot lfed(t-9)}}{2} \quad (4)$$

This feature reaches high values when both the *hfed* before and *lfed* after (or *hfed* after and *lfed* before) the current frame have high values. It attempts to follow the energy flow between low and high frequencies typical for speech. The 9 frame distance was set empirically (it would correspond to 1/0.18 = 5.6 Hz energy flow rate) and it is limited by the maximum allowed algorithmic delay of the SAD system.

For the confmtg task, both SPEECON and RT05 databases were used to train the SAD system. For the lectmtg task, also a small amount of CHIL data was added into the training of the final system.

The RT06 evaluation campaign has several evaluation subtasks, depending on the used set of microphones. We participated in two subtasks, single distant microphone denoted as *sdm* and multiple distant microphone, *mdm*. The *sdm* subtask involved the centrally located omni-directional table microphone, or the best microphone selected after listening to the recordings. The *mdm* subtask involved at least 3 omni-directional table microphones, including the one selected for the *sdm* subtask.

In the tests with the RT06 data we applied a slightly different post-processing on top of the classifier output than we did in the case of the RT05 data tests. First, we performed a majority voting, where the central frame of an 11 frame interval was marked as Speech if 6 or more frames of this interval were classified as Speech; otherwise it was marked as Non-Speech. Second, to each Speech segment, 0.2 s of Speech was added at the beginning and the end of segment.

In the case of mdm, we applied a separate SAD to each individual channel, without post-processing. Outputs of all SAD systems were merged by a majority voting performed for each frame favoring the Speech label in the case of a tie. Then the same post-processing as in the sdm case was applied on the output of merging.

Table 3. Error rates obtained for the RT06 evaluation tasks

Feat set D + *xfed*	NIST MR / SDER / NDER	
	confmtg	lectmtg
sdm	5.45 5.1 / 3.1 / 41.4	7.10 6.2 / 0.4 / 48.1
mdm	5.63 5.3 / 3.5 / 38.7	5.30 4.6 / 0.7 / 33.3

Table 3 shows error rates obtained by our SAD system in the RT06 evaluation tasks. Very low NIST error rates were obtained and our SAD system ranked among the best systems; however, the Non-Speech detection error rates are high. To reduce this error rate will be the objective of our future work. We could benefit from the multiple microphones in the case of lectmtg task, however there is no significant change in the case of confmtg task.

6. CONCLUSION

The presented work is oriented towards a robust Speech Activity Detection (SAD) in smart-room environments. The baseline SAD system used features obtained by applying Linear Discriminative Analysis (LDA) on a parameter set modeling the shape of the signal spectrum; Decision Tree was used to perform the Speech/Non-Speech classification on a frame-by-frame basis. Both the LDA transform and the Decision Tree classifier were trained by a portion of the Spanish SPEECON database. The SAD system was evaluated on an interactive meeting database from the NIST RT05 evaluation campaign.

In this work we improved significantly the performance of the baseline speech detector. We tested additional features that measure the signal energy dynamics at low and high frequencies. Also, two other classifiers were evaluated for the SAD task: a discriminative Support Vector Machine classifier and a generative Gaussian Mixture Model (GMM) classifier. We observed that appending the high and low frequency energy dynamics features to the LDA features improved the performance for both training and testing data across all the classifiers with a higher benefit from the low-frequency feature.

In a first stage, the highest NIST error rate reduction was achieved by using the GMM classifier and the LDA features with the low-frequency energy dynamics; the error of the baseline SAD was reduced from 20.69% to 8.47% in this case. Then, two modifications of the usual training algorithm of the SVM-based classifier were developed in order to cope with two problems of that classifier in our application: the very large amount of training data and the particular characteristics of the NIST metric. With those two modifications, the SVM system further reduced the error to 8.03%. Finally, very competitive results were obtained in the RT06 SAD evaluation task, and they were also reported in this paper.

7. ACKNOWLEDGEMENTS

The authors wish to thank Jaume Padrell for encouraging discussions on the topic. This work has been partially sponsored by the EC-funded project CHIL (IST-2002-506909) and the Spanish Government-funded project ACESCA (TIN2005-08852).

8. REFERENCES

[1] Padrell J., Macho D., Nadeu C., "Robust Speech Activity Detection Using LDA Applied to FF Parameters", Proc. ICASSP'05, Philadelphia, PA, USA, March 2005.

[2] Iskra D. J. et al., "SPEECON - Speech Databases for Consumer Devices: Database Specification and Validation", Proc. LREC, 2002.

[3] Fiscus J.G., Radde N., Garofolo J.S., Le A., Ajot J., Laprun C., "The Rich Transcription 2005 Spring Meeting Recognition Evaluation", Lecture Notes in Computer Science (LNCS), vol. 3869, pp.369-389, Springer, February 2006.

[4] Nadeu, C., Macho, D., and Hernando, J., "Frequency and Time Filtering of Filter-Bank Energies for Robust HMM Speech Recognition", Speech Communication, Vol. 34, pp. 93-114, 2001.

[5] Ouzounov A., "Robust Feature for Speech Detection", Cybernetics and Information Technologies, vol.4, No.2, pp.3-14, 2004.

[6] Quinlan, J. R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1992.

[7] B. Schölkopf, A. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.

[8] Graf H.P., Cosatto E., Bottou L., Durdanovic I., Vapnik V., "Parallel Support Vector Machines: The Cascade SVM", Proc. Eighteenth Annual Conference on Neural Information Processing Systems, 2004.

[9] Duda R., Hart P., Stork D., *Pattern Classification*, 2nd Edition, Wiley-Interscience, 2000.

[10] SVMlight: http://svmlight.joachims.org/

[11] Rabiner L., Juang B.H., *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[12] Fung G., Mangasarian O., "Proximal Support Vector Machine Classifiers", Proc. Int. Conference on Knowledge Discovery and Data Mining, pp. 77-86, 2001.

DETECCIÓN DE ACTIVIDAD DE VOZ ROBUSTA AL RUIDO BASADA EN MODELOS ACÚSTICOS

Ángel de la Torre, Javier Ramírez, Carmen Benítez, José C. Segura, Luz García, Antonio J. Rubio

Departamento de Teoría de la Señal Telemática y Comunicaciones, Universidad de Granada
[atv,javierrp,carmen,segura,luzgm,rubio]@ugr.es

RESUMEN

En este trabajo proponemos un detector de actividad de voz (VAD) derivado de la aproximación VTS (Vector Taylor Series approach). Se hace uso de una mezcla de Gaussianas (entrenada con voz limpia) para proporcionar una decisión apropiada para la detección de voz/no-voz. La aproximación VTS adapta la mezcla de Gaussianas a las condiciones de ruido, lo que proporciona un rendimiento estable para un rango amplio de SNRs. Hemos evaluado el método propuesto en cuanto a la habilidad en la detección de voz/no-voz, y también en cuanto a la aplicación de VAD para reconocimiento robusto de voz. En comparación con otros métodos de detección de actividad de voz, el VAD propuesto presenta el mejor compromiso en detección de voz/no-voz. Cuando el VAD es aplicado para filtrado de Wiener y eliminación de frames de silencio, el método propuesto también muestra los mejores resultados de reconocimiento.

1. INTRODUCCIÓN

Los sistemas de reconocimiento automático de voz se ven fuertemente afectados por el ruido. Se han desarrollado numerosas técnicas para paliar el efecto del ruido sobre la tasa de reconocimiento. La mayor parte de ellos frecuentemente requieren estimar una estadística del ruido mediante un detector de actividad de voz (VAD) suficientemente preciso. La tarea de clasificación en voz/no-voz no es tan trivial como inicialmente podría parecer y la mayor parte de los algoritmos para VAD fallan cuando el nivel de ruido de fondo se incrementa. Durante la última década numerosos investigadores han desarrollado diferentes estrategias para detectar voz en señales ruidosas [1, 2, 3, 4], poniendo especial atención en la derivación y estudio de características y reglas de decisión robustas al ruido.

En este artículo proponemos un algoritmo de detección de actividad de voz basado en modelos acústicos. La decisión voz/no-voz está basada en la aproximación VTS (Vector Taylor Series approach) [5, 6, 7, 8]. La aproximación VTS, propuesta inicialmente como una técnica de compensación de ruido para reconocimiento robusto de voz, se ha adaptado para la detección de actividad de voz. La formulación VTS está basada en una mezcla de Gaussianas en el dominio de las energías del banco de filtros en escaladas logarítmicamente (log-FBE). La mezcla de Gaussianas se utiliza para calcular la probabilidad de cada Gaussiana dada la trama de entrada ruidosa. En la formulación VTS estas probabilidades son usadas para obtener una estimación limpia de la trama. En el algoritmo VAD basado en VTS que proponemos, estas probabilidades son usadas para calcular la probabilidad de que la trama sea voz. La decisión del VAD es realizada comparando la probabilidad de que la trama sea voz con un umbral. Con este enfoque se esperan dos ventajas. Por una parte, la decisión del VAD se apoya en un modelo de mezcla de Gaussianas entrenado con voz limpia, y por tanto, la decisión del VAD está basada en eventos de voz observados en la base de datos de entrenamiento. Por otra parte, la aproximación VTS proporciona un método para adaptar la mezcla de Gaussianas a las condiciones de ruido. De este modo, el método propuesto permite la adaptación del VAD a las condiciones de ruido y, por tanto, cabe esperar que el rendimiento del VAD se mantenga estable para un rango extenso de SNRs.

2. VAD BASADO EN VTS

2.1. Vector Taylor Series approach

La aproximación VTS [5, 6, 7, 8] es un método de compensación del ruido que proporciona una representación de la voz limpia eliminando el ruido aditivo. Esta compensación del ruido se realiza en el dominio log-FBE y está basada en una mezcla de Gaussianas. La formulación VTS asume que el efecto del ruido puede describirse como un término aditivo en el dominio log-FBE,

$$\mathbf{y}(\mathbf{x},\mathbf{n}) = \mathbf{x} + \mathbf{g}(\mathbf{x},\mathbf{n}) \quad (1)$$

donde $\mathbf{x}$ y $\mathbf{y}$ son vectores en este dominio que representan la voz limpia y ruidosa, respectivamente para un frame dado, y $\mathbf{n}$ representa el ruido aditivo que afecta a este frame. Para el canal i-ésimo, $\mathbf{g}$ queda descrita por la ecuación,

$$g(i) = \log\left(1 + \exp\left(n(i) - x(i)\right)\right) \quad (2)$$

Se pueden definir dos funciones auxiliares $f(i)$ y $h(i)$ del siguiente modo,

$$f(i) \equiv \frac{1}{1+\exp(x(i)-n(i))} = -\frac{\partial g(i)}{\partial x(i)} = \frac{\partial g(i)}{\partial n(i)} \quad (3)$$

$$h(i) \equiv (1-f(i))f(i) = \frac{\partial^2 g(i)}{\partial x^2(i)} = \frac{\partial^2 g(i)}{\partial n^2(i)} \quad (4)$$

y usando estas definiciones, podemos aproximar $y(i)$ usando un desarrollo en serie de Taylor alrededor de unos valores $x_0(i)$ y $n_0(i)$. Similarmente, usando la aproximación del desarrollo en serie de Taylor podemos describir cómo una función densidad de probabilidad Gaussiana en el dominio log-FBE se ve afectada por el ruido aditivo. Consideremos una función densidad de probabilidad Gaussiana que representa voz limpia, con media $\mu_x(i)$ y matriz de covarianza $\Sigma_x(i,j)$, y asumamos un proceso de ruido aditivo de tipo Gaussiano, con media $\mu_n(i)$ y matriz de covarianza $\Sigma_n(i,j)$. Podemos hacer un desarrollo en serie de Taylor alrededor de $x_0(i) = \mu_x(i)$ y $n_0(i) = \mu_n(i)$. La media y la matriz de covarianza de la función densidad de probabilidad que describe la voz ruidosa puede obtenerse como los valores esperados,

$$\mu_y(i) = E[y(i)] \quad (5)$$

$$\Sigma_y(i,j) = E[(y(i)-\mu_y(i))(y(j)-\mu_y(j))] \quad (6)$$

y pueden ser estimados como una función de $\mu_x(i)$, $\mu_n(i)$, $\Sigma_x(i,j)$ y $\Sigma_n(i,j)$ usando el desarrollo en serie de Taylor como,

$$\mu_y(i) \approx \mu_x(i) + g_0(i) + \frac{1}{2}h_0(i)[\Sigma_x(i,i) + \Sigma_n(i,i)] \quad (7)$$

$$\Sigma_y(i,j) \approx (1-f_0(i))(1-f_0(j))\Sigma_x(i,j) + f_0(i)f_0(j)\Sigma_n(i,j) + \frac{1}{2}h_0^2(i)(\Sigma_x(i,i)+\Sigma_n(i,i))^2\delta_{i,j} \quad (8)$$

donde $g_0(i)$, $f_0(i)$ y $h_0(i)$ están evaluadas en $x_0(i) = \mu_x(i)$ y $n_0(i) = \mu_n(i)$. De este modo, la aproximación VTS proporciona una función densidad de probabilidad Gaussiana que representa la voz ruidosa, calculada a partir de la Gaussiana que describe la voz limpia y la Gaussiana que describe el ruido.

Si la voz limpia es modelada como una mezcla de K Gaussianas, la aproximación VTS proporciona una estimación de la voz limpia $\hat{\mathbf{x}}$ dadas la voz ruidosa observada $\mathbf{y}$ y la estadística del ruido (μ_n y Σ_n) como,

$$\hat{\mathbf{x}} \approx \mathbf{y} - \sum_k P(k|\mathbf{y})\mathbf{g}(\mu_{x,k}, \mu_n) \quad (9)$$

donde $\mu_{x,k}$ es la media de la k-ésima Gaussiana limpia y $P(k|\mathbf{y})$ es la probabilidad de que la observación $\mathbf{y}$ haya sido generada por la Gaussiana k, dada por,

$$P(k|\mathbf{y}) = \frac{P(k)\mathcal{N}(\mathbf{y}, \mu_{y,k}, \Sigma_{y,k})}{\sum_{k'} P(k')\mathcal{N}(\mathbf{y}, \mu_{y,k'}, \Sigma_{y,k'})} \quad (10)$$

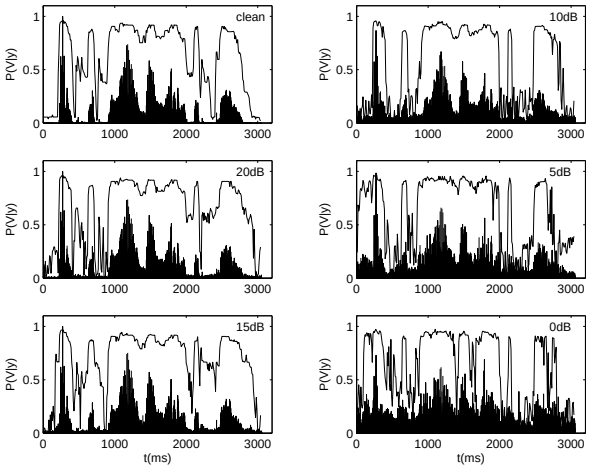


Figura 1. Probabilidad $P(V|\mathbf{y})$ de que el frame $\mathbf{y}$ sea voz para cada frame de una frase, evaluada a diferentes SNRs. La amplitud de la señal (para valores positivos) también se ha representado en cada gráfica.

donde $P(k)$ es la probabilidad a priori de la k-ésima Gaussiana y $\mathcal{N}(\mathbf{y}, \mu_{y,k}, \Sigma_{y,k})$ es la k-ésima Gaussiana ruidosa (con media $\mu_{y,k}$ y matriz de covarianza $\Sigma_{y,k}$) evaluada en $\mathbf{y}$. La media y la matriz de covarianza de la Gaussiana k-ésima ruidosa puede estimarse a partir de la estadística del ruido (μ_n y Σ_n) y de la k-ésima Gaussiana limpia ($\mu_{x,k}$ y $\Sigma_{x,k}$) usando las ecuaciones (7) y (8).

2.2. Aplicación de VTS a la detección de actividad de voz

Si a cada Gaussiana k se le asigna una probabilidad $P(V|k)$ (la probabilidad de que la Gaussiana k-ésima sea voz), la probabilidad de que un frame ruidoso de entrada $\mathbf{y}$ sea voz puede evaluarse como,

$$P(V|\mathbf{y}) = \sum_k P(V|k)P(k|\mathbf{y}) \quad (11)$$

donde $P(k|\mathbf{y})$ viene dada por la aproximación VTS (ecuación (10)).

La probabilidad $P(V|k)$ se puede estimar fácilmente para cada Gaussiana, dado que la mezcla de Gaussianas se ha construido a partir de una base de datos de entrenamiento limpia. En este trabajo hemos considerado la media para la componente de energía de la Gaussiana limpia E_k para la estimación de esta probabilidad, y se ha considerado una función lineal entre dos límites de energía (E_0 y E_1),

$$P(V|k) = \begin{cases} 0 & E_k < E_0 \\ (E_k - E_0)/(E_1 - E_0) & E_k \in [E_0, E_1] \\ 1 & E_k \geq E_0 \end{cases} \quad (12)$$

Los parámetros E_0 y E_1 se han ajustado empíricamente.

La figura 1 muestra la evolución en el tiempo de la probabilidad $P(V|\mathbf{y})$ evaluada para una frase extraída de la base de datos AURORA-2 [9], obtenida para diferentes SNRs. En este ejemplo se puede observar la efectividad del método a bajas SNRs. Se ontiene una probabilidad mayor que 0.8 para, al menos, algunos frames en cada sílaba incluso para SNRs bajas (esta frase corresponde con la cadena de dígitos ingleses 86Z1162). La decisión voz/no-voz puede realizarse usando un umbral T. El frame $\mathbf{y}$ es etiquetado como voz si $P(V|\mathbf{y}) > T$, y como no-voz en caso contrario. Adicionalmente, se puede incluir un time-in y un time-out de algunos frames con objeto de evitar que el VAD descarte aquellos frames de voz con poca energía al principio y al final de algunas sílabas.

3. MARCO EXPERIMENTAL

Se suelen llevar a cabo varios experimentos con objeto de evaluar el rendimiento de los algoritmos para VAD. El análisis está normalmente enfocado en la determinación de las probabilidades de error para diferentes escenarios de ruido y valores de la SNR [10, 4], así como en la influencia de la decisión del VAD sobre el rendimiento de los sistemas de procesamiento de voz [11, 12]. El marco experimental y los test objetivos llevados a cabo para evaluar el rendimiento del algoritmo propuesto se describen en esta sección.

3.1. Evaluación en diferentes entornos de ruido

En primer lugar, el VAD propuesto se evaluó en términos de la habilidad para discriminar entre voz y no-voz en diferentes escenarios y con diferentes niveles de ruido usando la base de datos AURORA-2 [9]. Esta base de datos se ha construido a partir de la base de datos limpia TIdigits (que consiste en secuencias de hasta siete dígitos conectados pronunciados por locutores de inglés americano), usada como fuente de voz, y una selección de 8 ruidos reales diferentes, que han sido añadidos artificialmente a la voz a SNRs de 20dB, 15dB, 10dB, 5dB, 0dB y -5dB. En el análisis de discriminación, la base de datos limpia TIdigits se usó para etiquetar manualmente cada frame de cada frase como voz o no-voz para referencia. El rendimiento de detección de actividad de voz es evaluado en términos de tasa de acierto de no-voz (HR0) y tasa de acierto de voz (HR1), definidos como la fracción de todos los frames de no-voz o voz que han sido correctamente detectados como no-voz o voz, respectivamente,

$$\text{HR1} = \frac{N_{1,1}}{N_1^{\text{ref.}}} \qquad \text{HR0} = \frac{N_{0,0}}{N_0^{\text{ref.}}} \tag{13}$$

donde $N_1^{\text{ref.}}$ y $N_0^{\text{ref.}}$ son el número de frames de no-voz o voz que realmente hay en la base de datos, y $N_{1,1}$ y $N_{0,0}$ son el número de frames de voz y no-voz que han sido correctamente clasificados, respectivamente.

La figura 2 compara el VAD basado en VTS propuesto (usando un umbral T=0.5) frente a algoritmos estandarizados incluyendo ITU-T G.729 [13], ETSI AMR [14] y ETSI AFE [15], y otros algoritmos recientemente reportados [1, 2, 3, 4]. Se muestran la tasa de acierto de no-voz (HR0) y la tasa de acierto de voz (HR1) para condiciones limpias y para niveles de SNR de entre 20 y -5 dB. Nótese que se proporcionan los resultados para los dos VADs definidos en el estándar [15] (uno para la estimación del espectro del ruido en la fase de filtrado de Wiener, WF, y el otro usado en la fase de eliminación de frames de no-voz, FD). Los resultados mostrados en estas figuras son valores promedio para el conjunto completo de ruidos. Puede concluirse, de la figura 2 que:

- El VAD ITU-T G.729 muestra una precisión pobre en detección de voz cuando se incrementa el nivel de ruido, mientras que la detección de no-voz es buena en condiciones limpias (85 %) y pobre (20 %) en condiciones ruidosas.
- El VAD ETSI AMR1 presenta un comportamiento extremadamente conservativo con una precisión alta en la detección de voz para todo el rango de niveles de SNR, pero a costa de unos resultados en detección de no-voz muy pobres a medida que se incrementa el nivel de ruido. Aunque el AMR1 parece un procedimiento apropiado para detección de voz en condiciones de ruido desfavorables, su comportamiento extremadamente conservativo degrada la precisión en detección de no-voz, llegando el HR0 a ser menor del 10 % por debajo de 10 dB, lo que lo hace poco útil para sistemas de procesamiento de voz.
- El VAD ETSI AMR2 alcanza una mejora considerable sobre los detectores G.729 y AMR1, alcanzando mejores precisiones en detección de no-voz, aunque aun sufre una degradación rápida de la detección de voz en condiciones de ruido desfavorables.
- El VAD usado en el estándar AFE para la estimación del espectro del ruido en la etapa de filtrado de Wiener está basado en la energía de la banda completa y proporciona un rendimiento pobre en detección de voz con una caída rápida del HR1 para valores bajos de la SNR. Por otra parte, el VAD usado en el AFE para eliminación de frames de no-voz (frame-dropping, FD) alcanza una alta precisión en detección de voz pero resultados moderados en detección de no voz.
- Finalmente, el VAD basado en VTS que proponemos alcanza el mejor compromiso de entre los dife-

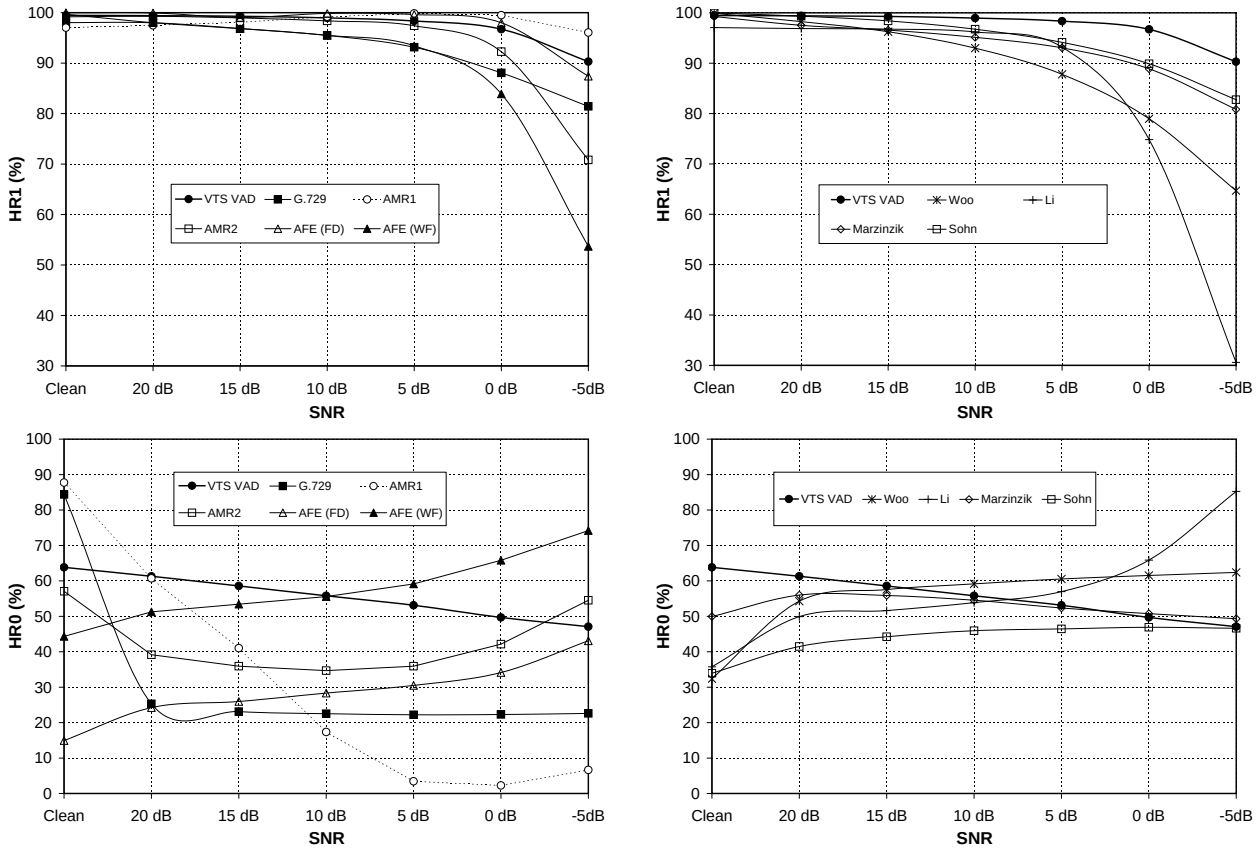


Figura 2. Análisis de la discriminación voz/no-voz en función de la SNR. Resultados promediados para todos los ruidos considerados en la base de datos AURORA-2. Se representan la tasa de acierto de voz y de no-voz (comparados con los obtenidos estándares y otros VADs recientemente reportados.)

rentes procedimientos evaluados. Este VAD obtiene un buen comportamiento en detección de periodos de no-voz y además muestra una caída lenta en el rendimiento para condiciones de ruido desfavorables en la detección de frames de voz (90 % a -5 dB).

La tabla 1 muestra un resumen de las ventajas proporcionadas por el VAD basado en VTS sobre otros métodos de detección de voz en términos de las tasas de detección de voz y no-voz promediadas sobre el rango completo de SNRs. De la tabla puede observarse que el método propuesto, con un HR1 promedio del 97.50 % y un HR0 promedio del 55.62 %, alcanza el mejor compromiso en detección de voz/no-voz cuando se compara con el resto de detectores de voz evaluados.

3.2. Evaluación del VAD en un sistema de reconocimiento robusto

Si bien el análisis de discriminación presentado en la sección anterior es efectivo para la evaluación de un algoritmo de detección de voz/no-voz, también hemos estudiado la influencia del VAD sobre un sistema de reconocimiento de voz. El marco de referencia es front-end para reconocimiento distribuido de voz (DSR) [16] propuesto por el grupo de trabajo STQ de la ETSI para la evaluación de algoritmos de extracción de características robustos al ruido para DSR. Se ha estudiado la influencia de la decisión del VAD sobre el rendimiento de diferentes aspectos de la extracción de características. La primera aproximación incorpora filtrado de Wiener (WF) al sistema base como método de supresión de ruido. El segundo algoritmo de extracción de características combina filtrado de Wiener y eliminación de frames de no-voz (frame dropping, FD). La tabla 2 muestra los resultados de reconocimiento para AURORA-2 en función de la SNR para experimentos de reconocimiento de voz basados en los

Tabla 2. Tasa de reconocimiento promedio (Word Accuracy) para experimentos "clean training" con la base de datos AURORA-2.

	Base	**Base + WF**					**Base + WF + FD**				
		G.729	AMR1	AMR2	AFE	**VTS-VAD**	G.729	AMR1	AMR2	AFE	**VTS-VAD**
Clean	99.03	98.81	98.80	98.81	98.77	98.85	98.41	97.87	98.63	98.78	98.58
20 dB	94.19	87.70	97.09	97.23	97.68	97.41	83.46	96.83	96.72	97.82	97.51
15 dB	85.41	75.23	92.05	94.61	95.19	95.07	71.76	92.03	93.76	95.28	95.37
10 dB	66.19	59.01	74.24	87.50	87.29	88.11	59.05	71.65	86.36	88.67	89.34
5 dB	39.28	40.30	44.29	71.01	66.05	70.93	43.52	40.66	70.97	71.55	73.83
0 dB	17.38	23.43	23.82	41.28	30.31	40.65	27.63	23.88	44.58	41.78	44.81
-5 dB	8.65	13.05	12.09	13.65	4.97	13.16	14.94	14.05	18.87	16.23	17.61
Average	60.49	57.13	66.30	78.33	75.30	**78.43**	57.08	65.01	78.48	79.02	**80.17**

Tabla 1. Tasas de acierto de voz y no-voz promediadas para SNRs entre condiciones limpias y -5dB.

Comparación con VADs estándar			
	G.729	AMR1	AMR2
HR0 (%)	31.77	31.31	42.77
HR1 (%)	93.00	98.18	93.76
	AFE (WF)	AFE (FD)	**VTS-VAD**
HR0 (%)	57.68	28.74	**55.62**
HR1 (%)	88.72	97.70	**97.50**

Comparación con otros VADs					
	Sohn	Woo	Li	Marz.	**VTS-VAD**
HR0 (%)	43.66	55.40	57.03	52.69	**55.62**
HR1 (%)	94.46	88.41	83.65	93.04	**97.50**

algoritmos de VAD G.729, AMR, AFE, y VTS VAD. Estos resultados se han promediado sobre los tres conjuntos de test definidos para los experimentos de reconocimiento de AURORA-2. Como conclusión puede apreciarse que el VAD propuesto supera a los estándares de detección de voz G.729, AMR1, AMR2 y AFE, cuando es aplicado para el filtrado de Wiener y también cuando es aplicado conjuntamente para filtrado de Wiener y para eliminación de frames de no-voz.

4. CONCLUSIONES

En este trabajo hemos propuesto un algoritmo VAD basado en modelos acústicos derivado de la aproximación VTS (Vector Taylor Series approach). El uso de una mezcla de Gaussianas (en el dominio log-FBE) entrenada con una base de datos de voz limpia proporciona una regla de decisión de voz apropiada para la detección de voz/no-voz. Por otra parte, la formulación VTS permite la adaptación de la mezcla de Gaussianas a las condiciones de ruido, alcanzándose un rendimiento estable del VAD propuesto para un rango extenso de SNRs y de tipos de ruido. El VAD basado en VTS propuesto se ha evaluado en términos de la capacidad para discriminación entre voz y no-voz en diferentes escenarios de ruido. Cuando se ha comparado con otros VADs estándar, hemos encontrado que el VAD propuesto proporciona el mejor compromiso en detección de voz/no-voz, con un HR1 promedio del 97.50 % y un HR0 promedio del 55.62 % (promediados entre condiciones limpias y -5dB). Con respecto al rendimiento en reconocimiento de voz, el VAD propuesto también proporciona los mejores resultados de reconocimiento cuando es usado para la estimación del espectro del ruido en el filtrado de Wiener y cuando es usado para eliminación de tramas de no-voz.

5. AGRADECIMIENTOS

Este trabajo ha recibido soporte del 6^o Programa Marco de la Unión Europea, a través del contrato número IST-2002-507943 (HIWIRE, Human Input that Works in Real Environments), y del Gobierno Español a través del proyecto SR3-VoIP (TEC2004-03829/TCM). Los puntos de vista expresados en este trabajo corresponden únicamente a los autores. La Unión Europea no es responsable de ningún uso que se pueda hacer de la información contenida en este trabajo.

6. BIBLIOGRAFÍA

[1] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 16, no. 1, pp. 1–3, 1999.

[2] K. Woo, T. Yang, K. Park, and C. Lee, "Robust voice activity detection algorithm for estimating noise

spectrum," *Electronics Letters*, vol. 36, no. 2, pp. 180–181, 2000.

[3] Q. Li, J. Zheng, A. Tsai, and Q. Zhou, "Robust endpoint detection and energy normalization for real-time speech and speaker recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 3, pp. 146–157, 2002.

[4] M. Marzinzik and B. Kollmeier, "Speech pause detection for noise spectrum estimation by tracking power envelope dynamics," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 6, pp. 341–351, 2002.

[5] P.J. Moreno, *Speech Recognition in Noisy Environments*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, Pensilvania, 1996.

[6] P.J. Moreno, B. Raj, and R.M. Stern, "A vector Taylor series approach for environment-independent speech recognition," in *Proc. of ICASSP-96*, 1996, pp. 733–736.

[7] R.M. Stern, B. Raj, and P.J. Moreno, "Compensation for environmental degradation in automatic speech recognition," in *ESCA-NATO Tutorial and Research Workshop on Robust Speech Recognition for Unknown Communication Channels*, 1997, pp. 33–42.

[8] J.C. Segura, A. de la Torre, M.C. Benítez, and A.M. Peinado, "Model-based compensation of the additive noise for continuous speech recognition. Experiments using the Aurora-II database and tasks," in *Proc. of EuroSpeech-2001*, 2001, pp. 221–224.

[9] H.G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noise conditions," in *ISCA ITRW ASR2000 Automatic Speech Recognition: Challenges for the Next Millennium*, Paris, France, September 2000.

[10] F. Beritelli, S. Casale, G. Rugeri, and S. Serrano, "Performance evaluation and comparison of G.729/AMR/Fuzzy voice activity detectors," *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 85–88, 2002.

[11] J. Ramírez, J. C. Segura, M. C. Benítez, A. de la Torre, and A. Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech Communication*, vol. 42, no. 3-4, pp. 271–287, 2004.

[12] J. Ramírez, José C. Segura, C. Benítez, A. de la Torre, and A. Rubio, "An effective subband OSF-based VAD with noise reduction for robust speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 13, no. 6, pp. 1119–1129, 2005.

[13] ITU, "A silence compression scheme for G.729 optimized for terminals conforming to recommendation V.70," *ITU-T Recommendation G.729-Annex B*, 1996.

[14] ETSI, "Voice activity detector (VAD) for Adaptive Multi-Rate (AMR) speech traffic channels," *ETSI EN 301 708 Recommendation*, 1999.

[15] ETSI, "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms," *ETSI ES 202 050 Recommendation*, 2002.

[16] ETSI, "Speech processing, transmission and quality aspects (stq); distributed speech recognition; front-end feature extraction algorithm; compression algorithms," *ETSI ES 201 108 Recommendation*, 2000.

VOICE ACTIVITY DETECTION USING A CONTEXTUAL INFORMATION AND MULTIPLE HYPOTHESIS TESTING

J. Ramírez, J. C. Segura, J. M. Górriz, A. de la Torre, L. García and C. Benítez

Dept. of Signal Theory, Networking and Communications
University of Granada (SPAIN)

ABSTRACT

This paper shows a revised statistical test for voice activity detection in noise adverse environments. The method is based on a revised contextual likelihood ratio test (LRT) defined over a multiple observation window. The new approach not only evaluates the two hypothesis consisting on all the observations to be speech or non-speech but all the possible hypothesis defined over the individual observations. The implicit hangover mechanism artificially added by the original method was not found in the revised method so its design can be further improved. With these and other innovations the proposed method showed a high speech/non-speech discrimination over a wide range of SNR conditions. The experimental framework showed that the revised method yields significant improvements over standardized VADs for discontinous voice transmission and distributed speech recognition, as well as over recently reported methods.

1. INTRODUCTION

Emerging applications in the field of speech processing are demanding increasing levels of performance in noise adverse environments. Examples of such systems are the new voice services including discontinuous speech transmission [1, 2, 3] or distributed speech recognition (DSR) over wireless and IP networks [4]. These systems often require a noise reduction scheme working in combination with a precise voice activity detector (VAD) in order to compensate its harmful effect on the speech signal.

During the last decade numerous researchers have studied different strategies for detecting speech in noise and the influence of the VAD on the performance of speech processing systems. Sohn *et al*. [5] proposed a robust VAD algorithm based on a statistical likelihood ratio test (LRT) involving a single observation vector. Later, Cho *et al* [6] suggested an improvement based on a smoothed LRT. Most VADs in use today normally consider hangover algorithms based on empirical models to smooth the VAD decision. It has been shown recently that incorporating contextual information in a multiple observation LRT (MO-LRT) [7] reports benefits for speech/pause discrimination in high noise environments. This paper analyzes this method and shows a new LRT VAD that extends the number of hypothesis on the individual multiple observation that are tested.

This work has been funded by the European Commission (HIWIRE, IST No. 507943) and the Spanish MEC project TEC2004-03829/FEDER.

2. MULTIPLE OBSERVATION LIKELIHOOD RATIO TEST

In a two-hypothesis test, the optimal decision rule minimizing the error probability is the Bayes classifier. Given an observation vector $\tilde{\mathbf{y}}$ to be classified, the problem is reduced to selecting the class (G_0 or G_1) with the largest posterior probability $P(G_i|\tilde{\mathbf{y}})$. From the Bayes rule, a likelihood ratio test (LRT) can be defined as:

$$L(\tilde{\mathbf{y}}) = \frac{p(\tilde{\mathbf{y}}|G_1)}{p(\tilde{\mathbf{y}}|G_0)} \underset{G_0}{\overset{G_1}{\gtrless}} \frac{P(G_0)}{P(G_1)} \quad (1)$$

where the observation vector is classified as G_1 if the likelihood ratio $L(\tilde{\mathbf{y}})$ is greater than the ratio $P(G_0)/P(G_1)$ between the *a priori* class probabilities, otherwise it is classified as G_0. Frequently, there is a need to shift the operating point of the classifier in favor of one of the two classes so that $L(\tilde{\mathbf{y}})$ is compared to a threshold η representing the separation between the classes.

A LRT for detecting the presence of speech in a noisy signal based on a Gaussian model was proposed by Sohn *et al.* [5] and several improvements [6, 8] have been considered to improve its performance. Among them, the multiple observation LRT (MO-LRT) [7] considers not just a single observation vector $\tilde{\mathbf{y}}_t$ measured at a frame t, but also an N-frame neighborhood $\{\tilde{\mathbf{y}}_{t-N}, ..., \tilde{\mathbf{y}}_t, ..., \tilde{\mathbf{y}}_{t+N}\}$:

$$\ell(\tilde{\mathbf{y}}_{t-N}, ...\tilde{\mathbf{y}}_{t+N}) = \frac{p_{\mathbf{y}_{t-N},...,\mathbf{y}_{t+N}|G_1}(\tilde{\mathbf{y}}_{t-N}, ..., \tilde{\mathbf{y}}_{t+N}|G_1)}{p_{\mathbf{y}_{t-N},...,\mathbf{y}_{t+N}|G_0}(\tilde{\mathbf{y}}_{t-N}, ..., \tilde{\mathbf{y}}_{t+N}|G_0)} \quad (2)$$

This test involves the evaluation of an N-th order LRT incorporating contextual information to the decision rule and exhibits significant improvements in speech/pause discrimination over the original LRT proposed by Sohn [5].

This smoothed test introduces a non-controllable hangover mechanism that needs to be studied and discussed. This paper reformulates the MO-LRT previously proposed in [7] and shows a new and effective LRT yielding high speech/pause discrimination accuracy. The hangover is then eliminated and not affected by the selection of the number of frames involved in the LRT.

3. REVISED MO-LRT

It is interesting to analyze the hypothesis that are being tested in the evaluation of the previous MO-LRT VAD. Note that the decision is made in favor of one of the two hypothesis:

$$\begin{array}{lll} G_1 & : & \hat{\mathbf{y}}_l = \hat{\mathbf{s}}_l + \hat{\mathbf{n}}_l \\ G_0 & : & \hat{\mathbf{y}}_l = \hat{\mathbf{n}}_l \end{array} \tag{3}$$

for $l = t - N, ..., t, ..., t + N$. The VAD operates on a frame by frame basis and assigns a class to the central frame at time t. In this way, the test evaluates the probability that "all" the observations in the N-frame neighborhood of the central frame to be non-speech or speech. This is the reason to revise the method in order to evaluate not just the two previous hypothesis G_0 and G_1 but also other hypothesis that could be equally possible.

Let $\aleph = \{H_m, m = 1, 2, ..., 2^{2N+1}\}$ be the set of all the possible hypothesis considering all the individual observations to be speech or non-speech in the multiple observation vector $\{\tilde{\mathbf{y}}_{t-N}, ..., \tilde{\mathbf{y}}_t, ..., \tilde{\mathbf{y}}_{t+N}\}$ that is reindexed as $\hat{\mathbf{Y}}$= $\{\hat{\mathbf{y}}_1, ..., \hat{\mathbf{y}}_{N+1}, ..., \hat{\mathbf{y}}_{2N+1}\}$ for convenience of the presentation. Each hypothesis H_m can be defined in terms of a binary integer representation:

$$m = \sum_{k=1}^{2N+1} 2^{b_k} \tag{4}$$

where b_k define if the observation k is non-speech ($b_k = 0$) or speech ($b_k = 1$):

$$\begin{array}{lll} b_k = 1 & : & \hat{\mathbf{y}}_k = \hat{\mathbf{s}}_k + \hat{\mathbf{n}}_k \\ b_k = 0 & : & \hat{\mathbf{y}}_k = \hat{\mathbf{n}}_k \qquad\qquad k = 1, 2, ..., N+1 \end{array} \tag{5}$$

Thus, each hypothesis H_m consists of $2N$+1 individual hypothesis involving the $2N$+1 observations. The classification problem is then reformulated as selecting the class i with the higher posterior probability $P(H_i|\hat{\mathbf{Y}})$ and assigning speech (G_1) or non-speech (G_0) to the current frame depending on the bit b_{N+1} associated to H_i.

If the set $\aleph$ of all the possible hypothesis is splitted depending on the value of the central frame bit b_{N+1} as:

$$\begin{array}{l} M_1 = \{H_m \in \aleph : b_{N+1} = 1\} \\ M_0 = \{H_m \in \aleph : b_{N+1} = 0\} \end{array} \tag{6}$$

the posterior probabilities are defined to be:

$$\begin{array}{l} p(G_1|\hat{\mathbf{Y}}) = \sum_{m \in M_1} p(H_m|\hat{\mathbf{Y}}) \\ p(G_0|\hat{\mathbf{Y}}) = \sum_{m \in M_0} p(H_m|\hat{\mathbf{Y}}) \end{array} \tag{7}$$

and

$$\begin{array}{l} P(G_1) = \sum_{m \in M_1} P(H_m) \\ P(G_0) = \sum_{m \in M_0} P(H_m) \end{array} \tag{8}$$

Using the Bayes rule:

$$\begin{array}{l} p(G_1|\hat{\mathbf{Y}}) = \frac{1}{P(\hat{\mathbf{Y}})} \sum_{m \in M_1} P(H_m) p(\hat{\mathbf{Y}}|H_m) \\ p(G_0|\hat{\mathbf{Y}}) = \frac{1}{P(\hat{\mathbf{Y}})} \sum_{m \in M_0} P(H_m) p(\hat{\mathbf{Y}}|H_m) \end{array} \tag{9}$$

and a revised LRT can be defined as:

$$\Lambda = \frac{p(G_1|\hat{\mathbf{Y}})}{p(G_0|\hat{\mathbf{Y}})} = \frac{\sum_{m \in M_1} P(H_m) p(\hat{\mathbf{Y}}|H_m)}{\sum_{m \in M_0} P(H_m) p(\hat{\mathbf{Y}}|H_m)} \tag{10}$$

An effective approximation to the statistical test described above is to replace the summation by the maximum value of the probability of the hypothesis in M_1 and M_0:

$$\Lambda^* = \frac{\text{máx}_{m \in M_1}\, p(\hat{\mathbf{Y}}|H_m)}{\text{máx}_{m \in M_0}\, p(\hat{\mathbf{Y}}|H_m)} \tag{11}$$

By taking logarithms this test is expressed in a more compact form:

$$\log \Lambda^* = \max_{m \in M_1} l_m - \max_{m \in M_0} l_m \tag{12}$$

where:

$$l_m = \sum_{k=1}^{2N+1} \log p(\hat{\mathbf{y}}_k|b_k) \tag{13}$$

If we restrict the number of possible hypothesis by removing those corresponding to more than one speech/non-speech or non-speech/speech transition in the N-frame neighborhood, the test can be rewritten in matrix form:

$$\mathbf{L} = \mathbf{K}\mathbf{B}_1 + (\mathbf{I} - \mathbf{K})\mathbf{B}_0 \tag{14}$$

where:

$$\begin{array}{l} \mathbf{L} = [l_1, l_2, ..., l_{2N+1}]^T \\ \mathbf{B}_0 = [\log p(\hat{\mathbf{y}}_1|0), ..., \log p(\hat{\mathbf{y}}_{2N+1}|0)]^T \\ \mathbf{B}_1 = [\log p(\hat{\mathbf{y}}_1|1), ..., \log p(\hat{\mathbf{y}}_{2N+1}|1)]^T \end{array} \tag{15}$$

and $\mathbf{K}$ is the Hankel matrix:

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & ... & 0 & ... & 0 & 0 \\ 0 & 0 & ... & 0 & ... & 0 & 1 \\ 0 & 0 & ... & 0 & ... & 1 & 1 \\ ... & ... & ... & ... & ... & ... & ... \\ 0 & 1 & ... & 1 & ... & 1 & 1 \\ 1 & 1 & ... & 1 & ... & 1 & 1 \\ 1 & 1 & ... & 1 & ... & 1 & 0 \\ & & & & & & \\ 1 & 1 & ... & 0 & ... & 0 & 0 \\ 1 & 0 & ... & 0 & ... & 0 & 0 \end{bmatrix} \tag{16}$$

Moreover, if the matrix $\mathbf{K}$ is splitted into two submatrices $\mathbf{K}_0$ and $\mathbf{K}_1$ by extracting from $\mathbf{K}$ the rows with a central 0 or 1, respectively, the test is easily reduced to:

$$\log \Lambda^* = \text{máx}\, \mathbf{L}_1 - \text{máx}\, \mathbf{L}_0 \tag{17}$$

where:

$$\begin{array}{l} \mathbf{L}_1 = \mathbf{K}_1\mathbf{B}_1 + (\mathbf{I} - \mathbf{K}_1)\mathbf{B}_0 \\ \mathbf{L}_0 = \mathbf{K}_0\mathbf{B}_1 + (\mathbf{I} - \mathbf{K}_0)\mathbf{B}_0 \end{array} \tag{18}$$

Note that $\mathbf{K}_0 = \mathbf{I} - \mathbf{K}_1$ and equation 18 is reduced to:

$$\begin{array}{l} \mathbf{L}_1 = \mathbf{K}_1\mathbf{B}_1 + (\mathbf{I} - \mathbf{K}_1)\mathbf{B}_0 \\ \mathbf{L}_0 = (\mathbf{I} - \mathbf{K}_1)\mathbf{B}_1 + \mathbf{K}_1\mathbf{B}_0 \end{array} \tag{19}$$

As an example, for $N = 1$, the matrices $\mathbf{K}$, $\mathbf{K}_0$ and $\mathbf{K}_1$ are defined to be:

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \mathbf{K}_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad \mathbf{K}_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{20}$$

and $\mathbf{L}_1$ and $\mathbf{L}_0$ are computed by:

$$\begin{array}{l} \mathbf{L}_1 = \mathbf{K}_1\mathbf{B}_1 + (\mathbf{I} - \mathbf{K}_1)\mathbf{B}_0 = \\ = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \log p(\hat{\mathbf{y}}_1|1) \\ \log p(\hat{\mathbf{y}}_2|1) \\ \log p(\hat{\mathbf{y}}_3|1) \end{bmatrix} + \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \log p(\hat{\mathbf{y}}_1|0) \\ \log p(\hat{\mathbf{y}}_2|0) \\ \log p(\hat{\mathbf{y}}_3|0) \end{bmatrix} \\ \mathbf{L}_0 = (\mathbf{I} - \mathbf{K}_1)\mathbf{B}_1 + \mathbf{K}_1\mathbf{B}_0 = \\ = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \log p(\hat{\mathbf{y}}_1|1) \\ \log p(\hat{\mathbf{y}}_2|1) \\ \log p(\hat{\mathbf{y}}_3|1) \end{bmatrix} + \\ \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \log p(\hat{\mathbf{y}}_1|0) \\ \log p(\hat{\mathbf{y}}_2|0) \\ \log p(\hat{\mathbf{y}}_3|0) \end{bmatrix} \end{array} \tag{21}$$

The algorithm for voice activity detection is based on a comparison of a likelihood ratio to a given threshold η:

$$\log \Lambda^* \underset{G_0}{\overset{G_1}{\gtrless}} \eta \tag{22}$$

For the computation of the logarithmic probability vectors $\mathbf{B}_0$ and $\mathbf{B}_1$, an adequate statistical model needs to be selected. In this work, the discrete Fourier transform (DFT) coefficients of the clean speech (S_j) and the noise (N_j) are assumed to be asymptotically independent Gaussian random variables:

$$\begin{array}{l} p(\hat{\mathbf{y}}|G_0) = \prod_{j=0}^{J-1} \frac{1}{\pi\lambda_N(j)} \exp\{-\frac{|Y_j|^2}{\lambda_N(j)}\} \\ p(\hat{\mathbf{y}}|G_1) = \prod_{j=0}^{J-1} \frac{1}{\pi[\lambda_N(j)+\lambda_S(j)]} \exp\{-\frac{|Y_j|^2}{\lambda_N(j)+\lambda_S(j)}\} \end{array} \tag{23}$$

where Y_j represents the noisy speech DFT coefficients and $\lambda_N(j)$ and $\lambda_S(j)$ denote the variances of N_j and S_j, respectively. Thus, the logarithmic probabilities found in $\mathbf{B}_0$ and $\mathbf{B}_1$ can be computed as in [7] through the "a priori" and "a posteriori" SNRs defined to be:

$$\gamma_j = \frac{|Y_j|^2}{\lambda_N(j)} \qquad \xi_j = \frac{\lambda_S(j)}{\lambda_N(j)} \tag{24}$$

that are estimated using the Ephraim and Malah minimum mean-square error (MMSE) estimator [9].

The algorithm is adaptive and suitable for non-stationary noise environments since the statistical properties are updated when the frame is classified as a non-speech frame. In this way, the variance of the noise λ_N is updated as:

$$\lambda_N(j) = \alpha\lambda_N(j) + (1 - \alpha)|Y_j|^2 \tag{25}$$

Figure 1 shows the operation of the original MO-LRT VAD and the revised one over an utterance of the Spanish SpeechDatCar database [10] in clean conditions (25 dB SNR). Note that the new algorithm removes the saving period at the word beginnings and endings being more accurate in such a low noise conditions. It is interesting to point out that the hangover of the original MO-LRT was a result of extending the decision over a neighborhood of the current frame. However, the new statistical test exhibits the same smoothing process and reduced variance of the decision variable with the benefit of being suitable for a more effective hangover mechanism development. Under the noisiest conditions (5 dB SNR), the new algorithm has a similar behavior to the previous VAD as shown in figure 2.

4. EXPERIMENTAL RESULTS

The ROC (receiving operating characteristic) curves have shown to be very effective for the evaluation of voice activity detectors [11, 12]. These plots, which show the trade-off between the error probabilities of speech and non-speech detection as the threshold η varies, completely describe the VAD error rate. In this analysis, the Spanish SpeechDat-Car (SDC)[10] database was used. This database consists of recordings from distant and close-talking microphones in car environments at different driving conditions. For the computation of the speech and non-speech distributions, a semiautomatic "speech/non-speech" labeling process was conducted on the close talking microphone.

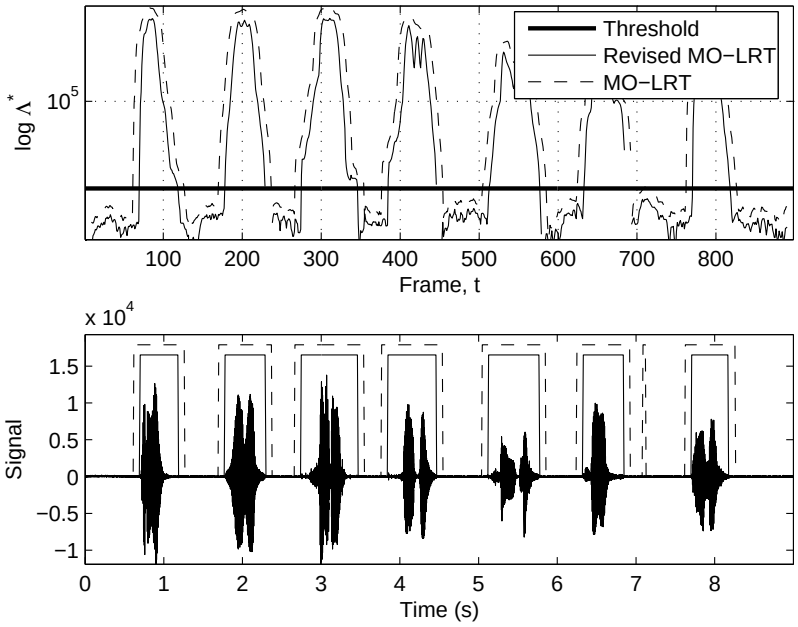


Figura 1. *Comparison between the original MO-LRT and the revised MO-LRT for VAD in clean conditions.*

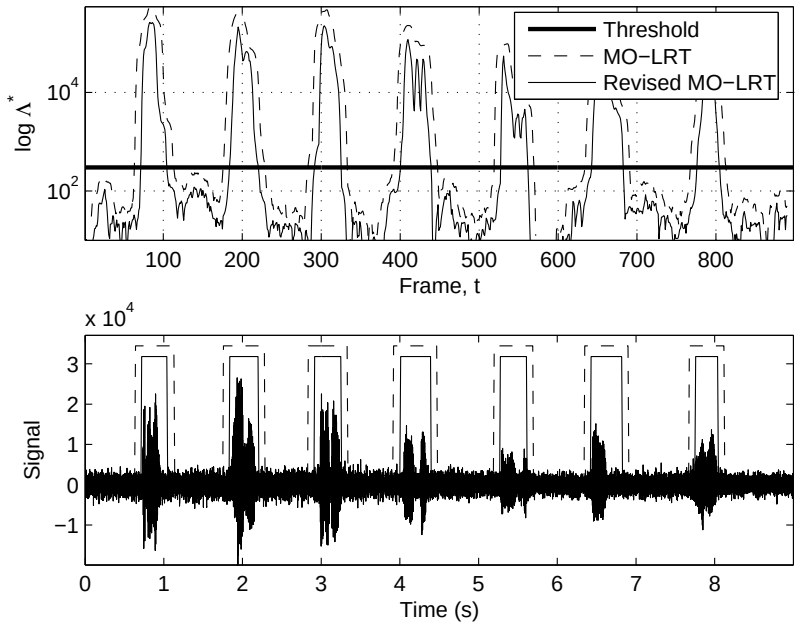


Figura 2. *Comparison between the original MO-LRT and the revised MO-LRT for VAD in high noise car environment.*

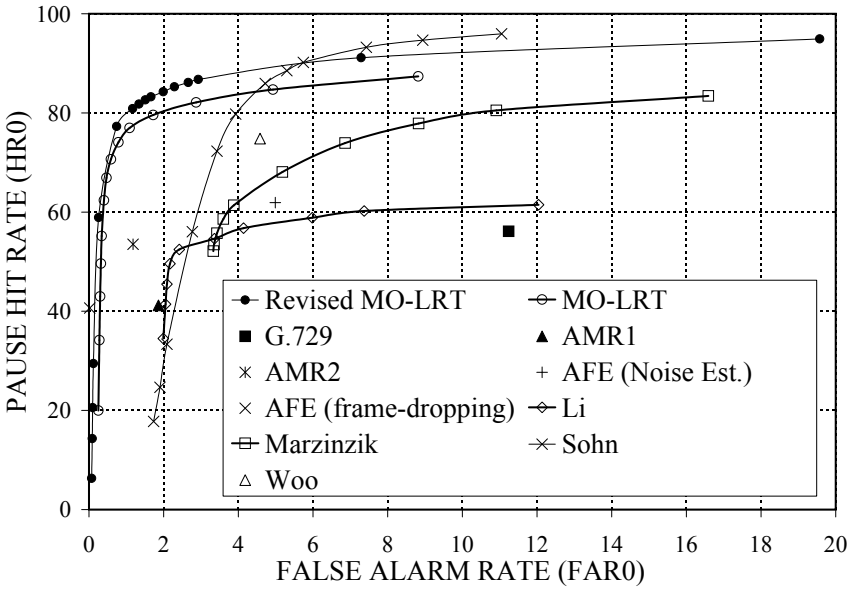


Figura 3. *ROC curves in quiet noise conditions (stopped car and engine running) and close talking microphone.*

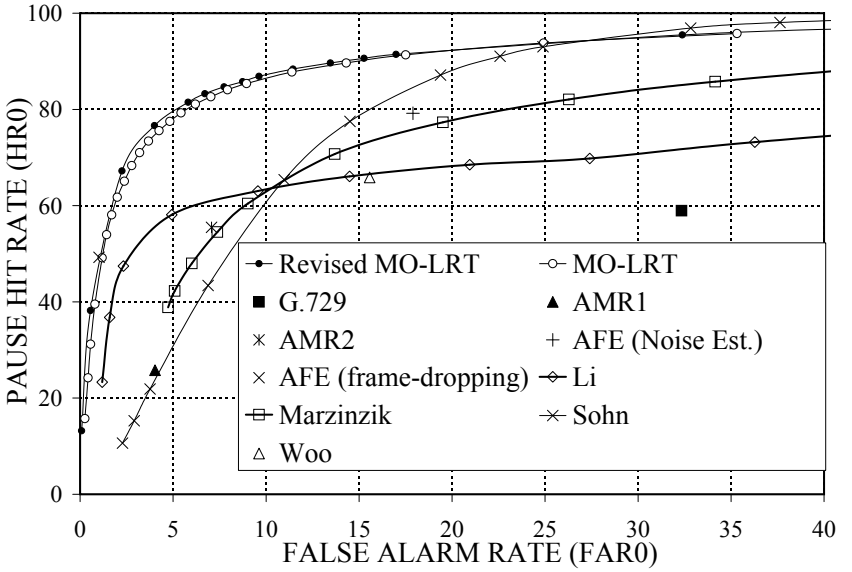


Figura 4. *ROC curves in high noise conditions (high speed over a good road) and distant talking microphone.*

Fig. 3 and 4 shows the non-speech hit rate (HR0) versus the false alarm rate (FAR0=1-HR1, where HR1 denotes the speech hit rate) for recordings from the distant microphone and under quiet and high noise conditions, respectively. The results show that the revised MO-LRT method yields better results than the previous method. These improvements are obtained by the robustness of the decision rule and by removing the implicit hangover found in the previous method and developing a more suitable design. It can be concluded from figures 3 and 4 that the proposed algorithm also outperforms a number of standardized VAD methods including the ITU-T G.729 [2], ETSI AMR (opts. 1 and 2) [3] and the ETSI Advanced Front-End (AFE) [4] for distributed speech recognition (DSR), as well as other recently published VAD methods [5, 13, 14, 12]. The best results are obtained for N= 8 while increasing the number of observations over this value reports no additional improvements. In particular, the proposed VAD outperforms the Sohn's VAD [5], that assumes a single observation in the decision rule and a HMM-based hangover mechanism.

5. CONCLUSIONS

This paper revises a multiple observation likelihood ratio test for voice activity detection in noisy environments. The new approach not only evaluates the two hypothesis consisting on all the observations to be speech or non-speech, but all the possible hypothesis defined over the individual observations. The revised statistical test exhibits the same smoothing process and reduced variance of the decision variable with the benefit of being suitable for a more effective hangover mechanism development. The experimental results showed a high speech/non-speech discrimination accuracy over a wide range of SNR conditions and significant improvements over standardized VADs such as ITU-T G.729, ETSI AMR and ETSI AFE, as well as other publicly available approaches.

6. REFERENCES

[1] A. Benyassine, E. Shlomot, H. Su, D. Massaloux, C. Lamblin, y J. Petit, "ITU-T Recommendation G.729 Annex B: A silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 64–73, 1997.

[2] ITU, "A silence compression scheme for G.729 optimized for terminals conforming to recommendation V.70," *ITU-T Recommendation G.729-Annex B*, 1996.

[3] ETSI, "Voice activity detector (VAD) for Adaptive Multi-Rate (AMR) speech traffic channels," *ETSI EN 301 708 Recommendation*, 1999.

[4] ETSI, "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms," *ETSI ES 201 108 Recommendation*, 2002.

[5] J. Sohn, N. S. Kim, y W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 16, no. 1, pp. 1–3, 1999.

[6] Y. D. Cho y A. Kondoz, "Analysis and improvement of a statistical model-based voice activity detector," *IEEE Signal Processing Letters*, vol. 8, no. 10, pp. 276–278, 2001.

[7] J. Ramírez, José C. Segura, C. Benítez, L. García, y A. Rubio, "Statistical voice activity detection using a multiple observation likelihood ratio test," *IEEE Signal Processing Letters*, vol. 12, no. 10, pp. 689–692, 2005.

[8] A. Sangwan, W.P. Zhu, y M.O. Ahmad, "On the competitive neyman-pearson approach for composite hypothesis testing and its application in voice activity detection," in *Proc. of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2006, vol. 3, pp. 301–304.

[9] Y. Ephraim y D. Malah, "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. ASSP-32, pp. 1109–1121, 1984.

[10] A. Moreno, L. Borge, D. Christoph, R. Gael, C. Khalid, E. Stephan, y A. Jeffrey, "SpeechDat-Car: A Large Speech Database for Automotive Environments," in *Proceedings of the II LREC Conference*, 2000.

[11] J. Ramírez, J. C. Segura, M. C. Benítez, A. de la Torre, y A. Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech Communication*, vol. 42, no. 3-4, pp. 271–287, 2004.

[12] M. Marzinzik y B. Kollmeier, "Speech pause detection for noise spectrum estimation by tracking power envelope dynamics," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 6, pp. 341–351, 2002.

[13] K. Woo, T. Yang, K. Park, y C. Lee, "Robust voice activity detection algorithm for estimating noise spectrum," *Electronics Letters*, vol. 36, no. 2, pp. 180–181, 2000.

[14] Q. Li, J. Zheng, A. Tsai, y Q. Zhou, "Robust endpoint detection and energy normalization for real-time speech and speaker recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 3, pp. 146–157, 2002.

GRAND CHALLENGES IN SPEECH RECOGNITION

Alex Acero

Microsoft Research.
Redmond, WA USA.

RESUMEN

Today, users can dial a number by simply speaking the name of the person they would like to contact, or find out whether a flight by calling a number and talking to a machine. But most people still haven't used speech technology at all or use it only infrequently, and those who run into an automated call center often request an operator or simply hang up. In the future, speech technology will be a mainstream way of interaction with software, devices and services, but advances in the state-of-the-art will be needed if we are to achieve that vision. Some of the challenges include making speech recognizers more robust to background noise and varying contexts, and designing user interfaces that are efficient and natural. In this talk, I will describe those grand challenges and some of the progress we have made at Microsoft Research. To combat background noise I will describe microphone arrays, as well as bone-conducting microphones that capture the vibration of the user's skin. I will also describe our efforts in building natural language understanding interfaces, and show tools for rapid application development. Finally I will illustrate the talk with examples of how to build speech interfaces, including multimodality for handheld devices.

Biografía: Alex Acero

Dr. Acero is Research Area Manager at Microsoft Research, overseeing natural language processing, communication, multimedia, and speech technologies. He joined Microsoft Research, Redmond, in 1994. He became Senior Researcher in 1996, manager of the speech research group in 2000, and Research Area Manager in 2005. Prior to Microsoft, Dr. Acero worked in Apple Computer's Advanced Technology Group, and Telefonica I+D. Dr. Acero is currently an affiliate Professor of Electrical Engineering at University of Washington. Dr. Acero is author of the books Acoustical and Environmental Robustness in Automatic Speech Recognition (Kluwer, 1993) and Spoken Language Processing (Prentice Hall, 2001), has written invited chapters in 3 edited books and over 120 technical papers. He holds 19 US patents. His research interests include speech recognition, synthesis and enhancement, speech denoising, language modeling, spoken language systems, statistical methods and machine learning, multimedia signal processing, and multimodal human-computer interaction. Dr. Acero is a Fellow of IEEE and 2006 Distinguished Lecturer for the IEEE Signal Processing Society. He was member of the board of governors of the IEEE Signal Processing Society between and 2003 and 2005. Dr. Acero served on the Speech Technical Committee of the IEEE Signal Processing Society between 1996 and 2002, chairing the committee in 2000-2002. He was Publications Chair of ICASSP98, Sponsorship Chair of the 1999 IEEE Workshop on Automatic Speech Recognition and Understanding, and General Co-Chair of the 2001 IEEE Workshop on Automatic Speech Recognition and Understanding. He's served as Associate Editor for Signal Processing Letters and is presently Associate Editor for IEEE Transactions of Speech and Audio Processing and member of the editorial board of Computer Speech and Language. Alex Acero received a Master's degree from the Polytechnic University of Madrid in 1985, another Master's degree from Rice University in 1987, and a Ph.D. degree from Carnegie Mellon University in 1990, all in Electrical Engineering.

A GOOD GESTURE: EXPLORING NONVERBAL COMMUNICATION FOR ROBUST SLDSs

Beatriz López Mencía[1], Álvaro Hernández Trapote[1], David Díaz Pardo de Vera[1], Doroteo Torre Toledano[2], Luis A. Hernández Gómez[1], Eduardo López Gonzalo[1]

[1] Grupo de Aplicaciones del Procesado de Señales, E.T.S.I.T, Universidad Politécnica de Madrid

[2] Escuela Politécnica Superior, Universidad Autónoma de Madrid

ABSTRACT

In this paper we propose a research framework to explore the possibilities that state-of-the-art embodied conversational agents (ECAs) technology can offer to overcome typical robustness problems in spoken language dialogue systems (SLDSs), such as error detection and recovery, changes of turn and clarification requests, that occur in many human-machine dialogue situations in real applications. Our goal is to study the effects of nonverbal communication throughout the dialogue, and find out to what extent ECAs can help overcome user frustration in critical situations. In particular, we have created a gestural repertoire that we will test and continue to refine and expand, to fit as closely as possible the users' expectations and intuitions, and to favour a more efficient and pleasant dialogue flow for the users. We also describe the test environment we have designed, simulating a realistic mobile application, as well as the evaluation methodology for the assessment, in forthcoming tests, of the potential benefits of adding nonverbal communication in complex dialogue situations.

1. INTRODUCTION

Spoken language dialogue systems and embodied conversational agents are being introduced in a rapidly increasing number of Human-Computer Interaction (HCI) applications. The technologies involved in SLDSs (speech recognition, dialogue design, etc.) are mature enough to allow the creation of trustworthy applications. However, robustness problems still arise in concrete limited dialogue systems because there are many error sources that may cause the system to perform poorly [1].

At the same time, embodied conversational agents (ECAs) are gaining prominence in HCI systems, since they make for more user-friendly applications while increasing communication effectiveness. There are many studies on the effects –from psychological to efficiency in goal achievement– ECAs have on users of a variety of applications (see [2] and [3]), but still very few (see [4]) on the impact of ECAs in directed dialogue situations where robustness is a problem.

We propose looking into the effects of adding an ECA to a concrete spoken dialogue system, and the potential benefits this may have, particularly regarding various difficult dialogue situations already identified by various leading authors in the field ([5] and [6]). This paper outlines the main elements of a research framework we have designed for these purposes.

Our research group is paying particular attention to videotelephony applications. We may now consider incorporating ECAs onto the new visual channel. Videotelephony has its own peculiarities which it may be relevant to take into account when developing ECAs for them (for instance, screen space is more limited).

2. HOW ECAS CAN BE USEFUL

There are many nonverbal elements of communication in everyday life that are important because they convey a considerable amount of information and qualify the spoken message, sometimes even to the extent that what is meant is actually the opposite of what is said. Showing objects, types of behaviour, mood, reactions, emotions and pointing towards something in the referential context (deictic gestures) are some of the functions of nonverbal language, which carries a great amount of semantic content related to people's attitudes and intentions in interaction processes (see [7]).

ECAs offer the possibility to combine several communication modes such as speech and gestures, making it possible, in theory, to create interfaces with which human-machine interaction is much more natural and comfortable. Despite the fact that we are still a long way from understanding how best to incorporate nonverbal communication to improve human-machine dialogue, ECAs are already being employed to improve interaction (see [8]). These are some situations in which an ECA could have a positive effect:

- Efficient turn management: the body language and expressiveness of agents are important not only to reinforce the spoken message, but also, as Cassell points out [2], to regulate the flow of the dialogue.
- Improving error recovery: the process of recognition error recovery usually leads to a certain degree of user frustration (see [9]). ECA's may help reduce frustration, and by doing so make error recovery more effective [10]. Indeed, it is common, once an error occurs, to enter in an error spiral, in which the system is trying to recover, the user gets ever more frustrated, and this frustration interferes in the recognition process (since, for example, users often repeat their previous utterance in a way that the system is less likely to understand), making the situation worse [11].
- Correct understanding of the state of the dialogue: a common problem in dialogue systems is that the user doesn't know whether or not the process is working normally [12].

This sometimes leads the dialogue to error states that could be avoided. The expressive capacity of ECAs could be used to reflect the state the system takes the dialogue to be in.

Our research framework, comprising a dialogue and ECA behaviour scheme, has been designed with these typical spoken dialogue system problems in mind, so that we may study the effect of an ECA in a variety of dialogue situations. In the following section we identify the main stages the dialogue can go through and we associate ECA behaviour and presentation strategies with each of them so that we may have a starting point with which to test real user-dialogue system interactions.

3. DIALOGUE STAGES AND ECA BEHAVIOUR

A variety of studies have been carried out on behavioural strategies for embodied conversational agents ([13], [14], [15], [16], [17]), which deal with behaviour in hypothetical situations and in terms of the informational goals of each particular interaction (be it human-human or human-machine). On our part, we will focus our attention on the overall dynamics in dialogue systems, focussing specifically on typical robustness problems, and how to favour smooth sailing through the different stages of the dialogue. Certain situations, combined with user expectations, can be sources of frustration, which negatively affects system robustness, and need to be considered with particular care, from dialogue initiation (welcoming messages) to termination.

We draw from existing research undertaken to try to understand the effects different gestures displayed by ECAs have on people, and we apply this knowledge to a real dialogue system. In Table 1 we show the basic set of gestures we are using as a starting point. They are based mainly on gestures described in [2] and [14], and on recommendations in [5], [15], [16], [17] and [18], to which we have added some suggestions of our own.

3.1. Initiation

The first contact users have with the system is through a welcome message at dialogue initiation. The inclusion of an ECA at this stage "humanises" the system [19]. This may predispose the user to be more forgiving when a problem arises in the dialogue. Interaction with a humanlike character may, however, induce in the user false expectations about the system, since she will tend anthropomorphisise it and believe it can deal with social interactions on a level similar to that of humans [20].

This is a problem, first because once a user has such high expectations the system can only end up disappointing her, and secondly because the user will tend to use more natural (and thus complex) communication, which the system is unable to handle, and the experience will ultimately be frustrating.

On the other hand, especially in the case of new users, contact with a dialoguing animated character may have the effect that the user's level of attention to the actual information that is being given is reduced (see [21] and [22]). Thus the goal is to present a human-like interface that is, at the same time, less striking and thus less distracting at first contact, and one that clearly "sets the rules" of the interaction and makes sure that the user keeps it framed within the capability of the system.

We have designed a welcome gesture for our ECA based on the recommendations in [17], to test whether or not it fosters a sense of ease in the user and helps her concentrate on the task at hand. Playing with the zoom, the size and the position of the ECA on the screen may also prove to be useful to frame the communication better. For instance, the character might appear first at mid-distance, visible from the waist up, giving a general greeting, then we might move on to a close-up of the face to give a welcome message more specific to the system, and then zoom out to a full-body view where the character explains the tasks and options ahead for the user; we have added this gestural sequence to a hand wave and smile [17] for our initial tests (see Table 1).

3.2. Termination

As we mentioned in Section 2, ECAs may be employed to help improve the user's understanding of the state of the dialogue. At dialogue termination the user should understand that the dialogue is being closed. We have implemented a gesture for our ECA, taking the main part (nod and hand wave) from [14], and adding turning around and walking away movements, dimming the lights at the end (see Table 1).

3.3. Turn Management

Turn management involves two basic actions: taking turn and giving turn. Again, an ECA's gestures can be designed to help the user see that the system is either taking the turn or handing it to the user. And again, in Table 1 we show the corresponding ECA gestures we will start testing with.

Note that apart from the ECA gestures, we also play with zoom and light intensity: when it's the ECA's turn to speak the camera zooms-in slightly and the light becomes brighter, and when it's the user's turn the camera zooms out and the lights dim. The idea is that, hopefully, the user will associate each level of light intensity with each of the turn modes, and so know when she is expected to speak.

Poor turn management is a major source of errors in SLDSs. An ECA can display body language and expression to indicate turn requests and changes more clearly, reducing the amount of errors [2]. The following are some typical examples of problem situations together with further considerations about ECA behaviour that could help avoid or recover from them:

• The user is not aware that she can interrupt, or be interrupted by, the ECA (barge-in). This leads to a less efficient communication and may be testing for the user's patience. An obvious solution is to have the ECA openly remind the user that she may interrupt. Also, the ECA could take a receptive position while talking, to encourage the user to interrupt whenever she wishes. For this purpose we will try taking a fixed mid-distance position, looking directly at the user.

• The user tries to interrupt at a point at which the barge-in feature is not active (deactivating barge-in can be useful in situations in which it is crucial for the user to listen to the system). If this happens the system does not process what the user has said, and when the system finally returns to listening mode there is silence from both parts: the system expects input from the user, and the user expects an answer. Often both finally break the silence at the same time and the cycle begins again, or, if the system caught part of the user's utterance, a recognition error will most likely occur and the system will fall into a recognition error recovery subdialogue that the user does not expect. To help avoid such faulty events the ECAs

demeanour should indicate as clearly as possible that the user is not being listened to at that particular moment. Speaking while looking away, perhaps at some object, and absence of attention cues (such as nodding) are possible ways to show that the user is not expected to interrupt the ECA. Since our present dialogue system produces fairly short utterances for the ECA, we are somewhat limited as to the active strategies to build into the ECA's behaviour. However, there are at least three cues the user could read to realise that the system didn't listen to what she said. The first is the fact that the system carries on speaking, ignoring the user's utterance. Second, at the end of the system's turn the ECA will perform a specific give-turn gesture. And third, after giving the turn the ECA will remain still and silent for a few seconds before performing a waiting gesture (leaning back slightly with her arms crossed, shifting the body weight from one leg to another; see Table 1). In addition, if the user still remains silent after yet another brief waiting period the system will offer help. It will be interesting to see at which point users realise that the system didn't register their utterance.

• A similar situation occurs if the Voice Activity Detector (VAD) fails and the system doesn't capture the user's entire utterance, or when the user simply doesn't say anything when she is expected to ("no input"). Again, both system and user end up waiting for each other to say something. And again, the strategy we use is to have the ECA display a waiting posture.

• It can also happen that the user doesn't speak but the VAD "thinks" she did, perhaps after detecting some background noise (a "phantom input"). The dialogue system's reaction to something the user didn't say can cause surprise and confusion in the user. Here the visible reactions of an ECA might help the user understand what has happened and allow her to steer the dialogue back on track.

3.4. Recognition Confidence Scheme

Once the user utterance has been recognised, information confirmation strategies are commonly used in dialogue systems. Different strategies are taken depending on the level of confidence in the correctness of the user locution as captured by the speech recognition unit (see [18]). Our scheme is as follows:

• High confidence: if recognition confidence is high enough to safely assume that no error has occurred, the dialogue strategy is made more fluent, with no confirmations being sought by the system.

• Intermediate confidence: the result is regarded as uncertain and the system tries implicit confirmation (by including the uncertain piece of information in a question about something else.) This, combined with a mixed initiative approach, allows the user to correct the system if an error did occur.

• Low confidence: in this case recognition has probably failed. When this happens the dialogue switches to a more guided strategy, with explicit confirmation of the collected information and no mixed initiative –*i.e.*, the user has no freedom and can only follow the instructions given by the system. The user's reply may confirm that the system understood correctly, in which case the dialogue continues to flow normally, or, on the other hand, it may show that there was a recognition error. In this case an error recovery mechanism begins.

In addition to the dialogue strategies, ECAs could also be used to reflect in their manner the level of confidence that the system has understood the user, in accordance with the confirmation dialogue strategies

While the user speaks, our ECA will, if the recognition confidence level is high, nod her head [14], smile and have her eyes fully open to give the user feedback that everything is going well and the system is understanding. If, on the other hand, confidence is low, in order to make it clearer to the user that there might be some problem with recognition and that extra care should be taken, an option might be for the ECA to gesture in such a way as to show that she isn't quite sure she's understood but is making an effort to. We have attempted to create this effect by having the ECA lean her head slightly to one side [18], stop smiling and mildly squint. Our goal, once again, is to find out whether these cues do indeed help users realise what the situation is. This is especially important if it helps to avoid the well known problem of falling into error spirals when a recognition error occurs in a spoken dialogue system [24].

In the case of intermediate recognition confidence followed by a mixed initiative strategy involving implicit confirmation, specific gestures could also be envisaged. We have chosen not to include specific gestures for these situations in our first trials, however, so as not to obscure our observations for the high and low confidence cases. A neutral stance for the intermediate confidence level should be a useful reference against which to compare the other two cases.

3.5. Recognition problems

There are basically four types of problems related with recognition. In subsection 3.3 we already talked about no input and phantom input situations. Now we will consider those situations in which the system finds the user's utterance incomprehensible (no-match situations) and those in which the system gets the user's message wrong (recognitions errors)

When a no-match occurs there are two ways in which an ECA can be useful. First, what the character should say must be carefully pondered to ensure that the user is aware that the system didn't understand what she said and that the immediate objective is to solve this particular problem. This knowledge can make the user more patient with the system and tolerate better the unexpected lengthening of the interaction [25]. Second, the ECAs manner should try to keep the user in a positive attitude. A common problem in no-match and error recovery situations is that the user becomes irritated or hyperarticulates in an attempt to make herself understood, which in fact increases the probability of yet another no-match or a recognition error. This we should obviously try to avoid. The ECA behaviour strategy we will test in no-match situations is to have the character lean towards the camera and raise her eyebrows (the idea being to convey a sense of surprise coupled with friendly interest). We have based our gesture on one given in [26]. Figure 1 shows an example dialogue sequence including the association between the different dialogue strategies and the ECA gesture sequences after a user's utterance.

If the user points out to the system that there has been a recognition error in a way that gives the correct information at the same time, then the ECA will confirm the corrected information with special emphasis in speech and gesture. For this purpose we have designed a beat gesture with both hands (see Table 1).

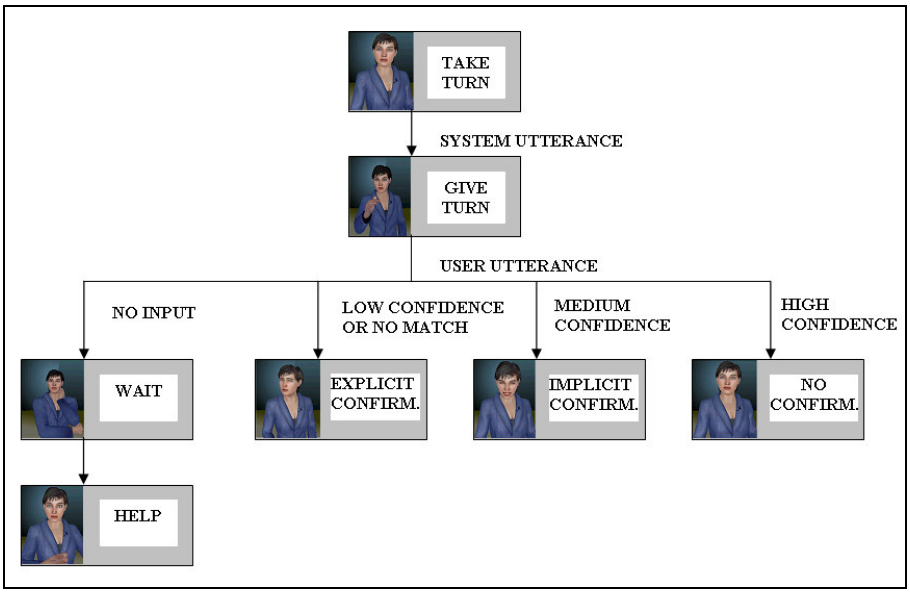


Figure 1. *Dialogue strategies and related gesture sequences.*

3.6. Help offers and requests

It will be interesting to see whether the fact that help is offered by an animated character (the ECA) is regarded by users to be more user-friendly than otherwise. If users feel more comfortable with the ECA, perhaps they will show greater initiative in requesting help from the system; and when it is offered by the system (when a problem situation occurs), the presence of a friendly ECA might help control user frustration.

While the ECA is giving the requested information, she will perform a beat gesture with both hands for emphasis, and she will also change posture. The idea is to see whether this captures the interest of the user, makes her more confident and the experience more pleasant or, on the contrary, it distracts the user and makes help delivery less effective.

Dialogue stage	*ECA behaviour (movements, gestures and other cues)*
Initiation (welcome)	1. Welcoming message: Look at the camera, smile, wave hand 2. Explanation of the task: Zoom in 3. Zoom out, lights dim
Termination	Look forward, nod, wave hand. Turn around, walk away, dim light.
Give turn	Look directly at the user, raise eyebrows, extending open hand towards the user Camera zooms out. Lights dim.
Take turn	Look directly at the user, change posture Camera zooms in. Light gets brighter.
Wait	Slight leaning back, one arm crossed and the other touching the cheek shift of body weight
Help	Beat gesture with the hands. Change posture
Error recovery with correction	Lean towards the camera, beat gesture
Confirmation (high confidence)	Nod, smile, eyes fully open
Confirmation (low confidence)	Slight leaning of the head to one side, stop smiling, mildly squint

Table 1. *Gesture repertoire for the main dialogue stages*

4. EVALUATION ENVIRONMENT

In order to test the various hypotheses put forward in the previous section we have developed an evaluation environment that simulates a realistic mobile videotelephony application (Figure 2) that allows users to remotely check the state (*e.g.*, on/off) of several household devices (lights, heating, etc.).Our dialogue system incorporates mixed initiative, error recovery subdialogues, context-dependent help and the production of guided or flexible dialogues according to the confidence levels of the speech recogniser.

Our environment uses Nuance Communications' speech recognition technology (www.nuance.com). The ECA character has been designed by Haptek, and we have integrated it with our dialogue system, which has required careful planning, designing and programming of all the gesture sequences (each involving a chain of basic movements) using the tools provided with the ECA software (www.haptek.com). (Internally, the ECA is a finite state machine that can be programmed using an editor included in the software.)

Figure 2. *Interface for a mobile device*

Our testing platform allows the collection of the three types of information we well need in the evaluation stage: objective system effectiveness and efficiency parameters (recorded automatically, directly from the users' interaction

with the dialogue system, subjective) [27], users' responses to evaluation questionnaires and video recordings of the users interacting with the system.

With all the gathered information we will generate a multimodal evaluation corpus to make the data available to other research groups and to allow comparative analysis between groups. The corpus will contain the *video* recordings of the users' interactions, annotated following standard procedures [28]; annotations of the dialogue states that were generated, following the guidelines in [27]; and *data* comprising the information collected in the questionnaires, which will cover aspects such as frustration, motivation and overall satisfaction [29].

5. TEST DESIGN

We have designed a test plan (subject recruitment, tasks, situations, etc.) with three groups of users: two interacting with the system *with* the ECA and the other without the ECA. For one of the groups with ECA we have programmed the agent with a number of context-specific gestures (i.e., gestures designed to carry or stress information, to capture the user's attention or to make meanings clearer) and a relatively high level of empathic expression (for instance, smiling). The other group will have an ECA with a relatively low level of expression ("neutral" empathy) and which makes no context-specific gestures (for instance, her hands never move). The idea is that the "neutral" ECA act as a reference to compare, on the one hand, with the group without the ECA to see what effects (on performance and user impressions) the sole presence of an ECA might have, and on the other, with the group with the expressive ECA, to study the effects of the different gestures and the overall enhanced expressiveness. The neutral ECA could thus help distinguish between observations related with gestures and expression and those related with ECA presence.

The subjects will carry out the tests in a silent and isolated place so that recognition errors due to ambient noise may be avoided and privacy secured. Before starting the test the subjects will fill out a questionnaire to gather the subjective impressions and expectations of the user. A second subjective questionnaire is presented at the end of the test to see to what extent the users' expectations were met, and how error recovery affected his/her satisfaction with the system. We will also observe users' subjective impression of success (*i.e.*, whether they believe they have carried out the interaction task successfully). In this questionnaire we include videos of the main gestures that the ECA performs at specific points of the dialogue to analyze the users' impressions for each of the gestures, and to find out whether or not they interpret the meaning of the gestures in the intended manner. By means of these questionnaires we hope to be able to validate and, if necessary, modify the nonverbal communication elements that have been introduced throughout the dialogue. (As we have seen, the gestures we have implemented are either adapted from some proposed by leading researchers in the field, or of our own creation. In either case we hope the method of data collection and analysis we have designed will prove useful in defining and refining the gesture repertoire of ECAs in dialogue systems.)

6. CONCLUSIONS AND FUTURE RESEARCH

The absence of studies on real dialogue situations with embodied conversational agents has motivated us to propose a research framework to evaluate the effects of their inclusion in automated dialogue systems, particularly in problem situations.

We have identified a range of problem situations that may arise in dialogue systems, and defined various strategies of ECA use to improve user-machine interaction throughout the whole dialogue.

We have completed the creation of an experimental framework described above to validate, discard or change the proposed strategies, and we are currently in a position to begin testing. The testing sessions will allow us to gather information that will make up a multimodal corpus for analysis and further research.

In future experiments we will attempt to go one step further and analyse how empathic emotions vs. self-oriented behaviour (see [3]) may affect the resolution of a variety of dialogue situations. To this end we plan to design ECA prototypes that incorporate specific emotions, hoping to learn how best to connect empathically with the user, and what effects this may have on dialogue dynamics and the overall user perception of the system.

Conversely, we plan to design tests to study the reactions of users to the emotional behaviour of the ECA, as a first step to modelling different types of users (*e.g.*, extroverted/introverted, patient/irritable, etc.).

ACKNOWLEDGEMENTS

This work has been possible thanks to the support grant received from project TIC2003-09068-C02-02 of the Spanish Plan Nacional de I+D.

REFERENCES

[1] Boyce, S. J., Spoken natural language dialogue systems: user interface issues for the future. In Human Factors and Voice Interactive Systems. D. Gardner-Bonneau Ed. Norwell, Massachusetts, Kluwer Academic Publishers: 37-62, (1999).

[2] Bickmore, T., Cassell, J., Van Kuppevelt, J., Dybkjaer, L. and Bernsen, N. (eds.), *Natural, Intelligent and Effective Interaction with Multimodal Dialogue Systems*, chapter Social Dialogue with Embodied Conversational Agents. Kluwer Academic, 2004.

[3] Brave, S, Nass, C, Hutchinson, K, Computers that care: investigating the effects of orientation of emotion exhibited by an embodied computer agent. Int. J. Human-Computer Studies, Nr. 62, Issue 2, pp. 161-178, (2005).

[4] Bell, L. and Gustafson, J., Child and Adult Speaker Adaptation during Error Resolution in a Publicly Available Spoken Dialogue System. Proceedings of Eurospeech 03, Geneve, Schweiz, 2003.

[5] Cassell J., Thorisson K.R., The power of a nod and a glance: envelope vs. emotional feedback in animated conversational agents. Applied Artificial Intelligence, vol.13, pp.519-538, (1999).

[6] Cassell, J. and Stone, M., Living Hand to Mouth: Psychological Theories about Speech and Gesture in Interactive Dialogue Systems. *Proceedings of the AAAI 1999 Fall Symposium on Psychological Models of Communication in Collaborative Systems*, pp. 34-42. November 5-7, North Falmouth, MA, 1999.

[7] Krauss, R. M., Chen, Y., and Chawla, P. Nonverbal behavior and nonverbal communication: What do conversational hand gestures tell us? In M. Zanna (Ed.),

Advances in experimental social psychology (pp. 389-450). San Diego, CA: Academic Press, 1996.

[8] Massaro, D. W., Cohen, M. M., Beskow, J., and Cole, R. A., Developing and evaluating conversational agents. In *Embodied Conversational Agents* MIT Press, Cambridge, MA, 287-318, (2000).

[9] Oviatt, S. & VanGent R., Error resolution during multimodal humancomputer interaction. Proc. International Conference on Spoken Language Processing, 1 204-207, (1996).

[10]Hone, K., Animated Agents to reduce user frustration, in *The 19th British HCI Group Annual Conference*, Edinburgh, UK, 2005.

[11]Oviatt, S., MacEachern, M., and Levow, G., Predicting hyperarticulate speech during human-computer error resolution. Speech Communication, vol.24, 2, 1-23, (1998).

[12] Oviatt, S. Interface techniques for minimizing disfluent input to spoken language systems. In Proc. CHI'94 (pp. 205-210) Boston, ACM Press, 1994.

[13] Poggi, I., How to decide which gesture to make according to our goals and our contextual knowledge. Paper presented at Gesture Workshop 2001 London 18th-20th April, 2001.

[14] Cassell, J., Bickmore, T., Vilhjálmsson, H., and Yan, H., More than just a pretty face: affordances of embodiment. In *Proceedings of the 5th international Conference on intelligent User interfaces*, 2000.

[15]Cassell, J., Nakano, Y. I., Bickmore, T. W., Sidner, C. L., and Rich, C., Non-verbal cues for discourse structure. In *Proceedings of the 39th Annual Meeting on Association For Computational Linguistics*, 2001.

[16] Chovil, N. Discourse-Oriented Facial Displays in Conversation. Research on Language and Social Interaction, 25, 163-194, 1992.

[17] Kendon, A. *Conducting interaction: patterns of behavior in focused encounters*, Cambridge University Press, 1990.

[18] San-Segundo, R., Montero, J.M., Ferreiros, J., Córdoba, R., Pardo., J.M., Designing Confirmation Mechanisms and Error Recover Techniques in a Railway Information System for Spanish. SIGDIAL 2001 WORKSHOP. Septiembre 1-2, 2001. Aalborg (Dinamarca).

[19] Oviatt, S., and Adams, B. Designing and evaluating conversational interfaces with animated characters. Embodied conversational agents, MIT Press: 319-345, (2000).

[20] Keeling, K., Beatty, S., McGoldrick, P., and Macaulay, L., *Face Value? Customer Views of Appropriate Formats for Embodied Conversational Agents (ECAs) in Online Retailing*. In Proceedings of the Proceedings of the 37th Annual Hawaii international Conference on System Sciences (Hicss'04), 2004

[21]Schaumburg, H., Computers as tools or as social actors? the users' perspective on anthropomorphic agents. *International Journal of Cooperative Information Systems. 10, 1, 2,* 217-234, (2001).

[22]Catrambone, R, Anthropomorphic agents as a user interface paradigm: Experimental findings and a framework for research. In: *Proceedings of the 24th Annual Conference of the Cognitive Science Society* (pp. 166-171), Fairfax, VA, August 2002.

[23] Corradini, A., Mehta, M. and Charfuelan, M., *Interacting with an Animated Conversational Agent in a 3D Graphical Setting*. In: Proceedings of the Workshop on Multimodal Interaction for the Visualization and Exploration of scientific data at the International Conference on Multimodal Interfaces (ICMI'05), October 3-6, Trento (Italy), pp. 63-70, 2005

[24] Bulyko, I., Kirchhoff, K., Ostendorf, M., Goldberg, J., Errorcorrection detection and response generation in a spoken dialogue system. Speech Communication 45, 271-288, (2005).

[25] Goldberg, J., Ostendorf, M., Kirchhoff, K., The impact of response wording in error correction subdialogs, In *EHSD-2003*, 101-106, (2003).

[26] Fagerberg, P., Ståhl, A., Höök, K. (2003). Designing Gestures for Afective Input: an Analysis of Shape, Effort and Valence. In *Proceedings of Mobile Ubiquitious and Multimedia,* MUM 2003, Norrköping, Sweden.

[27] Möller, S., Parameters for quantifying the interaction with spoken dialogue telephone services. ", In *SIGdial6-2005*, 166-177, (2005).

[28] Herramienta de anotación ANVIL: http://www.dfki.de/~kipp/anvil/description.html

[29] Walker, M.A., Litman, D.J., Kamm, C.A. and Abella, A., Evaluating spoken dialogue agents with PARADISE : Two case studies. Computer Speech and Language, 12, 317-347, (1998).

A multiple-Gaussian classifier for Language Identification using acoustic information and PPRLM scores

R. Córdoba, L.F. D'Haro, R. San-Segundo, J. Macías-Guarasa, F. Fernández, J.C. Plaza

Speech Technology Group. Dept. of Electronic Engineering. Universidad Politécnica de Madrid
E.T.S.I. Telecomunicación. Ciudad Universitaria s/n, 28040-Madrid, Spain
cordoba@die.upm.es

Abstract

We present several innovative techniques that can be applied in a PPRLM system for language identification (LID), obtaining a 61.8% relative error reduction from our base system. First, the application of a variable threshold in score computation, dependent on the average scores in the language model, provided a 35% error reduction. A random selection of sentences for the different sets and the use of silence models also improved the system. Then, to improve the classifier, we compared the bias removal technique (up to 19% error reduction) and a Gaussian classifier (up to 37% error reduction). Then, we included the acoustic score in the Gaussian classifier (2% error reduction) and increased the number of Gaussians to have a multiple-Gaussian classifier (14% error reduction). Finally, we included additional acoustic HMMs of the same language with success (18% relative improvement). We will show how all these improvements have been mostly additive.

1. Introduction

Automatic language identification (LID) has become an important issue in recent years in speech recognition systems. To do language identification, first we have to identify which factors are more critical to distinguish between languages. We can identify several factors of differentiation: the realization of allophones and sounds and information related to the sequence of allophones, which has demonstrated to be vital: some sequences of allophones do not exist in one language (or occur very little), so the identification of those sequences is crucial for LID. Another possibility is to use prosodic features as the intonation may differ drastically between languages.

The most used technique is the phone-based approach, like Parallel phone recognition followed by language modeling (PPRLM) [1][2], which classifies languages based on the statistical characteristics of the allophone sequences and has a very good performance. Another popular technique is the GMM classifier, which we will not consider here. In [3] the "GMM tokenizer" is described.

Another possibility is to base the identification on the score given by a full continuous speech recognizer. As we demonstrated in [4], the results obtained with this technique are probably the best that can be obtained, as it models both acoustic and phonetic information, together with the sequence of allophones and words, but it has some important disadvantages: a complete speech recognition system has to be trained, a lot of labeled data is needed and it would be difficult to have a real-time system for several languages.

An interesting variant of PPRLM is presented in [5] with several proposals: different ways to combine the allophone sequence information with the acoustic models, use of durations (prosodic information) and a tree-based language model. It is remarkable the integration of several sources of information.

In [6] they use PPR, include bias removal to improve the classification, and include acoustic and allophone sequence information in the classifier, using a Gaussian classifier similar to the one we propose.

This is a continuation of the work done in [2] and [7]. We are going to focus now on improving the classifier, using bias removal and a multiple-Gaussian classifier mixing acoustic and allophone sequence information. This work has been done under project INVOCA, for the public company AENA, which manages Spanish airports and air navigation systems.

2. System description

2.1. Database

We use a continuous speech database (Invoca database from now on), which consists of very spontaneous conversations between controllers and pilots. For speech recognition it is a very difficult task, noisy and very spontaneous, as in "Lufthansa four two seven nine start up approved clear to Frankfurt standard departure Somosierra one echo three six left squawk one zero two three report parking position".

We have one big drawback with the database: all speakers are native Spanish. So, many of them do not reflect all the phonetic variations in English. Besides, the controllers use to mix Spanish for greetings and goodbyes even when the rest of the sentence is in English. Also, many company/ airport names are pronounced in Spanish inside the English sentence.

For the training set, we had some 8 hours of speech for Spanish and 6 hours for English. For the validation set, we had some 1 hour for both languages and 700 sentences. We have considered sentences with a minimum of 0.5 sec., and a maximum of 10 sec., with an average duration of just 4.5 sec., which is an another important limitation in our system.

2.2. Brief description of PPRLM

The main objective of PPRLM (Parallel Phone Recognition Language Modeling) is to model the frequency of occurrence of different allophone sequences in each language. This system has two stages. First, a phone recognizer takes the speech utterance and outputs the sequence of allophones corresponding to it. Then, the sequence of allophones is used as input to a language model (LM) module. In recognition, the LM module scores the

probability that the sequence of allophones corresponds to the language. It can use several phone recognizers modeled for different languages. Interpolated n-gram language models are used to approximate the n-gram distribution as the weighted sum of the probabilities of the n-grams considered (weights α_1, α_2, and α_3 for unigram, bigram and trigram, respectively). In our case, we have considered up to trigrams.

2.3. Results presentation

In all our experiments we have obtained the results for all possible combinations of weights α_1, α_2, and α_3, in 0.1 steps. Throughout the paper we will present the results (Sentence error rate) for the average of all weight combinations (Average column in the tables) and for the best result (Minimum column). In general, best (minimum) results occur with the biggest contribution from the trigram score, reflecting that the trigram is the most discriminative feature for language identification. In all tables, we present in parenthesis the relative improvement in relation to the base system considered.

3. Initial improvements to the base system

3.1. Threshold in score computation

As the size of the database is small, there are quite a big number of trigrams that do not have enough training samples and, so, their estimates are not reliable. We tested several alternatives for LM smoothing, but the results were very similar, showing little improvement. We decided to apply a fixed threshold or additive factor to the score value, in a similar way to the variance flooring applied in HMM estimation: use as the minimum variance a fraction of the average variance. The objective is to give more importance to the allophone sequences that have a high probability in one language and, at the same time, reduce the effect of sequences that have not appeared in training. We considered two alternatives (in the log domain):

1) **n-gram specific fixed additive factor**. We propose the following formula for the score:

$$S(F) = 10\log\left(\prod_{i=0}^{N} P_i(F)\right) = -\sum_{i=0}^{N} 10\cdot\alpha_i \cdot\log\left(P_i(F) + \beta_i\right) \qquad (1)$$

where N is the order of the N-gram, α_i is the weight for the i^{th} n-gram and $P_i(F)$ is its probability. β_i is the additive factor. The optimum values were β_{uni}=0.027, β_{bi}=0.04 y β_{tri}=0.08. Obviously, it is not a nice approach as β values are too empiric.

2) **Variable additive factor**. We made the β_i dependent on the average scores in the LM ($\overline{p}_i$ for the i^{th} n-gram). λ is a smoothing factor.

$$S(F) = -\sum_{i=0}^{N} 10\cdot\alpha_i \cdot\log\left(P_i(F) + \frac{\overline{p}_i}{\lambda}\right) \qquad (2)$$

To estimate the optimum λ factor, we observed that very little differences in performance were observed using λ values between 4 and 8, which is a nice feature. In Table 1 we can see the results obtained with both alternatives: the improvement is outstanding, showing the suitability of this approach, especially for the second approach. Even though it is simple, it has been the best improvement in the experiments of this paper.

Table 1. Results for different additive factors

Thresh. technique	Average	Minimum
None	8.27	6.80
n-gram specific	7.70 (6.9%)	5.84 (14.1%)
Variable	6.26 (24.3%)	4.46 (34.3%)

3.2. Random selection of sentences / silence models

In our database, the same controller uttered a large group of sentences sequentially in the database until there was a shift change. We were afraid that our system was making some kind of speaker modeling instead of language modeling, which was confirmed by the results in Table 2. So, for similar cases we recommend to do a random selection of sentences. Besides, in our original system, we did not consider silence models in the output of the phone recognizer, but it seems that we were not estimating important trigrams which are especially relevant for language identification, e.g. 'ai-t-sil' in the word 'flight', which is extremely rare in Spanish. So, we run an experiment considering the silence models with the results shown in Table 2, with a relevant improvement.

Table 2. Results with random selection / silence models

	Average	Minimum
Original lists	6.26	4.46
Randomly selected	5.24 (16.6%)	4.24 (5.0%)
+ silence models	5.03 (4.0%)	3.92 (7.55%)

4. Bias removal in the classifier

As is described in [6], the general PPRLM approach has a flaw: there is the possibility of having a different bias in the log-likelihood score for the languages considered. This is especially relevant when the phone recognizers have a different number of units (we have 49 phonetic units for Spanish and 61 for English). The language with fewer units will have higher probabilities in the LM score, and so the classifier will tend to select that language. To eliminate this bias, two options are proposed in [6]. We have experimented with the first one: we use the original LM score minus the average of all LM scores in the training database (a language-dependent bias).

Database for bias estimation: We can divide the training database in 3 different sets: the first one to train the acoustic models, the second one to train the LMs and the third one to estimate the bias value. This could be the optimal option if the database were large enough, as all estimations are independent. The problem is that, as our database is small, all results worsened due to insufficient training data. So, we had to discard this option. Another option is to estimate the bias value in the original training sets. We considered two possibilities:

1. Estimate the bias with the LMs training list. This is the worst option: as the LMs have been estimated using this list, the bias value estimated is not reliable because it is too optimistic.

2. Estimate the bias with the acoustic models training list. Even though this data does not participate in the LM estimation, it could be a dangerous option. But we observed that the LM score distribution in this set was very similar to the score distribution in the test set. So, we followed this option.

In Table 3 we present the results obtained using bias removal in a system without the threshold described in Section

3.1. We can see an outstanding improvement, showing that this technique is effective when there is an obvious bias in the log-likelihood score as we had presumed.

Table 3. Results for bias removal

	Average	Minimum
No threshold	8.27	6.80
Bias removal	6.98 (15.6%)	5.5 (18.9%)

But we have to admit that the same technique applied to the best system so far – after the threshold technique – showed no relevant improvement, just 0-1% relative. Most probably, the additive factor compensates the bias effect.

5. Gaussian classifier

Another possibility to tackle the issue of different bias in the LM scores is to use a Gaussian classifier instead of the usual decision formula applied in PPRLM. With all the scores provided by every LM in the PPRLM module we prepare a score vector. With all the sentences in the training database we estimate the Gaussian distribution of their respective score vectors for every language. So, we will have a Gaussian distribution for each language in the system. Now, the recognized language is not the one with the largest average score. The distance between the input vector of LM scores and the Gaussian distributions for every language is computed, and the distribution which is closer to the input vector is the one selected as identified language.

Database for Gaussian estimation: For the Gaussian classifier, the same considerations as for bias removal can be made. In this case, the problem addressed is even more notorious, as we need more data to estimate a reliable Gaussian distribution than we need to estimate just the bias in the score. So, again we decided to use the acoustic models training list.

Score vector for the Gaussian classifier: As we have several scores in the PPRLM system, there are several options for the feature vector of scores:

1. Basic. Use all PPRLM scores as is (M acoustic models x N language models, 2 x 2 in our case). This would be the typical option. The problem is that these scores are quite unstable.

2. Individual scores. We then considered the possibility to model the distribution of each n-gram in the score computation for our feature vector: the score for unigram, bigram and trigram. The drawback now is the increase in dimensionality.

3. Differential scores. Instead of using absolute values, we considered differential scores, which for every sentence are computed as the difference between the score obtained by the LM of the same language of the acoustic models considered (Spa-Spa or Eng-Eng) and the score obtained by the other 'competing' language: SC0 – SC1 and SC3 – SC2 in Figure 1. So, this score can be computed both in training and testing. We also considered the differentiation between individual scores: unigram, bigram and trigram, with 6 features in total.

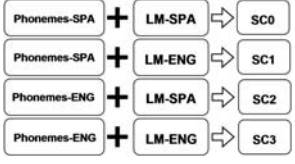


Figure 1. PPRLM Score average

We observed that these differential scores are much more homogeneous, being the result that the estimated distributions exhibit a much smaller overlap with the competing language.

In a multiple language system the proposal for the differential score would be:

$SC_{\text{current language}} - \text{Average}\ (SC_{\text{other languages}})$

In Table 4, we can see the results for the 3 techniques in a system without the Threshold described in Section 3.1. As can be seen, the results for the Basic and Individual options are similar and quite bad, probably because two facts: the great variations in score and the insufficient size of the database. Nevertheless, the results for the Differential scores are outstanding, more than 30% relative.

Table 4. Results for the Gaussian classifier

Score vector	Average	Minimum
No threshold	8.27	6.80
Basic	11.43 (-38.2%)	7.7 (-13.8%)
Individual	10.94 (-32.3%)	7.7 (-13.8%)
Differential	5.82 (29.6%)	4.3 (36.8%)

If we apply the technique with the best system so far, the minimum goes to **3.71%** (5.41% improvement), which is a smaller improvement, but are better than for the bias removal technique. Again, the improvement of the threshold technique is not additive with the Gaussian classifier. In any case, these results are a fantastic starting point, as it is easy to include acoustic information using this Gaussian classifier, and use multiple Gaussians.

5.1. Inclusion of acoustic information

One drawback in PPRLM modeling is that the basic technique only takes into account information regarding the allophone sequence. We propose the inclusion of acoustic information using our Gaussian classifier, adding new features to our score vector: the acoustic score obtained in the phone recognizers of both languages. We observed that the values of the acoustic score were not homogeneous at all. So, we decided to use again the "differential scores" idea: we used the difference between the score for the Spanish phone recognizer and the score for the English phone recognizer as feature value. So, we just have one feature in the acoustic score vector.

Database considered: Obviously, we need to estimate the acoustic score distributions using non-training data. So, the dataset chosen for this task is the LMs training list, because those sentences have not been used to train the phone models. So, we have estimated Gaussian distributions for allophone sequence scores and acoustic scores separately, as they use different lists for the estimation.

The new result using acoustic information was **3.67%** with a 2% relative improvement for the Minimum, but 13.5% for the Average. So, results show that acoustic information complements better the least robust systems.

5.2. Multiple-Gaussian classifier

One of the nicest characteristics of a Gaussian classifier is that we can grow up to multiple Gaussians to better model the distribution that represents our classes. We have used different number of Gaussians for allophone sequence score and acoustic

score, as their feature vector dimension is completely different: 6 and 1 features respectively. To increase the number of Gaussians we have followed the classical HMM modeling approaches (Gaussian splitting and Lloyd reestimation after each splitting), so we will not describe them here. In Table 5 we can see a summary of results obtained using different numbers of Gaussians for both scores.

Table 5. Multiple-Gaussian classifier

Number of Gaussians		Average	Minimum
LM	Acoustic		
1	1	4.69	3.67
2	1	4.35 (7.2%)	3.52 (4.1%)
2	2	4.14 (11.7%)	3.31 (9.8%)
3	1	**4.01 (14.5%)**	3.24 (11.7%)
3	2	4.12 (12.1%)	3.23 (12.0%)
3	3	4.06 (13.4%)	3.31 (9.8%)
4	2	4.20 (10.5%)	**3.16 (13.9%)**
4	3	4.08 (13.0%)	3.17 (13.6%)

We can extract several interesting conclusions:

- The improvements are really remarkable, up to 14% in minimum value and almost 15% in average.
- As we expected, the best system uses more Gaussians for LM score than for acoustic score.
- It is a nice feature that all systems provide better results than the mono-Gaussian system, showing that there is enough training data for the multiple-Gaussian system.
- There are better improvements in the Average value. Again, the more powerful estimation of multiple Gaussians has more relevance in the less robust systems (the ones with bigger weights for unigram and bigram).
- The difference between the Average and Minimum values has reduced drastically, showing these techniques' robustness, so the n-gram weights are less relevant.

5.3. Additional acoustic HMMs for the classifier

We considered the inclusion of new HMM models in our system, as it was quite easy with our Gaussian classifier. So far, nobody has reported the use of several models of the same language but different channel conditions in PPRLM. We had two additional acoustic models for Spanish: one based on SpeechDat, telephone noisy speech, and another one with speech recorded in quiet conditions ('Quiet'). So, both of them are quite different to the original Invoca database used so far. We wanted to test if some additional improvements could be obtained using them. These are the conclusions:

- Using them with no adaptation, results do not improve.
- Using them with task adaptation (models adapted using MAP with the Invoca training list) the improvements are remarkable: **2.60%** error rate with an 18% relative improvement.
- Only one of them is needed, the inclusion of both SpeechDat and 'Quiet' did not improve, probably because the increase of dimensionality in the feature vector causes a poor estimation.

So, it is clear that they can provide complementary information to the classifier when task adaptation is used.

5.4. Inclusion of four-grams in PPRLM

The inclusion of four-grams did not improve our system. They were poorly estimated in all cases. So, we discourage it unless for a huge database system.

6. Conclusions

We have described several improvements in a language identification system using PPRLM scores and acoustic information. The system has improved remarkably, up to **2.60%** error rate with an overall 61.8% relative improvement, especially considering that the average duration of the sentences is just 4.5 seconds. Increasing the sentence minimum duration to 2 seconds instead of 0.5 (5.3 seconds average duration) we obtain a **0.82%** error rate. So, most errors in our system come from extremely short sentences.

The application of the variable additive factor in score computation provided a significant error reduction in all cases. It even compensated the bias mismatch in the LM scores, as the results have shown.

For the classifier, we compared the bias removal technique (up to 19% error reduction) and a Gaussian classifier (up to 37% error reduction), showing that the last one provides better results and has the potential to include additional information. The use of differential scores to estimate the Gaussian distributions is also crucial for the technique.

The inclusion of acoustic score in the Gaussian classifier provided a 2% error reduction and the increase in the number of Gaussians provided an additional 14% error reduction. The inclusion of additional HMMs of the same language but different channel conditions provides a nice improvement if task adaptation is used.

7. References

[1] Zissman, M.A., "Comparison of four approaches to automatic language identification of telephone speech," IEEE Trans. Speech & Audio Proc., vol. 4(1), pp. 31-44, 1996.

[2] Córdoba, R., G. Prime, J. Macías-Guarasa, J.M. Montero, J. Ferreiros, J.M. Pardo, "PPRLM Optimization for Language Identification in Air Traffic Control Tasks". Eurospeech 2003, pp. 2685-2688.

[3] Torres-Carrasquillo, P.A., Reynolds, D.A., Deller Jr., J.R., "Language identification using Gaussian mixture model tokenization", IEEE ICASSP 2002, pp. I-757-760.

[4] Fernández, F., R. de Córdoba, J. Ferreiros, V. Sama, L. F. D'Haro, J. Macías-Guarasa. "Language Identification Techniques based on Full Recognition in an Air Traffic Control Task". ICSLP 2004, pp. II-1565-1568.

[5] Navratil, J. 2001. "Spoken Language Recognition – A Step Toward Multilinguality in Speech Processing". IEEE Trans. Speech&Audio Proc., Vol. 9(6), 2001, pp. 678-685.

[6] Ramasubramaniam, V., et al. 2003. "Language Identification using Parallel Phone Recognition". Workshop on Spoken Language Processing, India.

[7] Córdoba, R., R. San-Segundo, J. Macías, Juan M. Montero, R. Barra, L.F. D'Haro, J.C. Plaza, J. Ferreiros. 2006. "Integration of acoustic information and PPRLM scores in a multiple-Gaussian classifier for Language Identification". IEEE Odyssey 2006.

BASQUE-SPANISH-ENGLISH MACHINE TRANSLATION USING FINITE-STATE TRANSDUCERS

*Alicia Pérez, M. Inés Torres, V. Guijarrubia**

Dep. Electricity and Electronics
Faculty of Science and Technology
University of the Basque Country
manes@we.lc.ehu.es

Francisco Casacuberta

Dep. of Information Systems and Computation
Faculty of Computer Science
Technical University of Valencia
fcn@dsic.iti.es

ABSTRACT

The goals of this work are, on the one hand to develop a trilingual corpus suitable for example-based text and speech-input machine translation between English, Spanish and Basque; and on the other hand, to compare the same translation method for under two very different pair of languages: Spanish-Basque and Spanish-English.

The stochastic finite-state transducers have been trained from raw bilingual examples. As the experimental results show, they seem to be a good choice for both text and speech-input machine translation.

1. INTRODUCTION

Two main trends can be distinguished in machine translation (MT): rule-based and statistics-based. The first one is deductive and includes linguistic knowledge, whereas the second one is built on the basis of a collection of samples, so it is inductive. Nowadays some efforts are being made in order to take advantage of both information sources. For both rule-based and statistical frameworks, finite-state transducers (FST) have proved to offer a great versatility to compose with other finite-state models.

In this paper we focus on statistical models, however, we also offer some Spanish-Basque translation results obtained with *Opentrad* [1], an open-source deep-transfer MT toolkit developed by several Universities and enterprises in a Spanish challenging project. Specifically, we make use of FST, which are also the basis of *Opentrad*.

A stochastic FST (SFST) can be built for any pair of languages, whenever a representative amount of examples is available. As it is known, inductive approaches require great collection of bilingual data, that is, the same sentences translated into the languages we want to work with [2]. However, only bilingual corpora for some of the European languages are available. In this work we present a suitable trilingual corpus, in English, Spanish and Basque, completing the work initiated in [3], and letting us start with multilingual translation involving Basque language, as well as to face with speech-input MT.

1.1. Basque language

Basque is a minority language but shares official status, along with Spanish, in the Basque Country.

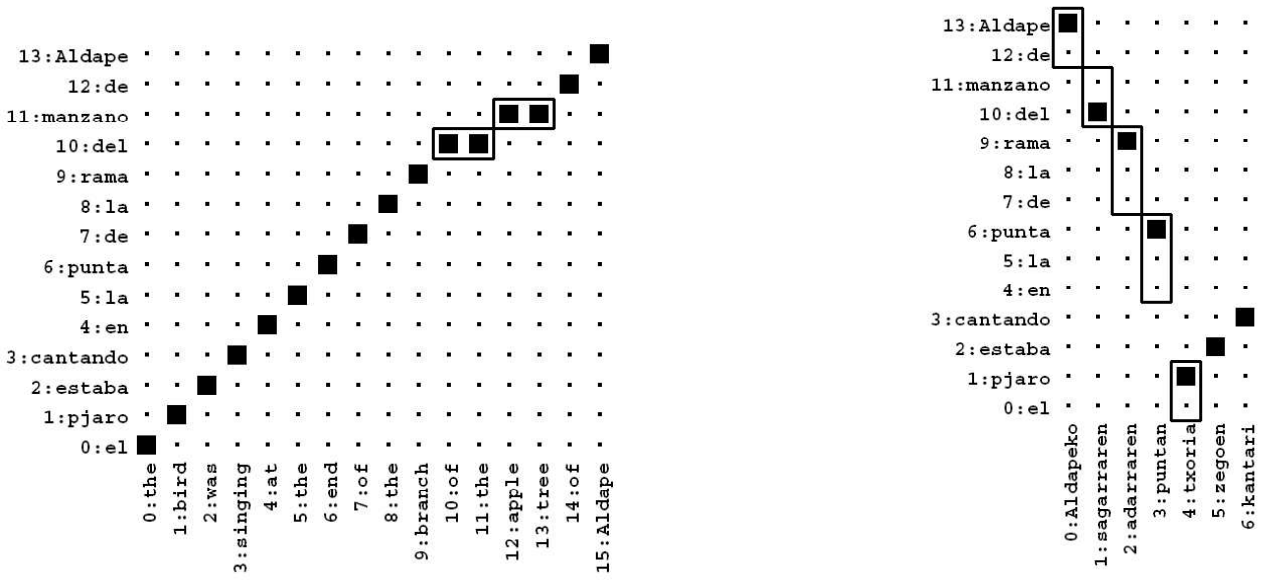
Basque is a pre-Indoeuropean language of unknown origin. Thus, with regard to etymology, Basque and Spanish are very different. On the other hand, Basque is an extremely inflected language, both in nouns and verbs. In addition, both languages present a different arrangement of the words within the sentence since, opposite to Spanish, Basque has left recursion. Figures 1(a) and 1(b) show the word arrangement through Spanish-English and Spanish-Basque alignment matrices respectively. The relationships provided by the statistical alignment model are shown by filled squares, while the linguistic ones are in hollow squares, e.g. the first Basque word is connected to the last two Spanish words, even the statistical alignment model provides only the first one.

2. STATISTICAL FRAMEWORK

Finite state transducers have proved to be useful in language processing and in automatic speech recognition systems. Recently they have also been proposed for SMT applications [4, 5]. Stochastic finite state transducers (SFST) can be automatically learnt from bilingual corpora by efficient algorithms, such as GIATI (Grammar Inference and Alignments for Transducers Inference).

The GIATI methodology is used in this work to build the translation models. Given a bilingual corpus, GIATI algorithm provides a probabilistic finite-state transducer [5].

*This work has been partially supported by The University of the Basque Country and by the Spanish CICYT, under grants 9/UPV 00224.310-15900/2004 and TIN2005-08660-C04-03 respectively.



(a) Spanish-English

(b) Spanish-Basque

Figure 1. These alignment matrices show the word ordering for a sentence in the three languages under consideration.

GIATI algorithm works as follows:

1. Given a bilingual corpus, find a monotone segmentation, and thereby, assign an output sequence to each input word, leading to the so called *extended corpus*.

2. Infer a probabilistic finite state automaton from the extended corpus. We promote the use of a *k-testable in the strict sense* (k-TSS) language model [6], instead of n-gram models, since the k-TSS models keep the syntactic constraints of the language.

3. Split the output sequence from the input word, on each edge of the automaton, getting, in this way, the finite state transducer.

Once we have the transducer and an input sentence $\mathbf{s} \in \Sigma^+$, the translation process implies searching the most likely output string $\widehat{\mathbf{t}} \in \Delta^*$ through all the possible output strings as summarized in equation (1). Where $d(\mathbf{s}, \mathbf{t})$, represents a path in the SFST, compatible both with the input sentence $\mathbf{s}$ and the output $\mathbf{t}$. Therefore, the searching criteria in the SFST deals with the joint probability of sentence pairs.

$$\widehat{\mathbf{t}} = \arg\max_{\mathbf{t}} P(\mathbf{s}, \mathbf{t}) \approx \arg\max_{\mathbf{t}} \max_{d(\mathbf{s},\mathbf{t})} P(d(\mathbf{s}, \mathbf{t})) \quad (1)$$

2.1. Two architectures for speech translation

The goal of the statistical speech translation (summarized in eq. (2)) is to find the likeliest target language string ($\mathbf{t}$) given the acoustic representation ($\mathbf{x}$) of a source language string ($\mathbf{s}$).

$$\widehat{\mathbf{t}} = \arg\max_{\mathbf{t}} P(\mathbf{t}|\mathbf{x}) = \arg\max_{\mathbf{t}} \sum_{s} P(\mathbf{t}, \mathbf{s}|\mathbf{x}) \quad (2)$$

Two architectures can be used in order to build the speech translation system: the serial and the integrated one [5]. The serial architecture consists of a text to text translator after the speech decoder, whereas the integrated architecture works as a speech recognition system which makes use of a translation model instead of the usual language model. The translation model, in fact, involves two language models: the source and the target language models. Then, once the decoder has found the best path throughout the trellis, the output-language string related to the best sequence of states is provided. Both architectures are described in detail in the following subsections.

2.1.1. Serial architecture

The speech is decoded in a conventional speech recognition system. The output string provided by the speech recognizer is the input (source) string for the text to text translator (see Fig. 2). The full process can be described in two steps:

1. Word decoding of $\mathbf{x}$ (the acoustic representation):

$$\widehat{\mathbf{s}} \approx \arg\max_{\mathbf{s}} P(\mathbf{s}) P(\mathbf{x}|\mathbf{s}) \quad (3)$$

where $P(\mathbf{s})$ is the source language model (k-TSS LM is used in this work).

2. Translation of $\widehat{\mathbf{s}}$ (the expected decodification of $\mathbf{x}$):

$$\widehat{\mathbf{t}} \approx \arg\max_{\mathbf{t}} P(\mathbf{t}|\widehat{\mathbf{s}}) = \arg\max_{\mathbf{t}} P(\mathbf{t},\widehat{\mathbf{s}}) \quad (4)$$

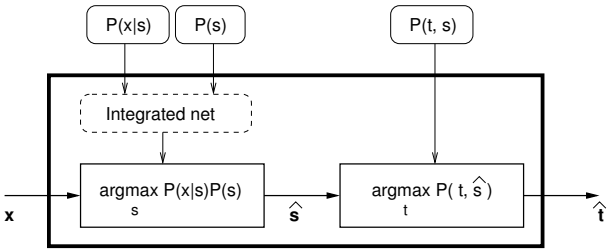


Figure 2. Serial architecture for speech translation. The output of the system is the text translation of an input speech signal. The system is supported on three knowledge sources: the acoustic model, the input language model and the translation model. The system consists of two main blocks: the speech decoder and the text to text translator. The acoustic model and the input language model are integrated into a finite state net whose output is supplied to the speech decoder to make decisions. The translator makes use of the translation model, which is also a finite state model.

In practice, $\widehat{\mathbf{s}}$ (the input sentence to the translator) is likely to be corrupted, since the speech recognition system is not ideal. Thus, we can not expect the output translation to be as close to the reference as it could be in case of a perfect input. Moreover, the weakest process is the translator, therefore we should preserve it from errors as much as possible. The recognition system alone introduces a $WER = 3.5$.

2.1.2. Integrated architecture

The strict way to deal with speech translation is:

$$\begin{aligned} \widehat{\mathbf{t}} &= \arg\max_{\mathbf{t}} \sum_{s} P(\mathbf{t},\mathbf{s}|\mathbf{x}) \\ &= \arg\max_{\mathbf{t}} \sum_{s} P(\mathbf{t},\mathbf{s}) P(\mathbf{x}|\mathbf{t},\mathbf{s}) \end{aligned} \quad (5)$$

Let's assume that the acoustic signal representation depends only on the source string, i.e. $P(\mathbf{x}|\mathbf{t},\mathbf{s})$ is independent of $\mathbf{t}$, then eq. (5) can be written as,

$$\widehat{\mathbf{t}} = \arg\max_{\mathbf{t}} \sum_{s} P(\mathbf{t},\mathbf{s}) P(\mathbf{x}|\mathbf{s}) \quad (6)$$

In practice, the sum over all possible source strings can be approximated by the maximum term involved.

$$\widehat{\mathbf{t}} \approx \arg\max_{\mathbf{t}} \max_{s} P(\mathbf{t},\mathbf{s}) P(\mathbf{x}|\mathbf{s}) \quad (7)$$

In this approach, the acoustic knowledge is introduced in the whole FST. The main feature of this approach is it's ability to carry out both the recognition and the translation at the same time. The problem is solved as the equation (7) suggests, without any more assumptions. In the integrated architecture (shown in Fig. 3), the speech recognizer and the translation system have been coupled into a unique automaton, the finite-state transducer inferred by GIATI technique. The whole system works as a speech recognizer, where the output string is the text translation, instead of the text transcription, of the input speech.

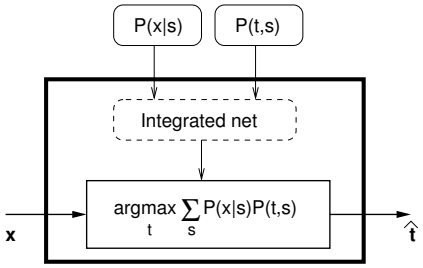


Figure 3. Integrated architecture for speech translation. The output of the system is the text translation of an input speech signal. The system is supported on two knowledge sources: the acoustic model and the translation model, both finite state models.

The strings of extended symbols are used to estimate the language model of such a recognizer, that is, the translation model of the integrated system. However, only the lexicon of the input language was transcribed into acoustic models and it was then integrated in the whole speech translation system.

The integrated automaton can be built on the basis of the text to text translator, just expanding the source word on one edge by its phonetic transcription, and each phoneme by the corresponding HMM.

3. EXPERIMENTAL RESULTS

In this section we are going to describe the corpus built during the last months, and then we show preliminary experimental results for text and speech input machine translation using the described techniques.

3.1. Task and corpus

METEUS is the weather forecast corpus composed from 28 months of daily weather forecast reports in the Spanish and Basque picked from those published in Internet. In those report pairs bilingual alignment was assured at paragraph level, but segmentation into sentences was solved by using statistical techniques, specifically *RECalign*, a greedy algorithm [7]. Afterwards each sentence was translated into English by experts, leading to a trilingual corpus aligned at sentence level.

The main features of METEUS corpus are shown in Table 1. The test set consists of 500 training independent pairs, all of them different. For speech input machine translation experiments, this test set was recorded

by 36 bilingual speakers uttering 50 sentence-pairs each, resulting in around 3.25 hours of audio signal for each language. Thus, the test set is the same for text and speech translation in order to assess the model under the same conditions.

At this point, with the Basque-Spanish sentence pairs and their counterparts in English, we conclude the work [3], and we can start experimentation with example-based machine translation involving Basque language and including English for a matter of comparison with other works of reference.

		Spanish	Basque	English
Training	Sentences	14615		
	Different	7225	7523	6634
	Words	191156	187462	195627
	Vocabulary	702	1147	498
	Average Length	13.0	12.8	13.3
Test	Sentences	500		
	Words	8706	8274	9150
	Average Length	17.4	16.5	18.3
	Perplexity (3grams)	4.8	6.7	5.8

Table 1. METEUS corpus main features.

There is a great difference in terms of vocabulary size for the three languages taken into account within the same application (see Table 1). Basque language is a highly inflected language with many different running words, whereas English is the simplest one. The reliability of the statistics over a smaller number of words with the same amount of training sentences, is likely to be higher, therefore, we expect worse probability distributions to be estimated over the model involving the Basque language.

With regard to the practical issues, let us add that the underlaying translation model has been inferred with our own k-TSS language modeling toolkit, with Witten-Bell discount for smoothing purposes. With respect to the acoustic models, the speech signal database was parameterized into 12 Mel-frequency cepstral coefficients (MFCCs) with delta (ΔMFCC) and acceleration (Δ^2MFCC) coefficients, energy and delta-energy (E, ΔE), so four acoustic representations were defined. For the speech recognition system, a total of 24 context-independent acoustic-units were used. Each phone-like unit was modeled by a typical left to right non-skipping self-loop three-state continuous hidden Markov model, with 32 Gaussians per state and acoustic representation. To train these models, a phonetically balanced Spanish database, called Albayzin [8], was used. The decoding engine implements Viterbi algorithm with beam-search in an attempt to improve the searching time.

3.2. Text-input translation results

The SFST was learned from the training set described in Table 1. Then, the sentences from the test set were translated by that SFST. The translation given by the system was compared with the reference sentence. Two well-known evaluation measures were taken into account, WER (*Word Error Rate*) and PER (*Position-independent Error Rate*).

In Table 2 we present some preliminary translation experiments from Spanish into Basque and into English. Moreover, we compare our Spanish-Basque translation results with those provided by *Opentrad*, the first publicly available translator including this pair of languages.

Opentrad has been enriched with a bilingual vocabulary and the output has been post-processed in order to get closer results to the reference-style. Besides, all output words with an alarm symbol have been revised as a professional translator would do.

	Text-input			
	Spanish-Basque		Spanish-English	
	WER	PER	WER	PER
Transfer	84.1	75.1	–	–
Statistical	46.5	37.6	28.1	20.2

Table 2. Error rates for text-input machine translation for the 500 test sentences in Table 1 using different techniques: transfer and statistics.

As it has been mentioned at the beginning of this work, these are preliminary results, just to make measures about the recently harvested trilingual corpus. However, they have already shown the difficulty of the translation into Basque since there are much more errors under the same conditions (Table 2).

If we compare the transfer system versus the statistical one, the later seems to performs much better, anyway the difference is not that high if we compare the obtained translations. The point is that the statistical system can also learn the translation style, which is a good choice for simple tasks with restricted domain and construction.

Since only a single reference is available, when the translation given by the system does not match its expected translation or reference, it is punished even if it could be acceptable from a human point of view. Despite WER and PER are a bit pessimistic evaluation measures, at leas, they are automatic and therefore, objectives. Nevertheless, 50 randomly obtained translations were evaluated by 3 experts, within a ranking of 1-5 (1 the worst score). *Opentrad* was given 2 points by the three evaluators, and statistical FST was given 3 points by two evaluators and 4 points by the other one. Both, objective and subjective meassures show that a great effort has to be done in order to improve Spanish-Basque translation results for a real practical MT application.

Here we show Spanish into Basque translation examples obtained by the Transfer (T-sys) and Statistical (S-sys) systems for the first two test-sentences. In the first example, the statistical system (S-sys) performs well except for the last word (*da*), which should be *dira*, since *egonkor mantendu* could be taken as a synonym of *ez dira*

aldatuko. However, the output of the transfer system (T-sys) is not well formed. In the second example, both systems offer an appropriate output even if they do not match the reference.

1. Las temperaturas máximas sin cambios o ligeramente más altas.

   ref Tenperatura maximoak ez dira aldatuko edo gutxi igoko dira.

   T-sys Aldaketarik gabe edo arinki areago garaiak tenperatura handienak.

   S-sys Tenperatura maximoak egonkor mantendu edo gutxi igoko da.

2. Brumas y bancos de niebla matinales.

   ref Goizean lanbroa eta lainoguneak izango dira.

   T-sys Lanbroak eta goizeko lainoguneak.

   S-sys Goizean lanbroa eta lainoguneak azalduko dira.

3.3. Speech-input machine translation results

Speech-input machine translation could be a cheap choice in order to translate TV weather forecasts. On Table 3 we show some translation results making use of the two architectures previously mentioned. Those architectures behave different for different language pairs. The integrated one works better for Spanish-English than for Spanish-Basque. Moreover, in terms of WER, there is not hardly any difference between text-input and speech-input machine translation for Spanish-English.

	Speech-input			
	Spanish-Basque		**Spanish-English**	
	WER	PER	WER	PER
Integrated	57.6	55.6	29.9	27.6
Serial	51.7	42.4	31.0	29.2

Table 3. Error rates for speech-input machine translation with both Integrated and Serial architectures.

4. CONCLUDING REMARKS AND FURTHER WORK

A great effort have been made to harvest a trilingual text and speech corpus in Basque, Spanish and English for example based machine translation. It has been successfully used for SFST training. Both text and speech-input preliminary translation results have been reported. At this point, we can compare the Basque language related translation results with other better exploited languages, such as Spanish or English.

Since multilingual corpora are too scarce, multi-target translation is still a vaguely studied field. With this trilingual corpus we aim to make an attempt in that area in future works.

Experimental results show that SFST models perform much better on similar languages and they work worse with morphologically rich languages, in special dealing with long-distance reorderings, as in the case of Spanish-Basque language pair (as shown in Figure 1). This being so, we are planing as further work, to make use of reordering models that have been proved to report good results in similar circumstances [9].

5. BIBLIOGRAPHY

[1] Antonio M. Corbí-Bellot, Mikel L. Forcada, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, Iñaki Alegria, Aingeru Mayor, y Kepa Sarasola, "An open-source shallow-transfer machine translation engine for the romance languages of spain," in *Proceedings of the Tenth Conference of the European Association for Machine Translation*, Budapest, Hungary, May 2005, pp. 79–86.

[2] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, y R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[3] Alicia Pérez, Inés Torres, Francisco Casacuberta, y Víctor Guijarrubia, "A Spanish-Basque weather forecast corpus for probabilistic speech translation," in *Proceedings of the 5th SALTMIL Workshop on Minority Languages*, Genoa, Italy, 2006.

[4] Enrique Vidal, "Finite-state speech-to-speech translation," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Munich, Germany, Apr. 1997, vol. 1, pp. 111–114.

[5] F. Casacuberta y E. Vidal, "Machine translation with inferred stochastic finite-state transducers," *Computational Linguistics*, vol. 30, no. 2, pp. 205–225, 2004.

[6] I. Torres y A. Varona, "k-tss language models in speech recognition systems," *Computer Speech and Language*, vol. 15, no. 2, pp. 127–149, 2001.

[7] I. García-Varea, D. Ortiz, F. Nevado, P.A. Gómez, y F. Casacuberta, "Automatic segmentation of bilingual corpora: A comparison of different techniques," in *Proceedings of the Second Iberian Conference on Pattern Recognition and Image Analysis*, vol. 3523 of *Lecture Notes in Computer Science*, pp. 614–621. Springer-Verlag, Estoril (Portugal), June 2005.

[8] A. Moreno, D. Poch, A. Bonafonte, E. Lleida, J. Llisterri, J. B. Mariño, y C. Nadeu, "Albayzin speech database: Design of the phonetic corpus," in *Proc.*

of the European Conference on Speech Communications and Technology (EUROSPEECH), Berlín, Germany, 1993.

[9] Stephan Kanthak, David Vilar, Evgeny Matusov, Richard Zens, y Hermann Ney, "Novel reordering approaches in phrase-based statistical machine translation," in *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, Ann Arbor, Michigan, June 2005, pp. 167–174, Association for Computational Linguistics.

CHINESE-SPANISH STATISTICAL MACHINE TRANSLATION EXPERIMENTS

Rafael E. Banchs, Josep M. Crego, Patrik Lambert and José B. Mariño

Department of Signal Theory and Communications, Universitat Politècnica de Catalunya

ABSTRACT

This article presents some experimental results on Chinese-Spanish machine translation. The implemented translation system is based on the statistical framework and, more specifically, it implements the bilingual n-gram approach. Since, as far as we know, no Chinese-Spanish parallel corpus is currently available for training purposes, an alternative method for artificially constructing a training corpus is proposed and evaluated. Finally, the obtained translations are compared, in terms of translation quality, with those obtained by using a rule-based translation engine which is available on-line.

1. INTRODUCTION

In spite of the global effort currently invested in machine translation technologies, it is interesting to note that most of it is currently concentrated in translation pairs such as Spanish-English, Chinese-English, Arabic-English and Japanese-English; while other language pairs such as Chinese-Spanish are, as far as we know, somehow unattended from both research and commercial perspectives. In response to this, the main goal of this work is to build and evaluate a direct Chinese to Spanish Statistical Machine Translation (SMT) system.

Nevertheless, the main drawback for building such a SMT system is the inexistence, at least as a publicly available resource, of a bilingual Chinese-Spanish parallel corpus large enough to perform an appropriate training of the bilingual translation model. In this way, an alternative experimental method for building the required training corpus is proposed and studied.

This procedure relies on using an English to Spanish SMT system for translating the English side of a Chinese-English parallel corpus into Spanish. The resulting Chinese-Spanish parallel corpus is depurated by an additional filtering stage in order to eliminate possible noisy data resulting from the translation errors implied in its generation.

Finally, the resulting translation system is evaluated by comparing the obtained translations, in terms of translation quality, with those obtained by using a rule-based translation system which is publicly available on-line.

This document is structured as follows. First, section 2 presents a very brief overview of the SMT system used within this work. Next, section 3 describes in detail the method proposed for constructing the Chinese-Spanish parallel training corpus. Then, section 4 describes and discusses the experiments conducted and their corresponding results. Finally, section 5 presents the conclusions, as well as the further work to be performed in the near future for improving the presented Chinese to Spanish SMT system.

2. THE BILINGUAL N-GRAM SMT APPROACH

This section presents a very brief description of the SMT approach that was used for all experiments performed within this work. For a more detailed description see [1], [2] and [3]. This approach implements a translation model that has been derived from the finite-state perspective; more specifically, from [4]. However, different from it, where the translation model is implemented by using a finite-state transducer, the SMT approach used here considers a translation model which is based on 3-grams of bilingual units which are referred to as tuples.

Tuples are extracted from Viterbi alignments (more specifically, from the union of source-to-target and target-to-source alignments) in such a way that a maximal monotonic segmentation of each bilingual sentence pair is achieved. Figure 1 illustrates the corresponding tuple segmentation for a given bilingual sentence pair.

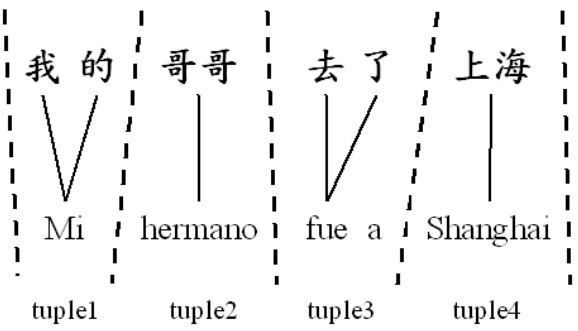


Figure 1. *Example of tuple segmentation for a given bilingual sentence pair.*

This work has been partly funded by TALP (Centre de Tecnologies i Aplicacions del Llenguatge i la Parla) and by the Spanish Department of Education and Science.

In addition to the tuple 3-gram translation model, the considered SMT system implements four additional feature functions which are log-linearly combined with the translation model for decoding purposes. These features are: a target language model implemented by means of word 4-grams, a word bonus model that compensates the system preference for short translations over large ones, and two complementary translation models (source-to-target and target-to-source) which are implemented by using the IBM-1 lexical parameters.

Finally, a specific n-gram based SMT search engine was used for decoding. A detailed description can be found in [2]. This decoder implements a beam-search strategy based on dynamic programming and allows for non-monotonic search. However, all experiments performed in this work were performed by using monotonic search. Although word reordering plays a very important role in the translation task under consideration, however, as a first approximation, and in order to maintain computational time manageable, we have opted for monotonic decoding. Additionally, an optimization tool based on a downhill simplex method was also developed. This algorithm allows for estimating log-linear weights for each feature so that the translation *BLEU* is maximized over a development set.

3. PARALLEL CORPUS CONSTRUCTION

As already mentioned, the main drawback for building the proposed SMT system is the inexistence, at least as a publicly available resource, of a bilingual parallel corpus for performing the translation model training. Although much information in Chinese and Spanish is available through the LDC (Linguistic Data Consortium *http://www.ldc.upenn.edu/*), the intersection of databases containing either Spanish or Chinese is null, so the extraction of a Chinese-Spanish parallel corpus from the existing databases is not possible at all. For this reason, an alternative method for building the required training corpus is proposed and evaluated. The method is depicted in detail within this section.

The proposed method relies on using an English to Spanish SMT system for constructing a Chinese-Spanish parallel corpus by translating the English side of a Chinese-English parallel corpus into Spanish. The Chinese-English parallel corpus used corresponds to a collection of speeches from United Nations, which is available through the LDC. More specifically, a 100K-sentence subset of the *UN Chinese English Parallel Text* (LDC2004E12) was used. On the other hand, for the English-Spanish parallel corpus, the European Parliament data available through the TC-STAR (Technology and Corpora for Speech-to-Speech Translation *http://www. tc-star.org/*) was considered. More specifically, a 100K-sentence subset of the training data from the TC-STAR second evaluation campaign was used. Table 1 presents basic statistics for the Chinese-English and Spanish-English parallel data sets.

Corpus	Language	Sentences	Words	Vocab.
ZH-EN	Chinese	105 K	1.9 M	29.5 K
	English	105 K	2.1 M	34.8 K
ES-EN	Spanish	105 K	2.0 M	40.0 K
	English	105 K	2.0 M	27.0 K

Table 1. *Number of sentences, running words and vocabulary for Chinese-English and Spanish-English corpora (K stands for thousands and M for millions).*

In this way, an English to Spanish translation system, based on the approach described in section 2, was trained by using the corresponding 100K-sentence training data presented in table 1 and optimized by maximizing the translation *BLEU* over a development set extracted from the original Spanish-English parallel data. This system was used to translate into Spanish the English side of the Chinese-English data set in table 1.

Finally, Chinese-Spanish development and test sets were manually constructed. The development corpus was created by manually translating into Spanish the English side of a 330-sentence set extracted from the original Chinese-English parallel data. This set was selected such that no overlap occurred with the 100K-sentence training set. Similarly, a 100-sentence test set was built by means of a similar procedure. Table 2 presents the basic statistics for the resulting Chinese-Spanish training set, as well as for the manually constructed development and test sets.

Corpus	Language	Sentences	Words	Vocab.
TRAIN	Chinese	105 K	1.9 M	29.5 K
	Spanish	105 K	2.0 M	34.8 K
DEV	Chinese	330	6.0 K	1.6 K
	Spanish	330	6.8 K	2.0 K
TEST	Chinese	100	1.9 K	813
	Spanish	100	2.1 K	908

Table 2. *Number of sentences, running words and vocabulary for the constructed Chinese-Spanish data sets (K stands for thousands and M for millions).*

4. EXPERIMENTS AND RESULTS

This section presents the two performed experiments and their corresponding results. The first one evaluates the possibility of improving translation accuracy for the direct Chinese-Spanish translation system by filtering the artificially constructed training corpus. The second one compares the performance of the developed SMT system with a rule-based MT system.

4.1. Corpus Filtering

Previous experimentation has confirmed that the artificial construction of a Chinese-Spanish bilingual corpus does not necessarily conduce to significant improvements in translation accuracy with respect to the

simpler approach of performing Chinese to Spanish translations by using English as a bridge [5]. According to this result, it seems that the only way to actually exploit a procedure of this kind is by being able to retain the most useful sentence pairs of the artificially generated corpus. In this way, the proposed methodology is complemented with a corpus preprocessing stage in which the constructed Chinese-Spanish corpus is filtered in order to eliminate possible noisy data resulting from the translation errors implied in its generation.

To this end, a filtering strategy based on language model statistics is implemented. This filtering strategy consists on using a Spanish language model for selecting those best Spanish sentences in the Chinese-Spanish parallel corpus. Notice that this filtering is conducted only in the Spanish side of the corpus because it corresponds to the one which was artificially generated by translating the English side of the original Chinese-English parallel corpus. So, the noise expected to occur in the Chinese-Spanish corpus should be related to the translation errors produced by the English to Spanish translation system.

In order to implement the filter, a 3-gram language model, trained from the Spanish side of the 100K-sentence English-Spanish corpus in table 1 was used. Since language model probabilities are affected by sentence length, this filtering was performed independently for each subset of Spanish sentences of equal length.

According to this, the 100K-sentence Chinese-Spanish data set presented in table 2 was filtered by using the proposed language model criterion. Different threshold values were considered in order to generate several training subsets of some predefined different sizes. More specifically, training subsets of 95, 90, 85, 80, 70, 60, 50, 40, 30, 20 and 10K-sentences were generated.

Notice that, although training data size reduction has a negative impact on translation accuracy, it is expected that the noisy data reduction provided by the filtering process prevails over the data reduction effect so the overall system performance is incremented. In order to evaluate the effect of filtering independently from the effect of reducing the size of the training data set, a control experiment was performed for each of the eleven filtering experiments under consideration. Such control experiments consisted in training a translation system by using a randomly generated equal-sized training subset. In this way, each filtering experiment could be compared with a control one that was trained with the same amount of data, but selected at random.

In summary, a total amount of twenty two training subsets were generated. Eleven of them by using the proposed filtering strategy, and the other eleven at random. Accordingly, twenty two Chinese to Spanish translation systems were independently trained and optimized. The corresponding results, in terms of translation *BLEU* measured over the development data set, are presented in figure 2.

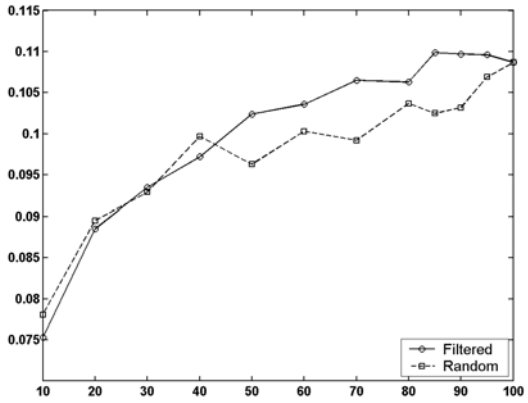


Figure 2. *Translation BLEU, over the development set, for the eleven filtered-data and control translation systems. The horizontal axis represents the size of the training data set in thousands of sentences.*

As seen from figure 2, down to the 50K-sentence corpus, filtering does help improving translation accuracy with respect to those systems trained with the same amount of non-filtered data. Note also, how this improvement tends to fade out for smaller-sized corpora. Additionally, note that no actual translation quality improvement has been achieved by any of the filtered systems with respect to the original 100K-sentence system. This suggests that the negative effect resulting from data training reduction is at least as relevant as the positive effect resulting from filtering.

In order to alleviate the incidence of data reduction on overall system performance, the use of an independently trained target language model was evaluated. In this way, all experiments presented in figure 2 were repeated by using a Spanish language model (trained from data in table 1) as the decoder's target language model feature, instead of training an individual model for each system from its corresponding training data. These results are presented in figure 3.

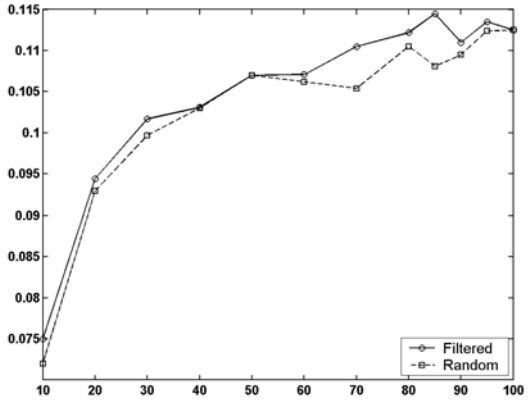


Figure 3. *Translation BLEU, over the development set, for the eleven filtered-data and control translation systems. A common target language model feature was used for all systems.*

As seen from figure 3, in this second set of experiments, filtered-data and control systems achieved better *BLEU* scores than their corresponding systems in figure 2. This can be explained in terms of the better target language model that was used for this second set of experiments. Additionally, it can be noticed that similar to figure 2, filtered-data systems performed slightly better than control systems for larger-sized data sets. However, in this case, the observed gap is smaller than the one observed in figure 2.

4.2. Comparison with a Rule Based MT System

As an additional evaluation of the developed translation system, a comparison with a rule-based translation system was performed. In order to do this, a translation system which is publicly available on-line was used (*http://www.worldlingo.com/en/products_services/worldlingo_translator.html*). In order to avoid any possible bias resulting from the optimization process, the test set described in table 2 was used for this evaluation.

According to this, the test data set was translated from Chinese into Spanish by using the best performing system from figure 3 and the on-line translation system. Both outputs were automatically evaluated in terms of translation accuracy (*BLEU* and *NIST*) and error rates (*PER* and *WER*). Results are presented in table 3.

System	*BLEU*	*NIST*	*PER*	*WER*
Developed	0.1336	4.3101	57.99%	79.73%
On-line	0.0697	2.8355	74.15%	93.71%

Table 3. *Translation accuracy scores and error rates computed over the test set translations for both the developed statistical translation system and the on-line translation system.*

As seen from table 3, the developed translation system outperforms the on-line translation system according to all evaluation metrics. Although it is known that accuracy scores such as *BLEU* and *NIST* tend to favor statistical systems over rule-based systems, notice that both error rates (*PER* and *WER*) are also significantly better for the developed system than for the on-line one.

5. CONCLUSIONS AND FURTHER WORK

This work presented some experimental results on Chinese to Spanish machine translation. A method for artificially constructing and filtering a Spanish-Chinese parallel corpus was presented and evaluated. Results demonstrated that the negative effect resulting from data reduction is at least as relevant as the positive effect resulting from filtering for the overall system performance. However, this situation can be somehow alleviated by using a larger and independently trained target language model.

Additionally, the constructed translation system was compared with a rule-based translation system which is publicly available on-line. The developed system outperformed the on-line one. However, this comparison is not completely fair because, first, the statistical system was adapted to the task under consideration while the rule-based one was not, and second, it is known that accuracy scores such as *BLEU* and *NIST* tend to favor statistical systems over rule-based systems. According to this, more work should be done in order to perform a more fair comparison between the considered systems.

For further research we are planning to work in two main directions. First, we will attempt improving the Chinese-Spanish parallel corpus construction technique. In this sense, different alternatives for filtering the artificially constructed data set should be designed and evaluated, such as the use of dictionaries and morpho-syntactic information, as well as the confidence estimation techniques for Statistical Machine Translation described in [6]. The second main direction of work should be related to improvements in the translation system, by including additional features and allowing for non-monotonic search in the translation task under consideration.

6. REFERENCES

[1] J.B. Mariño, R.E. Banchs, J.M. Crego, A. de Gispert, P. Lambert, J.A.R. Fonollosa, M. Ruiz, 2005, "Bilingual n-gram statistical machine translation", *in Proc. of the X MT-summit*, pp. 275-282.

[2] J.M. Crego, J.B. Mariño, A. de Gispert, 2005, "A ngram-based statistical machine translation decoder", *in Proc. of the 9th European Conference on Speech Communication and Tech-nology*, Interspeech.

[3] J. M. Crego, A. de Gispert, and J. Mariño, "The TALP ngram-based SMT system for IWSLT'05" Pittsburgh, USA, October 2005, pp. 191–198.

[4] F. Casacuberta, E. Vidal, 2004, "Machine translation with inferred stochastic finite-state transducers", *Computational Linguistics*, vol 30, pp. 205-225.

[5] R.E. Banchs, J.M. Crego, P. Lambert, J.B. Mariño, 2006, "A feasibility study for chinese-spanish statistical machine translation", in *Proc. of the 5th Int. Symposium on Chinese Spoken Language Processing.*

[6] N. Ueffing, H. Ney, 2004, "Bayes Decision Rules and Confidence Measures for Statistical Machine Translation", in *EsTAL - España for Natural Language Processing*, Springer Verlag, LNCS, Alicante, Spain, pp. 70-81.

DIFERENTES APROXIMACIONES A LA CODIFICACIÓN DEL VOCABULARIO EN MODELADO DE LENGUAJE CONEXIONISTA

Salvador Tortajada Velert, María José Castro Bleda

stortajada@dsic.upv.es, mcastro@dsic.upv.es

Departamento de Sistemas Informáticos y Computación,
Universidad Politécnica de Valencia, Valencia, España

RESUMEN

En los últimos años ha crecido el interés en la aplicación de las redes neuronales artificiales aplicadas a problemas de procesamiento del lenguaje natural. Este trabajo emplea modelos conexionistas para modelado del lenguaje haciendo hincapié en diversas técnicas de codificación de los datos con el fin de superar las limitaciones impuestas por las grandes dimensiones de los vocabularios. Los resultados obtenidos muestran algunas de las ventajas y desventajas de estas distintas aproximaciones.

1. INTRODUCCIÓN

En este trabajo se propone el uso de redes neuronales artificiales (RNA) para estimar modelos del lenguaje. Un modelo del lenguaje es un conjunto de mecanismos que definen la estructura del lenguaje restringiendo adecuadamente las secuencias de unidades lingüísticas probables que definen el conjunto de frases permitidas. Generalmente, el modelado del lenguaje no es un fin en sí mismo, sino que se aplica en otros sistemas como, por ejemplo, reconocimiento automático del habla o traducción automática. Las aproximaciones clásicas al modelado del lenguaje están basadas mayoritariamente en N-gramas [1] y, en menor medida, en gramáticas estocásticas [2]. A pesar de ello, el uso de RNA para el modelado del lenguaje ha sido empleado con éxito anteriormente [3–8]. Sin embargo, las características propias del lenguaje y las grandes dimensiones del vocabulario con el que se suele trabajar impiden obtener resultados más satisfactorios. Para solventar este problema se propone el uso de un tipo de codificación distinto, más compacto y que reduce las dimensiones de los parámetros a estimar sin necesidad de reducir la talla del vocabulario.

En este trabajo se han empleado redes conectadas hacia adelante, Perceptrón Multicapa (PM), entrenadas con el algoritmo básico de retropropagación del error para estimar modelos de lenguaje. Los resultados obtenidos muestran que, efectivamente, el uso de este tipo de redes ofrece ciertas ventajas y, además, invitan a reflexionar acerca de la importancia de este paradigma dentro de la disciplina del tratamiento del lenguaje natural.

Este trabajo se ha desarrollado en el marco del proyecto EDECAN (TIN2005-08660-c04-02) subvencionado por el MEC.

2. CODIFICACIÓN LOCAL Y DISTRIBUIDA

Uno de los problemas más graves a la hora de enfrentarse con un problema de lingüística computacional es la dimensionalidad del vocabulario. La cantidad de términos que se pueden encontrar en los distintos ámbitos del lenguaje y las distintas combinaciones posibles son enormes. Esta dificultad afecta al tratamiento del lenguaje natural en general y al modelado del lenguaje en particular. La solución empleada en este trabajo para los modelos conexionistas está directamente relacionada con la manera en la que las unidades lingüísticas son representadas en el modelo.

2.1. Codificación local

Habitualmente la representación del vocabulario en los modelos conexionistas se realiza siguiendo una *codificación local* o "1-de-C", donde cada patrón tiene una dimensión igual a la talla del vocabulario Ω, siendo $|\Omega| = C$. Dicha codificación emplea una única unidad de la capa de entrada para representar una palabra. Esto es, aquella unidad i que representa a la palabra i-ésima del vocabulario se activa y el resto quedan desactivadas. La codificación local es una forma sencilla, explícita y fácil de interpretar y manipular. Sin embargo, para vocabularios de grandes dimensiones, una codificación "1-de-C" provoca una gran dispersión de los datos, así como una gran ineficiencia puesto que se genera un número de conexiones y parámetros excesivo.

A continuación podemos observar un ejemplo de esta representación con dos palabras de un vocabulario hipotético de talla 16:

Azul	0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
Rojas	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0

2.2. Codificación distribuida

La solución empleada en otros trabajos -y que se sigue en éste-, consiste en utilizar un tipo de codificación

distinta [5,9], conocida como *codificación distribuida*. En la codificación distribuida cada unidad de entrada puede participar en la representación de una o más palabras. Se entiende que al intervenir varias unidades en la representación de las palabras, se necesitarán menos unidades de entrada y, por lo tanto, se tendrán menos parámetros a estimar. Dentro de la codificación distribuida existen dos tipos principales que presentamos a continuación.

2.2.1. Distribuida simbólica

Siguiendo con el ejemplo del vocabulario hipotético, podemos ver cómo se codificarían las palabras *azul* y *rojas* con una representación distribuida simbólica:

Azul	1 0 1 0 1 0 0
Rojas	1 0 0 1 1 0 0

En este tipo de representación cada unidad corresponde a una característica de la palabra, sea semántica o sintáctica. Así, por ejemplo, la primera unidad podría indicar que las palabras son un adjetivo; la siguiente unidad señalaría que es un verbo; la tercera podría indicar si el género es masculino o femenino; etcétera. Este tipo de codificación es rápidamente interpretable por un observador humano. Sin embargo, la reducción de las dimensiones depende de la cantidad de características de los términos que se desee representar y, además, este tipo de representaciones son, generalmente, *ad-hoc*. Por último, la automatización de este tipo de codificación distribuida podría plantear un nuevo problema, ya que el conocimiento previo necesario para saber algo tan sencillo como si una palabra del vocabulario es masculina o femenina requiere un esfuerzo computacional añadido.

2.2.2. Distribuida subsimbólica

En este tipo de codificación, las unidades, al contrario que en el caso anterior, no tienen una interpretación directa preestablecida de ningún tipo. Así, las palabras de nuestro ejemplo, *azul* y *rojas*, podrían ser representadas del siguiente modo:

Azul	0.22 0.04 0.89 0.45
Rojas	0.12 0.91 0.09 0.77

Este tipo de representación puede ser automatizada y puede aprenderse a partir de un corpus de entrenamiento [9]. Las dimensiones suelen ser del orden de $\lg |\Omega|$, que es un factor de disminución considerable. En cambio, la codificación no puede ser interpretada por un observador humano.

2.3. Codificación del vocabulario

En este trabajo se ha probado una codificación distribuida subsimbólica en varios de los experimentos. El proceso de codificación seguido ha variado en función del uso, o no, del contexto que rodeaba a las palabras. Así,

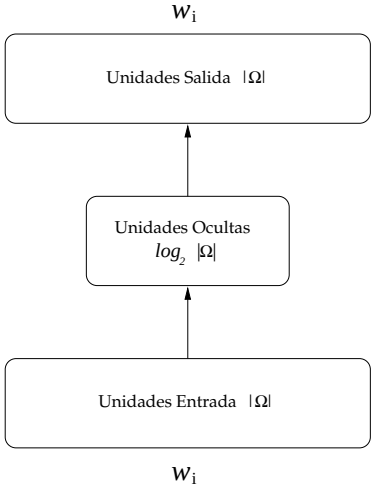


Figura 1. Perceptrón Multicapa para obtener una codifi cación distribuida subsimbólica a partir de la representación local de los datos. La entrada y salida son codifi caciones locales de la palabra. Mediante el entrenamiento de la red podemos conseguir una codifi cación distribuida en la capa oculta.

el codificador empleado -también basado en RNA- puede utilizar contexto o cifrar únicamente la palabra aislada. Este artículo se centra en la segunda opción, dejando la primera para futuras investigaciones.

La codificación sin emplear contexto se ha realizado a partir de un PM cuyas capas de entrada y salida consistían en una representación del vocabulario, Ω, codificado de forma local; y cuya capa oculta se compone de un número de unidades inferior a $|\Omega|$. Se puede ver un esquema en la Figura 1. Normalmente, con la codificación distribuida se reduce el número de unidades empleadas en un factor logarítmico. Es decir, $H = \lg |\Omega|$, donde H es el número de unidades ocultas empleadas en el PM codificador y, por tanto, el número de unidades empleadas para representar cada palabra del vocabulario.

Al entrenar no es necesario establecer un criterio de parada puesto que estamos ante un problema cerrado: los patrones son iguales en la entrada y en la salida y no existe peligro de sobreentrenamiento. Por lo tanto, la complejidad se limita a entrenar la red hasta obtener un error cuadrático medio que converja a cero.

3. MODELOS DEL LENGUAJE CONEXIONISTAS

Los modelos de lenguaje se estiman habitualmente haciendo uso de una aproximación estadística. Estos modelos se basan en la predicción de cada unidad lingüística de la frase dadas las anteriores. Los modelos basados en N-gramas son los más extendidos en la actualidad y consisten en aproximar la probabilidad como si únicamente influyeran las últimas $N-1$ palabras previas. Esto es,

$$Pr(W) \approx \prod_{i=1}^{|W|} Pr(w_i | w_{i-N+1} \dots w_{i-1}).$$

Partición	*No. de palabras*
Entrenamiento	1 004 073
Validación	80 156
Test	89 537

Tabla 1. Corpus Wall Street Journal.

Estos modelos se pueden aprender de forma automática a partir de un conjunto de entrenamiento. Los valores de cada N-grama pueden ser estimados por máxima verosimilitud.

Los modelos conexionistas introducen una aproximación diferente a la basada en N-gramas. En [3] se presentó este nuevo modelo. Posteriormente, en otros trabajos se han investigado y aplicado los modelos conexionistas para este fin [4–8].

La idea consiste en simular la ecuación de los modelos de lenguaje estadísticos. Para ello, en la capa de entrada se introduce la historia de la palabra a predecir, esto es, $h(w_i) = w_{i-N+1} \dots w_{i-1}$ y en la salida se espera que se prediga la palabra w_i. De este modo, utilizando la función de activación ***softmax*** en las unidades de salida se puede aproximar la probabilidad $Pr(w_i|h(w_i))$.

Los modelos conexionistas aportan dos ventajas respecto a los modelos convencionales. Primero, el número de parámetros a calcular crece linealmente con N en lugar de exponencialmente. Segundo, el suavizado se realiza automáticamente y se puede esperar una mejor generalización de los modelos. A pesar de esto, los modelos conexionistas presentan un inconveniente respecto a la aproximación basada en N-gramas: el coste temporal para entrenar el modelo.

4. LA TAREA PENN TREEBANK

El corpus usado en los experimentos es la parte del Wall Street Journal que había sido procesada en el Penn Treebank Project [10]. Este corpus consiste en un conjunto de textos en inglés extraídos del periódico Wall Street Journal. El número total de palabras es de 1 millón aproximadamente, con un vocabulario de más de $49\,000$. Todo el corpus fue etiquetado automáticamente con dos tipos de etiquetas: etiquetas POS y etiquetas sintácticas. El etiquetado POS consiste en un conjunto de 45 etiquetas diferentes, mientras que el sintáctico lo forman 14 categorías.

El corpus se dividió en tres conjuntos: entrenamiento, validación y prueba. Las características de las tres particiones se describen en la Tabla 1. Hemos utilizado en la experimentación las etiquetas POS y un vocabulario compuesto por las $10\,000$ palabras más frecuentes que aparecen en la partición de entrenamiento.

5. N-GRAMAS CONEXIONISTAS UTILIZANDO CATEGORÍAS

La primera aproximación al modelado de lenguaje con N-gramas ha sido utilizando las categorías del Penn Treebank, ya que nos ha permitido trabajar con un vocabulario muy reducido (45 categorías originales más el símbolo de contexto y el de token desconocido) sobre un corpus muy extenso. Así pues, hemos modelizado bigramas y trigramas con dicho corpus empleando un PM. Con el propósito de comparar los resultados obtenidos con los modelos de N-gramas estadísticos clásicos se ha empleado el toolkit SRILM para modelado de lenguaje con suavizado *Good-Turing* [11].

5.1. Experimentación: Codificación local

Para modelizar un bigrama con un PM se empleó una red con topología 47-H-47, donde H es el número de unidades ocultas, tanto en una como en dos capas. Las primeras 47 unidades representan la palabra de entrada, mientras que las 47 unidades de salida corresponden a las probabilidades que asocia el PM a cada una de las 47 posibles sucesoras. Para determinar el valor de H se ha realizado un barrido exhaustivo, probando diferentes combinaciones de una y dos capas, con diferente número de unidades en cada una. El mejor resultado se ha obtenido con una capa oculta de 16 unidades.

Para el caso del trigrama el proceso seguido fue prácticamente el mismo. Tras establecer una topología para el PM, que en este caso sería de 94-H-47, se pasó directamente a la búsqueda del mejor valor de H. En este caso, el mejor valor obtenido para H ha sido de 128 unidades con una única capa oculta.

5.1.1. Análisis de los resultados

Tras modelizar los N-gramas con $N = 2, 3$ podemos extraer una serie de conclusiones muy interesantes (todos los resultados se muestran en la Tabla 2). En ambos casos se han obtenido mejores resultados con el modelo conexionista que el correspondiente N-grama estadístico, siendo el número de parámetros mucho menor. El suavizado implícito de la red neuronal es capaz de afrontar muy bien la falta de muestras, mucho mejor que los suavizados de los N-gramas estadísticos. Por su parte, el entrenamiento de un N-grama conexionista es mucho más costoso en tiempo y además requiere un proceso de búsqueda de la mejor topología y de los mejores parámetros.

5.2. Experimentación: Codificación distribuida

La realización de un modelo conexionista basado en N-gramas empleando una codificación distribuida es una primera aproximación para llegar a realizar un modelo de lenguaje conexionista enfocado a superar las limitaciones que presentan los vocabularios de grandes dimensiones.

En esta aproximación tratamos de comprobar que una codificación distribuida de los datos -categorías- no sólo reduce el número de unidades en la capa de entrada, y por lo tanto el número de parámetros a estimar, sino que también proporciona un comportamiento similar al modelo con codificación local.

ML (POS)	*#*	*C*	*PPE*
Bigrama estadístico	2 209	-	9.52
Bigrama-PM, Local	1 567	257	9.47
Bigrama-PM, Distr.	807	146	9.61
Trigrama estadístico	103 823	-	8.36
Trigrama-PM, Local	12 207	115	8.27
Trigrama-PM, Distr.	7 692	30	8.49

Tabla 2. Perplejidad del conjunto de test (PPE) para los modelos de lenguaje del corpus de etiquetas POS del Penn Treebank con codifi cación local y distribuida, número de parámetros (#) y de ciclos (C) que ha costado entrenar la red neuronal, comparados con los modelos estadísticos.

En primer lugar se ha obtenido la codificación distribuida subsimbólica de las categorías POS del Penn Treebank con un PM tal como se explica en la Figura 1. El mejor resultado se obtuvo con una capa oculta de $\lceil \lg |\mathcal{C}| \rceil = 6$ unidades. Una vez obtenida la codificación distribuida, se entrena un PM con la mejor topología encontrada en los experimentos con codificación local, tanto para bigramas como para trigramas (una capa oculta de 16 unidades).

Dado que contamos con una codificación distribuida en la entrada y una codificación local en la salida, la topología final del PM es 6-16-47. Al emplear codificación distribuida en la entrada en lugar de local, el número de unidades se reduce de 47 a 6. Dicha reducción implica, a su vez, una disminución del número de parámetros de $1\,567$ a 807, notablemente inferior a los $2\,209$ parámetros necesarios para estimar los bigramas estadísticos.

Para el modelo con trigramas, la red entrenada y probada tiene una topología 12-128-47. La razón es que la capa de entrada cuenta ahora con las w_{i-2} y w_{i-1} palabras anteriores. Puesto que las palabras están representadas de forma distribuida tenemos $2\lceil \lg |\mathcal{C}| \rceil$ unidades en la capa de entrada; el número de unidades ocultas viene impuesto por el mejor resultado obtenido en las pruebas anteriores; la palabra w_i en la capa de salida se representa mediante codificación local. El número de parámetros total es $7\,692$, inferior los $12\,207$ parámetros a estimar con la codificación local o a los $103\,823$ parámetros del trigrama estadístico.

5.2.1. Análisis de resultados

Se puede observar en la Tabla 2 que la reducción de parámetros a estimar no es gratuita. Aunque la reducción del modelo y su estimación sean computacionalmente más eficientes, los resultados presentan una perplejidad ligeramente peor.

Debemos tener en cuenta que los resultados obtenidos para los modelos con codificación distribuida no han sido obtenidos a partir de un barrido exhaustivo, como se hizo para los modelos conexionistas con codificación local, sino que se han empleado aquellas topologías que mejores resultados ofrecieron en las pruebas mencionadas. Por lo tanto estos resultados son susceptibles de ser mejorados, aunque nada lo garantiza.

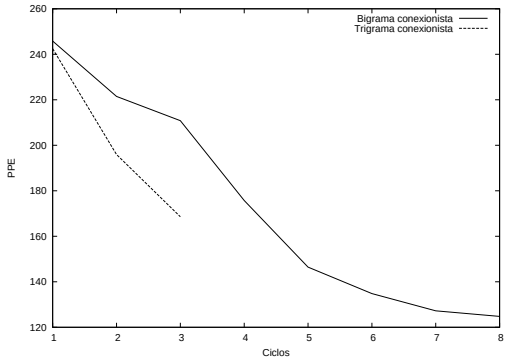


Figura 2. Evolución de la perplejidad de los modelos de lenguaje conexionistas (bigramas y trigramas) con codifi cación local para el corpus de palabras del Penn Treebnak. Los trigramas conexionistas todavía no han llegado a la zona de convergencia, por lo que, con un mayor número de ciclos, se podría obtener un modelo de mayor precisión.

ML (palabras)	*#*	*C*	*PPE*
Bigrama estadístico	10^8	-	137.06
Bigrama-PM, Local	$10^{7.16}$	8	124.79
Trigrama estadístico	10^{12}	-	119.81
Trigrama-PM, Local	$10^{7.39}$	3	168.60

Tabla 3. Perplejidad del conjunto de validación (PPE) para los modelos de lenguaje del corpus Penn Treebank con codifi cación local, número de parámetros (#) y de ciclos (C) que ha costado entrenar la red neuronal, comparados con los modelos estadísticos.

6. N-GRAMAS CONEXIONISTAS UTILIZANDO PALABRAS

Tras utilizar como vocabulario de la tarea las etiquetas POS y ver que los resultados obtenidos eran esperanzadores, se decidió pasar al conjunto de $10\,000$ palabras como vocabulario de la tarea.

6.1. Experimentación: Codificación local

Esta aproximación ha supuesto trabajar con una topología de $10\,000$ unidades en la capa de entrada para el caso del bigrama ($20\,000$ para el trigrama), y de $10\,000$ en la capa de salida.

En el caso de los bigramas, a falta de un barrido exhaustivo para seleccionar la mejor topología, se pueden ver los resultados obtenidos para el corpus de validación tras dos semanas de cálculo ininterrumpido en la Tabla 3. Como se puede observar, la PPE obtenida es muy buena, en comparación con la PPE del bigrama estadístico.

En el caso de trigramas, sólo hemos podido completar 3 ciclos de entrenamiento, que han tardado 30 días en terminar. Sin embargo, como muestra la Figura 2, la curva de perplejidad todavía puede disminuir con más ciclos de entrenamiento de modo que se podría reducir considerablemente.

6.2. Experimentación: Otras codificaciones

Una alternativa a la codificación local es la codificación distribuida, explicada en la sección 2. Sin embargo, la codificación del vocabulario no ha sido satisfactoria al no poder encontrar un mínimo local aceptable del error cuadrático medio. Otra alternativa estudiada es la codificación binaria, que consiste en obtener un diccionario con las palabras que forman el vocabulario y su correspondiente traducción en dígitos binarios. Esto consigue comprimir el tamaño de los vectores a longitud $\lg |\Omega|$. Sin embargo, los resultados obtenidos ofrecían un rendimiento del modelo bastante inferior a la codificación local.

7. CONCLUSIONES Y TRABAJO FUTURO

En el presente documento se han presentado los modelos de lenguaje conexionistas como modelos de lenguaje basados en un PM y planteados como alternativa a los modelos de N-gramas estadísticos tradicionales, prestando especial atención al problema de la dimensionalidad del vocabulario a través de la codificación distribuida y binaria.

En los modelos basados en clases (con vocabularios pequeños) se ha demostrado la capacidad de las redes neuronales para mejorar la eficiencia de los modelos estadísticos, así como el menor número de parámetros empleados. Del mismo modo, se demuestra que podemos disminuir la dimensionalidad para representar el vocabulario mediante la codificación distribuida, aún a costa de perder algo de eficiencia a cambio de emplear menos parámetros.

El hándicap de la dimensionalidad se presenta como el problema más grave en los modelos de lenguaje conexionistas. Como se ha podido comprobar, la codificación distribuida, empleada para reducir el número de unidades necesarias para representar el vocabulario, no funciona correctamente para tallas muy grandes. En este sentido giran los trabajos de Bengio y otros autores [5, 9]. Las posibles soluciones al problema están abiertas y siguen siendo un camino a investigar para la mejora de las prestaciones de los modelos conexionistas en general, y su aplicación al modelado del lenguaje, en particular. En esta línea se han presentado resultados de modelos de lenguaje que combinan RNA con otras técnicas [8, 12].

8. BIBLIOGRAFÍA

[1] F. Jelinek, *Statistical methods for speech recognition*, MIT Press, Cambridge, MA, USA, 1997.

[2] C. Chelba y F. Jelinek, "Structured language modeling," *Computer Speech and Language*, vol. 14, pp. 283–332, 2000.

[3] M. Nakamura y K. Shikano, "A study of English word category prediction based on neural networks," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'89)*, Glasgow (Scotland), May 1989, pp. 731–734.

[4] W. Xu y A. Rudnicky, "Can Artificial Neural Networks learn Language Models?," in *Proceedings of the 6th International Conference in Spoken Language Processing (ICSLP'00)*, Beijing (China), 2000.

[5] Y. Bengio, R. Ducharme, P. Vincent, y C. Jauvin, "A Neural Probabilistic Language Model," *Journal of Machine Learning Research*, 2003.

[6] M.J. Castro y F. Prat, "New Directions in Connectionist Language Modeling," *LNCS*, vol. 2686, pp. 598–605, 2003.

[7] A Emami y F. Jelinek, "Exact Training of a Neural Syntactic Language Model," in *ICASSP*, 2004, vol. 1, pp. 245–248.

[8] H. Schwenk y J.-L. Gauvain, "Training neural network language models on very large corpora," in *Joint Conference HLT/EMNLP*, 2005, pp. 201–208.

[9] G.A. Casañ y M.A. Castaño, "Improvements on Automatic Word Codification for Connectionist Machine Translation," in *ECAI*, 2004, pp. 576–580.

[10] M.P. Marcus, B. Santorini, y M.A. Marcinkiewicz, "Building a Large Annotated Corpus of English: The Penn Treebank," *Computational Linguistics*, , no. 19, pp. 313–330, 1994.

[11] A. Stolcke, "SRILM - An extensible language modeling toolkit," in *International Conference on Spoken Language Processing*, 2002, vol. 2, pp. 901–904.

[12] F. Blat, M.J. Castro, S. Tortajada, y J.A. Sánchez, "A hybrid approach to statistical language modeling with multilayer perceptrons and unigrams," in *Proceedings of the 8th International Conference on Text, Speech and Dialogue*, 2005.

ESTIMACIÓN DE MODELOS DE TRADUCCIÓN DE SECUENCIAS DE PALABRAS A PARTIR DE CORPUS MUY GRANDES MEDIANTE THOT

Daniel Ortiz-Martínez[1] *, Ismael García-Varea*[2] *, Francisco Casacuberta*[1]

[1] Universidad Politécnica de Valencia

[2] Universidad de Castilla-la Mancha

RESUMEN

En el ámbito de la traducción automática estadística, los últimos tiempos se han caracterizado por la popularización de los modelos de traducción basados en secuencias de palabras, así como por la aparición de corpus bilingües más y más grandes como el bien conocido corpus EUROPARL. La coincidencia de estos dos acontecimientos ha planteado un problema importante debido a que los modelos de traducción basados en secuencias de palabras requieren un espacio de almacenamiento considerable cuando se estiman a partir de grandes corpus de entrenamiento. Para resolver este problema así como muchos otros relacionados con la estimación y la aplicación de modelos estadísticos de secuencias de palabras, se ha desarrollado la herramienta de libre uso denominada Thot, cuya funcionalidad básica se describe en este artículo. Adicionalmente, se incluyen experimentos de traducción para el corpus EUROPARL usando modelos de traducción generados con Thot y decodificadores basados en el estado del arte.

1. INTRODUCCIÓN

Desde comienzos de la decada de los 90, el interés en la disciplina de la traducción automática estadística (TAE) no ha hecho sino incrementarse, debido a los buenos resultados que ha obtenido al aplicarse en corpus de dominio restringido.

El proceso de traducción puede formularse desde un punto de vista estadístico de la siguiente forma: Sea $f_1^J = f_1 \dots f_J$ la frase origen que queremos traducir en su equivalente en el lenguaje destino $e_1^I = e_1 \dots e_I$. Se considera que cualquier posible frase de la lengua destino es traducción de la frase origen con una probabilidad a posteriori determinada $Pr(e_1^I|f_1^J)$. Según la regla de Bayes, la frase destino que buscamos $\hat{e}_1^I$ será aquella que maximiza[1] el producto del modelo del lenguaje destino $Pr(e_1^I)$ y el modelo de traducción $Pr(f_1^J|e_1^I)$. La ecuación que modela el proceso es la siguiente:

$$\hat{e}_1^I = \arg\max_{e_1^I} \{Pr(e_1^I) \cdot Pr(f_1^J|e_1^I)\} \qquad (1)$$

En función de cómo se concibe la relación entre las palabras origen y destino, se han propuesto diferentes *modelos de traducción*; que intentan explicar la manera en que se genera la frase destino a partir de la frase origen. Esta relación se describe a través del concepto de *alineamiento*; dando lugar a diferentes *modelos estadísticos de alineamiento*. Los modelos de alineamiento IBM y HMM fueron propuestos en [1] y en [2] respectivamente. Estos modelos pertenecen a la categoría de los *modelos basados en palabras*, ya que asumen que en el proceso de traducción se establecen relaciones entre palabras individuales de las frases origen y destino. En los últimos tiempos, se ha demostrado que estos modelos no capturan adecuadamente la información de contexto a la hora de traducir, por lo que se han propuesto modelos que trabajan con grupos de palabras en lugar de palabras, los cuales han constituído una mejora con respecto a los modelos basados en palabras originales descritos en [1]. Se han propuesto diferentes modelos que trabajan con grupos de palabras, por ejemplo, los *modelos sintácticos* se describen en [3], los *alignment templates* se describen en [4]. Adicionalmente, en [5, 6, 7, 8] se ha descrito una reformulación del enfoque *alignment template* que ha dado lugar a los modelos llamados *phrase based*, a los cuales nos vamos a referir en castellano como *modelos de secuencias de palabras*.

En el transcurso de la investigación en traducción automática estadística, se han ido desarrollando herramientas software con objeto de ayudar a los investigadores a mejorar sus sistemas de traducción. Estas herramientas abarcan desde software de entrenamiento de modelos basados en palabra (como la herramienta Giza++) hasta algunos sistemas de traducción específicos, incluyendo el decodificador para modelos basados en secuencias denominado Pharaoh [9]. En el estado del arte actual, resulta muy necesaria una herramienta de entrenamiento de modelos de secuencias para ayudar en la mejora de los sistemas de traducción. En este artículo se presenta una herramienta de libre uso pensada para tal fin.

Este trabajo ha sido parcialmente subvencionado por el proyecto CICYT TIC2003-08681-C02-02, la *Agencia Valenciana de Ciencia y Tecnología* dentro del contrato GRUPOS03/031, la *Generalitat Valenciana*, y el proyecto HERMES II (Vicerrectorado de Investigación - UCLM-06/07)

[1] Obsérvese que la expresión debe ser maximizada también para I; sin embargo, para simplificar se asume que I es conocida.

2. MODELOS DE SECUENCIAS DE PALABRAS

Como ya hemos comentado, éstos modelos surgen como alternativa a los modelos de palabras (o más comúnmente conocidos como modelos de IBM [1]) para superar las limitaciones que presentan, de modo que en lugar de trabajar con diccionarios estadísticos de palabras ($Pr(f_j|e_i)$), trabajan con diccionarios estadísticos de secuencias ($Pr(\tilde{f}_k|\tilde{e}_k)$).

La traducción, usando modelos de secuencias de palabras, de una frase de entrada f_1^J en la frase destino equivalente e_1^I consiste, desde un punto de vista generativo, en escoger la forma en que dicha frase de entrada es segmentada en K secuencias $f_1^J = \tilde{f}_1^K$, seleccionar las secuencias en el lenguaje destino que traducen las secuencias origen y, por último, reordenar; con lo que terminamos obteniendo $e_1^I = \tilde{e}_1^K$. Podemos asumir que las relaciones entre las palabras origen y destino se sumarizan mediante una variable oculta $\tilde{\mathbf{a}} = \tilde{a}_1^K$, que contiene todas las decisiones que se hacen durante la historia generativa.

$$\begin{aligned} Pr(f_1^J|e_1^I) &= \sum_{\tilde{\mathbf{a}}} Pr(\tilde{\mathbf{a}}, \tilde{f}_1^J|\tilde{e}_1^I) \\ &= \sum_{\tilde{\mathbf{a}}} Pr(\tilde{\mathbf{a}}|\tilde{e}_1^I) Pr(\tilde{f}_1^J|\tilde{\mathbf{a}}, \tilde{e}_1^I) \end{aligned} \quad (2)$$

Se pueden hacer diferentes asunciones a partir de la ecuación anterior, aunque lo normal es que los modelos terminen convirtiéndose en diccionarios estadísticos de segmentos. Por ejemplo, en [5] se propone el siguiente modelo:

$$p_\theta(f_1^J, e_1^I) = \alpha(e_1^I) \sum_{\tilde{\mathbf{a}}} \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_{\tilde{a}_k}) \quad (3)$$

donde $\tilde{a}_k$ denota el índice de la secuencia origen $\tilde{e}$ que se alinea con la k'ésima secuencia destino $\tilde{f}_k$ y se asume que todas la posibles segmentaciones tienen la misma probabilidad.

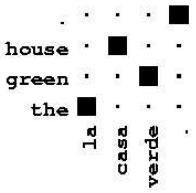
2.1. Extracción de pares de secuencias

Existen principalmente tres maneras de obtener los pares de secuencias, tal y como se describe en [8]:

1. A partir de matrices de alineamiento a nivel de palabra, las cuales establecen una correspondencia entre las palabras de las frases origen y destino.
2. Mediante criterios sintácticos (véase [3])
3. A partir de alineamientos a nivel de frase por medio de un modelo de probabilidad conjunta (véase [6]).

En este artículo nos vamos a centrar en el primer método, en el que nos basamos en matrices de alineamiento a nivel de palabra para extraer los pares de secuencias. Dichas matrices se obtendrán automáticamente como subproducto de la estimación de modelos IBM. Concretamente, dado un par de frases y su correspondiente matriz de alineamiento A, se extraerán aquellos pares de secuencias que sean *consistentes* con la matriz de alineamiento. La condición de consistencia viene dada por la ecuación (4) [4]. La Figura 1 muestra un ejemplo en el que se da un par de frases con su matriz de alineamiento, y el conjunto de todos los pares bilingües consistentes que se pueden extraer.

$$\begin{aligned} \mathcal{BP}(f_1^J, e_1^I, A) = \{(f_j^{j+m}, e_i^{i+n} : \forall (i', j') \in A : \\ j \leq j' \leq j+m \iff i \leq i' \leq i+n\} \end{aligned} \quad (4)$$



secuencia origen	secuencia destino
La	the
casa	house
verde	green
casa verde	green house
La casa verde	the green house
.	.
casa verde .	green house .
La casa verde .	the green house .

Figura 1. Conjunto de pares bilingües consistentes (derecha) para una matriz de alineamiento dada (izquierda).

Un importante inconveniente que poseen los modelos de secuencias es la gran cantidad de espacio en memoria requerida por sus parámetros, por lo que serán necesarias técnicas especiales para su estimación y manejo cuando se trabaja con corpus muy grandes.

3. DESCRIPCIÓN DE LA HERRAMIENTA

La herramienta Thot ha sido desarrollada utilizando el lenguaje de programación C++. Los principios de diseño que han dirigido el proceso de desarrollo han sido: eficiencia, extensibilidad, flexibilidad (trabaja con distintos y bien conocidos formatos de datos) y usabilidad (la herramienta es fácil de usar y su código es fácil de incorporar a nuevo código).

En los apartados siguientes describiremos la funcionalidad básica ofrecida por Thot.

3.1. Operaciones entre matrices de alineamiento

Como se comentó en el apartado 2.1 es habitual la aplicación de operaciones entre matrices de alineamiento con objeto de mejorarlas.Thot incorpora las siguientes operaciones sobre matrices de alineamiento:

Unión : Obtiene la unión de dos matrices.

Intersección : Obtiene la intersección de dos matrices.

Suma : Obtiene la suma de dos o más matrices.

Simetrización : Obtiene una matriz de alineamiento a medio camino entre la unión y la intersección de dos matrices. Esta operación fue propuesta por primera vez en [4] y existen diferentes versiones de la misma.

El formato de fichero para los alineamientos es el generado por la herramienta Giza++. El formato de salida puede ser el formato de Giza++ así como otros dos: en forma de matriz bidimensional (legible por un ser humano), o bien un formato intermedio que se puede convertir fácilmente en otros.

3.2. Modalidades de estimación de modelos de secuencias

Thot permite estimar modelos de secuencias a partir de matrices de alineamiento a nivel de palabra en formato Giza++, tal y como se explicó en la sección 2, estimación a la que nos referiremos de ahora en adelante como estimación *RF* por sus siglas en inglés (*relative frequency*).

La estrategia de estimación RF es heurística por dos razones. En primer lugar los pares bilingües son extraídos a partir de matrices de alineamiento a nivel de palabra, por lo que nos vemos obligados a aplicar un criterio heurístico de consistencia sobre los pares. En segundo lugar, los pares extraídos no son considerados como parte de segmentaciones bilingües completas a la hora de estimar las probabilidades. El primer problema no puede resolverse sin cambiar por completo la estrategia de estimación. Para el segundo, en cambio, pueden ofrecerse alternativas de solución.

Con este propósito, la herramienta implementa una nueva propuesta de estimación que hemos llamado estimación por *pseudo máxima verosimilitud* o estimación pML por sus siglas en inglés (*pseudo maximum-likelihood*)[2] que es diferente la la estrategia de estimación habitual. En concreto, dicha estimación consiste en un proceso de tres pasos que se repiten para cada par de frases del corpus con su correspondiente matriz de alineamiento (f_1^J, e_1^I, A):

1. Obtener el conjunto $\mathcal{BP}(f_1^J, e_1^I, A)$ de todos los pares de secuencias consistentes.
2. Obtener el conjunto $\mathcal{S}_{\mathcal{BP}(f_1^J, e_1^I, A)}$ de todas las segmentaciones bilingües posibles[3] del par (f_1^J, e_1^I) que pueden construirse utilizando pares bilingües consistentes.
3. Actualizar los contadores (en realidad contadores fraccionales) para cada par de secuencias distinto $(\tilde{f}, \tilde{e})$ en el conjunto $\mathcal{S}_{\mathcal{BP}(f_1^J, e_1^I, A)}$:

$$fracCount(\tilde{f}, \tilde{e}) += \frac{N(\tilde{f}, \tilde{e})}{|\mathcal{S}_{\mathcal{BP}(f_1^J, e_1^I, A)}|}$$

donde $N(\tilde{f}, \tilde{e})$ es la cantidad de veces que aparece el par $(\tilde{f}, \tilde{e})$ en $\mathcal{S}_{\mathcal{BP}(f_1^J, e_1^I, A)}$, y $|\cdot|$ denota la talla de un conjunto.

[2]Se ha elegido este nombre porque consideramos que la propuesta sería equivalente a la primera iteración del algoritmo EM, que debería utilizarse en una estimación más rigurosa

[3]Una segmentación bilingüe de talla K de un par de frases (f_1^J, e_1^I) se define como una tripleta $(\tilde{f}_1^K, \tilde{e}_1^K, \tilde{a}_1^K)$, donde $\tilde{a}_1^K$ es una correspondencia uno a uno entre las secuencias en las que se ha dividido ambas frases.

Finalmente, la probabilidad de cada par de secuencias $(\tilde{f}, \tilde{e})$ se calcula de la siguiente forma:

$$p(\tilde{f}|\tilde{e}) = \frac{fracCount(\tilde{f}, \tilde{e})}{\sum_{\tilde{f}} fracCount(\tilde{f}, \tilde{e})}$$

Para clarificar lo anterior puede consultarse la Figura 2 que muestra el conjunto de posibles segmentaciones bilingües de un par de frases para la matriz de alineamiento que se puede ver en la Figura 1.

El paso 2 implica que si un par bilingüe no puede formar parte de una segmentación bilingüe para un par de frases dado, dicho par bilingüe no será extraído. Por esta razón, la estimación pML extrae una menor cantidad de pares que la estimación RF.

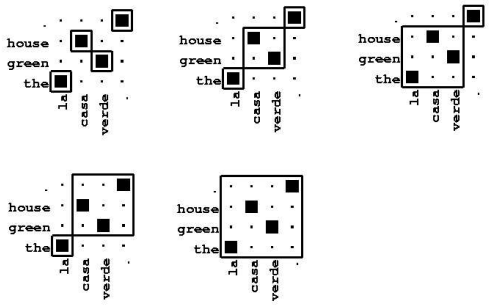


Figura 2. Posibles segmentaciones bilingües para una matriz de alineamiento a nivel de palabra dada.

Adicionalmente, la estimación pML permite obtener modelos de secuencias más completos, incluyendo or ejemplo, un submodelo para la talla de la segmentación K, funcionalidad que ha sido incluída en la herramienta. Por otro lado, y como principal desventaja, esta modalidad de estimación tiene un elevado coste computacional debido a la necesidad de obtener la segmentación bilingüe para cada par de frases.

3.3. Estimación a partir de corpus muy grandes

Uno de los mayores inconvenientes que presentan los modelos de traducción de secuencias reside en sus elevados requerimientos de memoria. Existen diversas alternativas para paliar este importante problema. Una de ellas consiste en limitar el tamaño máximo que pueden tener las secuencias, con lo que se corre el riesgo de obtener peores modelos (no obstante, existen resultados empíricos [8] que demuestran que dicha limitación puede imponerse sin que se produzca una disminución apreciable en la calidad de la traducción).

Otra posibilidad consiste, tanto para la estimación RF como pML, en quedarse con aquellos pares bilingües que satisfagan la restricción de monotonicidad. Todas estas variantes han sido implementadas por la herramienta, cuya salida puede obtenerse en un formato nativo propio, o en el formato de entrada esperado por el traductor de libre uso Pharaoh [9].

No obstante, las técnicas anteriores son insuficientes a la hora de manejar los enormes corpus que se vienen presentando actualmente. Una posible solución a este problema, que permite el entrenamiento de corpus de tamaño arbitrario, viene dada por el Algoritmo 1. Dicho algoritmo se apoya en el trabajo con cuentas en lugar de probabilidades. Consiste en entrenar el corpus dividiéndolo en trozos de tamaño fijo (tam_fragm) volcando cada uno de los submodelos que se obtienen en disco. Una vez volcados los modelos se juntan en un único fichero, se ordenan lexicográficamente y se fusionan las cuentas, dando lugar a un modelo idéntico al que se obtendría de haber entrenado el corpus completo de una sola vez.

Algoritmo 1 algoritmo_entrenamiento (fichAlin)

```
partir(fichAlin,tam_fragm)
para_todo f fragmento de fichAlin hacer
  entrena(f) >> fichCuentas
fin_para
ordenarCuentas(fichCuentas) > fichCuentasOrdenado
fundirCuentas(fichCuentasOrdenado) > modeloSecuencias
```

La única sobrecarga temporal que introduce el algoritmo se debe a la necesidad de ordenar el fichero de cuentas resultante. Creemos que el algoritmo es claramente susceptible de ser paralelizado, por lo que la mencionada sobrecarga temporal puede ser contrarrestada.

3.4. Segmentación de corpus bilingües

Dado un par de frases (f_1^J, e_1^I) y una matriz de alineamiento a nivel de palabra, la herramienta incorpora una funcionalidad que permite obtener la mejor segmentación bilingüe posible en K bisegmentos, e implícitamente el mejor alineamiento a nivel de secuencias $\tilde{a}_1^K$ (o alineamiento de Viterbi) entre ellas, de acuerdo con el siguiente algoritmo:

1. Para todo $K \in \{1 \cdots \text{mín}(J, I)\}$
   - *a*) Extraer todas las segmentaciones bilingües de talla K sujeto a las restricciones $A(f_1^J, e_1^I)$.
   - *b*) Computar y almacenar la probabilidad $p(\tilde{f}_1^K, \tilde{a}_1^K | \tilde{e}_1^K)$ de dichas segmentaciones.
2. Devolver la segmentación bilingüe $(\tilde{f}_1^K, \tilde{e}_1^K, \tilde{a}_1^K)$ de mayor probabilidad.

donde $p(\tilde{f}_1^K, \tilde{a}_1^K | \tilde{e}_1^K) = \prod_{k=1}^K p(\tilde{f}_{\tilde{a}_k} | \tilde{e}_k)$

4. EXPERIMENTOS

En esta sección se presentan resultados utilizando la funcionalidad principal que incorpora la herramienta Thot. En la experimentación se han utilizado tres corpus bien conocidos en el ámbito de la traducción automática estadística, a saber, EUTRANS-I, HANSARDS y EUROPARL cuyas principales características se muestran en la tabla 1.

4.1. Experimentos de generación de alineamientos

Para experimentar con la obtención de las mejores segmentaciones bilingües utilizamos un subconjunto del conjunto de test de EUTRANS-I consistente en 40 pares de frases aleatoriamente seleccionadas. Este corpus fue segmentado por lingüistas [10].

La Tabla 2 muestras las medidas estándar de calidad de segmentación *Recall*, *Precision*, y *F-measure* para tres técnicas diferentes de segmentación bilingüe, incluyendo la que proporciona la herramienta Thot. Las otras dos consisten en la técnica de alineamientos recursivos (RECalign) y los alineamientos de GIATI (GIATIalign) que se describen y utilizan en [10].

Tal y como muestra la Tabla 2, la calidad de segmentación bilingüe alcanzada por Thot mejora la que obtienen las otras dos técnicas.

Técnica	Recall	Precision	F-measure
RECalign	52.96	79.01	**63.41**
GIATIalign	39.99	85.52	**54.50**
Thot	72.58	65.49	**68.85**

Tabla 2. Calidad de segmentación bilingüe para 40 pares de frases extraídas del corpus EUTRANS-I.

4.2. Experimentos de traducción

Se llevaron a cabo experimentos de traducción utilizando la funcionalidad de la herramienta Thot y el traductor Pharaoh; a saber: operaciones entre alineamientos, estimación RF y pML y su aplicación en experimentos de calidad de traducción. Para los experimentos se han usado las medidas de error bien conocidas *Word Error Rate* (WER), *Position independent Error Rate* (PER) y *Bleu*.

Operaciones sobre matrices de alineamiento

Usando la funcionalidad que provee la herramienta, se estimó un modelo de secuencias RF con objeto de traducir el conjunto de test del corpus EUTRANS-I mediante el traductor Pharaoh. El modelo se estimó a partir de un conjunto de matrices de alineamiento a nivel de palabra que fueron obtenidos aplicando distintas operaciones sobre matrices. El tamaño máximo de segmento fue establecido en 6.

La Tabla 3 muestra las medidas WER, PER, Bleu y el número de secuencias extraídas según la operación aplicada (véase la sección 3.1). Como muestra la Tabla 3, la operación de simetrización produce los mejores resultados. Tal y como se esperaba, los peores resultados se obtienen cuando no se aplica ninguna operación. La operación de intersección extrae un mayor número de pares bilingües debido a que la matrices presentan una mayor cantidad de palabras no alineadas.

Estimación RF contra pML

Se llevaron a cabo experimentos con el corpus de

		EuTrans-I		Hansards		Europarl	
		Español	**Inglés**	**Francés**	**Inglés**	**Español**	**Inglés**
Entrenamiento	Frases	10 000		128 000		730 740	
	Palabras	97 131	99 292	2 062 403	1 929 186	15 725 136	15 222 505
	Vocabulario	686	513	37 542	29 414	113886	72 742
Test	Frases	2 996		500		3 064	
	Palabras	35 023	35 590	3 890	3 929	91 730	85 232
	Perplejidad (Trigramas)	–	3.62	–	79.4	–	115.0

Tabla 1. Estadísticas de los corpus EuTrans-I y Hansards

Op	**WER**	**PER**	**Bleu**	**#Phrases**
ninguna	8.0	6.7	0.894	29809
and	7.2	6.4	0.895	59715
or	8.0	5.6	0.900	14450
sum	8.0	5.7	0.900	14450
symmetr.	7.3	6.1	0.907	18686

Tabla 3. Efecto de las operaciones sobre matrices de alineamiento, tamaño máximo de secuencia=6, estimación no monótona RF, sobre el corpus EuTrans-I.

EuTrans-I aplicando las diferentes modalidades de estimación descritas en la sección 3.2.

La Tabla 4 muestra una comparativa entre las técnicas de estimación RF y pML en sus versiones monótona y no monótona. En la tabla se muestran: el número de pares bilingües extraídos (no se aplicaron operaciones sobre alineamientos y el tamaño máximo de secuencia se fijó en 6), el coste temporal del entrenamiento[4] y las medidas de error WER, PER y *Bleu* obtenidas cuando se traduce el corpus de test de la tarea EuTrans-I (de nuevo sin usar operaciones con matrices de alineamiento y con un tamaño máximo de secuencia igual a 6). Como era de esperar, la extracción monótona genera menos pares, y la estimación pML requiere mucho más tiempo que la RF, lo que se debe a lo comentado en la sección 3.2.

Estimación	**Tiempo**	**WER**	**PER**	**Bleu**	**#Pares**
RF mon	29	8.7	7.3	0.884	27429
RF	31	8.0	6.7	0.894	29809
pML mon	1969	8.6	7.2	0.890	25262
pML	2213	7.9	6.6	0.901	27788

Tabla 4. Comparación entre las técnicas de estimación RF y PML para el corpus de EuTrans-I.

En lo que respecta a las tasas de error obtenidas en experimentos de traducción, como puede observarse, la estimación pML obtiene resultados ligeramente mejores que la RF. Este resultado era de esperar, ya que se computó la log-verosimilitud según la ecuación 3 para ambas modalidades de estimación, y se observó que era mejor para la estimación pML, tanto para el corpus de entrenamiento como para el de test (ésto también es cierto si se sigue la aproximación del máximo).

Se llevaron a cabo experimentos para determinar el efecto del tamaño máximo de secuencia. La Tabla 5 muestra la influencia de este parámetro en la estimación RF. En la tabla se dan la medidas de error habituales, así como el tiempo de estimación y la cantidad de pares extraídos. Tal como muestra la Tabla 5, valores superiores a 4 no mejoran los resultados apreciablemente pero incrementan el tiempo de estimación. Se ha observado la misma situación para el caso de la estimación pML.

	Tiempo	**WER**	**PER**	**Bleu**	**#pares**
1	12.760	35.8	31.8	0.567	1656
3	17.370	9.8	8.4	0.867	10761
5	25.600	8.2	6.8	0.892	23871
7	38.320	8.0	6.7	0.894	33991

Tabla 5. Influencia del tamaño máximo de secuencia, estimación RF, para el corpus EuTrans-I.

Experimentos de calidad de la traducción

Finalmente, se realizaron experimentos de calidad de la traducción ajustando los parámetros de la herramienta Thot y los del traductor Pharaoh. En concreto, se estimó un modelo RF a partir de matrices de alineamiento simetrizadas. El tamaño máximo de secuencia se estableció en 6.

La Tabla 6 muestra las medidas de error WER, PER y *Bleu* para las tareas de EuTrans-I, Hansards y Europarl como resultado de traducir los conjuntos de test descritos en la tabla 1. Se comparó la calidad de la traducción obtenida por Pharaoh con la obtenida por otra herramienta de traducción: el *ISI ReWrite Decoder*, un traductor voraz de libre uso (ver [11]). En general, y como era de esperar, para todas las tareas estudiadas, los resultados obtenidos por Pharaoh fueron mucho mejores que los que se obtuvieron con el traductor voraz. Adicionalmente, los resultados obtenidos por Pharaoh podrían mejorarse ajustando los pesos que asigna el traductor a los modelos de traducción y lenguaje.

[4]Los resultados fueron obtenidos en un PC con procesador 1.6Ghz AMD Athlon y 512 MB de memoria utilizando Linux como sistema operativo. Todos los tiempos se dan en segundos.

Tarea	WER	PER	Bleu
EUTRANS-I	25.2/6.7	22.3/5.3	0.55/0.90
HANSARDS	57.0/52.8	52.0/48.1	0.22/0.31
EUROPARL	64.8/61.4	48.8/45.8	0.17/0.25

Tabla 6. Experimentos de calidad de la traducción para distintas tareas de traducción y los traductores *ISI ReWrite Decoder* y Pharaoh(izda. y dcha. respectivamente).

5. CONCLUSIONES

En este artículo hemos presentado las principales características de la herramienta Thot, que está disponible como software *open source* en *Sourceforge*: `http://thot.sourceforge.net/`.

El principal propósito del toolkit consiste en proporcionar una manera rápida y efectiva de entrenar modelos estadísticos de secuencias con objeto de ser utilizados en traducción automática u otras tareas relacionadas con el procesamiento del lenguaje natural. Se ha prestado especial atención al problema de la estimación a partir de corpus muy grandes, que tan importante viene siendo en los últimos tiempos.

Las principales características que ofrece la herramienta son:

- Diferentes formas de combinar alineamientos a nivel de palabra con vistas a la obtención de mejores matrices de alineamiento y mejores modelos de secuencias.
- Distintas modalidades de estimación de modelos de secuencias, de acuerdo con los enfoques descritos a lo largo del artículo, incluyendo un enfoque nuevo que hemos denominado estimación *pseudo_ML*.
- Estimación de modelos de frase a partir de corpus de tamaño arbitrario.
- Obtención de alineamientos a nivel de secuencia en base a un modelos de secuencias para un par de frases y su correspondiente matriz de alineamiento.

Creemos que la herramienta que se ha presentado (junto con otras herramientas de libre uso relacionadas con traducción automática estadística) pueden constituir un valioso recurso para la comunidad de traducción automática, pudiendo usarse para construir sistemas de traducción propios con bajos costes de desarrollo. Thot ha sido implementado siguiendo principios estándar de diseño como por ejemplo usabilidad y versatilidad en formatos. Estas características lo hacen interesante no sólo para expertos en el campo de la traducción automática sino para un público más general con conocimientos limitados sobre los fundamentos matemáticos de dicha disciplina.

Como trabajos futuros se plantea paralelizar diversas fases del entrenamiento y mejorar la documentación de la herramienta .

6. BIBLIOGRAFÍA

[1] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, y R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[2] Hermann Ney, Sonja Nießen, Franz J. Och, Hassan Sawaf, Christoph Tillmann, y Stephan Vogel, "Algorithms for statistical translation of spoken language," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 1, pp. 24–36, Jan. 2000.

[3] Kenji Yamada y Kevin Knight, "A syntax-based statistical translation model," in *Proc. of the 39th Annual Meeting of ACL*, Toulouse, France, July 2001, pp. 523–530.

[4] Franz Joseph Och, *Statistical Machine Translation: From Single-Word Models to Alignment Templates*, Ph.D. thesis, Computer Science Department, RWTH Aachen, Germany, October 2002.

[5] J. Tomás y F. Casacuberta, "Monotone statistical translation using word groups," in *Procs. of the Machine Translation Summit VIII*, Santiago de Compostela, Spain, 2001, pp. 357–361.

[6] Daniel Marcu y William Wong, "A phrase-based, joint probability model for statistical machine translation," in *Proc. of the EMNLP*, Philadelphia, USA, July 2002, pp. 1408–1414.

[7] R. Zens, F.J. Och, y H.Ñey, "Phrase-based statistical machine translation," in *Advances in artificial intelligence. 25. Annual German Conference on AI*, vol. 2479 of *Lecture Notes in Computer Science*, pp. 18–32. Springer Verlag, September 2002.

[8] P. Koehn, F. J. Och, y D. Marcu, "Statistical phrase-based translation," in *Proceedings of the HLT/NAACL*, Edmonton, Canada, May 2003.

[9] Phillip Koehn, "Pharaoh: a beam search decoder for phrase-based statistical machine translation models. User manual and description," Technical report, USC Information Science Institute, Dec. 2003.

[10] F. Nevado, F. Casacuberta, y J. Landa, "Translation memories enrichment by statistical bilingual segmentation," in *Proc. of the Fourth Int. Conf. on LREC*, Lisbon, 2004.

[11] Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, y Kenji Yamada, "Fast decoding and optimal decoding for machine translation," in *Proc. of the 39th Annual Meeting of ACL*, Toulouse, France, July 2001, pp. 228–235.

INTERFAZ PARA EL SISTEMA DE AYUDA A LA TRADUCCIÓN DEL PRHLT

María-Teresa González Cavero[1], *Francisco Casacuberta Nolla*[1]

[1]Dept. de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
46071 Valencia, Spain
{mgonzalez,fcn}@dsic.upv.es

RESUMEN

Hoy en día hay muchos estudios realizados sobre traducción automática, pero no se ha llegado a obtener una calidad en la traducción suficiente como para que los textos traducidos automáticamente no sean revisados por un traductor humano. Por todo ello se han investigado otras técnicas como es la traducción automática asistida por ordenador. En estas técnicas, un traductor humano interactua con un sistema de traducción automática que le sirve de herramienta de apoyo para ofrecerle dinámicamente una lista de traducciones que mejor complementan la frase que está traduciendo.

Para que el usuario interactue con el sistema de traducción automática es necesario implementar una interfaz de ayuda a la traducción que sea fácil y cómoda de utilizar. En otros proyectos como el TT2 ya se creó una interfaz de este tipo y en evaluaciones humanas se comprobaba que la ganancia de productividad obtenida se encontraba entre un 15 y un 20 %. En este trabajo proponemos otra interfaz basada en la del proyecto TT2. Lo que se pretende con esta interfaz es mejorar algunos aspectos de la interacción con el usuario que existían en la del proyecto TT2.

1. INTRODUCCIÓN

El procesamiento de lenguaje natural es una área de investigación de gran actividad y desarrollo que ha visto incrementada su relevancia y capacidad con los recientes avances de la informática.

Como resultado de la imposibilidad de obtener traducciones de alta calidad con los sistemas de traducción automática actuales, se introdujo una aproximación alternativa [1] que incorpora el conocimiento humano en el marco de la traducción automática. En este marco, llamado traducción asistida por ordenador (CAT[1]), un traductor humano interactúa con un sistema de traducción automática que le sirve de herramienta de apoyo para ofrecerle dinámicamente una lista de traducciones que mejor completan la frase que está traduciendo. De esta manera, el sistema automático aprovecha las correcciones del traductor humano para producir mejores traducciones.

El sistema de traducción asistida por ordenador que utilizaremos está basado en el paradigma de los modelos de estados finitos, más concretamente en transductores estocásticos de estados finitos (SFST[2])[2]. Una característica interesante de los SFSTs es que pueden ser aprendidos eficientemente de forma automática a partir de muestras finitas como puede ser un corpus bilingüe [3].

En otros trabajos realizados en proyectos como el TT2 [4] se ha demostrado que tener un sistema de traducción automática asistida (llamados sistemas CAT) por traductores profesionales proveen mejores traducciones y en menor tiempo. En evaluaciones realizadas por traductores humanos profesionales se comprobó que había un aumento del 15-20 % de la productividad al utilizar sistemas CAT [5]. Además, las evaluaciones automáticas nos muestran también que el usuario podría de esta manera sólo necesitar el 11'2 % de las pulsaciones de teclas que serían necesarias sin utilizar un sistema CAT [6].

Como se puede suponer, este tipo de sistemas necesitará algún tipo de interfaz con el usuario para que éste puede interactuar con el sistema de una manera fácil y cómoda. En proyectos como TT2 [4], ya se creó una interfaz con este fin. En este trabajo describiremos otra de parecidas características. A la hora de implementar la nueva interfaz, fue importante tener presente que el sistema desarrollado debe funcionar bajo condiciones de usabilidad y tiempo de respuesta muy estrictas para que pueda ser de utilidad a los traductores humanos.

En este documento en una primera sección describiremos la traducción automática asistida por ordenador, para luego describir en otra sección la interfaz creada para interactuar con el usuario de un sistema CAT. Por último, algunas conclusiones y trabajos futuros.

2. TRADUCCIÓN ASISTIDA POR ORDENADOR

Las técnicas actuales de traducción automática están todavía lejos de producir traducción de alta calidad. Este inconveniente nos lleva a una aproximación alternativa al

Este trabajo ha sido parcialmente financiado por el proyecto Itefte (CICYT TIC2003-08681-C02-02) y por la Agencia Valenciana de Ciencia y Tecnología (AVCiT) (Grupos I+D+I GRUPOS03/031).

[1]Del inglés *Computer Assisted Translation*

[2]Del inglés *Stochatic Finite State Transducer*

problema de la traducción que introduce el conocimiento humano en el campo de la traducción automática. En este marco, llamado Traducción Asistida por Ordenador (CAT), traductores humanos interactúan con un sistema de traducción que ofrece, de forma interactiva, la traducción que mejor completa la parte que ya ha sido traducida.

En [7] se propuso el paradigma CAT ejemplificado en el siguiente escenario: inicialmente el traductor humano parte de una frase en la lengua origen que debe traducirse a una frase del lenguaje destino. Dada esta frase en la lengua origen, el sistema de traducción es capaz de proporcionar una traducción en la lengua destino. En la mayoría de los casos esta traducción no es perfecta, por lo que requerirá la intervención del traductor humano para corregir los posibles errores. En el paradigma CAT se supone que el traductor empieza a leer y validar la frase ofrecida por el sistema de traducción de izquierda a derecha hasta cierta palabra, donde decide llevar a cabo una corrección. Por consiguiente, el traductor humano habrá aceptado un prefijo de la frase en la lengua destino y habrá introducido una corrección en forma de nuevo carácter. Este prefijo aceptado junto con el carácter introducido por el traductor humano dan lugar a un prefijo en la lengua destino, que será utilizado para refinar la búsqueda de la siguiente traducción del sistema. Como resultado de esta búsqueda, el sistema ofrecerá el sufijo de la frase de destino que mejor completa el prefijo validado por el traductor. Idealmente, la calidad del sufijo ofrecido por el sistema de traducción debería ir mejorando conforme el prefijo aceptado por el traductor humano es más largo.

Por tanto, el sistema estará compuesto por un módulo de traducción que ofrecerá traducciones de las frases en la lengua origen, y un segundo módulo que a partir del prefijo validado por el traductor humano y de las traducciones ofrecidas por el anterior módulo, defina cuál es el mejor sufijo que completa el prefijo del traductor humano.

A continuación decribimos formalmente el problema y cómo realizamos el proceso de traducción con esta clase de sistemas.

2.1. Traducción automática con SFSTs

En el marco estadístico de la traducción automática, dada una frase en la lengua origen $\mathbf{s}$, el objetivo es encontrar una frase en la lengua destino $\hat{\mathbf{t}}$ que cumpla lo siguiente:

$$\hat{\mathbf{t}} = \underset{\mathbf{t}}{\operatorname{argmax}} \Pr(\mathbf{t} \mid \mathbf{s}) = \underset{\mathbf{t}}{\operatorname{argmax}} \Pr(\mathbf{s} \mid \mathbf{t}) \cdot \Pr(\mathbf{t}) \quad (1)$$

La Ecuación 1 también puede escribirse en forma de probabilidad conjunta que puede ser modelada por un SFST $\mathcal{T}$:

$$\begin{aligned} \hat{\mathbf{t}} &= \underset{\mathbf{t}}{\operatorname{argmax}} \Pr(\mathbf{s} \mid \mathbf{t}) \cdot \Pr(\mathbf{t}) \\ &= \underset{\mathbf{t}}{\operatorname{argmax}} \Pr(\mathbf{t}, \mathbf{s}) \\ &\approx \underset{\mathbf{t}}{\operatorname{argmax}} \Pr_{\mathcal{T}}(\mathbf{t}, \mathbf{s}) \end{aligned} \quad (2)$$

Los SFSTs se han utilizado con éxito en muchas tareas de traducción [8]. Además, existen algoritmos de búsqueda eficientes como Viterbi para la búsqueda del mejor prefijo y el algoritmo de enumeración recursivo (REA) para *n*-best.

Una posible manera de inferir SFSTs es con el método GIATI [9].

2.2. Búsqueda interactiva en un SFST

El concepto de búsqueda interactiva está estrechamente relacionada con el paradigma CAT. Este paradigma introduce un nuevo factor $\mathbf{t}_p$ en la Ecuación 1 de traducción automática. $\mathbf{t}_p$ representa el prefijo de una frase traducida obtenida como resultado de la interacción entre un traductor humano y un sistema de traducción automática.

Un ejemplo de esta interacción se puede ver en la Figura 1. En cada iteracción, un prefijo ($\mathbf{t}_p$) de una frase traducida ha sido aceptado por un traductor humano en una iteración anterior y el sistema CAT computa los mejores (o n-best) sufijos como hipótesis ($\hat{\mathbf{t}}_s$) para completar el prefijo.

Dado $\mathbf{t}_p\hat{\mathbf{t}}_s$, el ciclo CAT continúa cuando el usuario establece como aceptado un prefijo nuevo y más largo. Para ello, el traductor humano ha aceptado una parte (a) de $\mathbf{t}_p\hat{\mathbf{t}}_s$ (o más típicamente $\hat{\mathbf{t}}_s$). Después de este punto, el usuario puede teclear alguna tecla (k) con el fin de corregir alguna parte incorrecta. Por lo tanto, el nuevo prefijo está formado por $\mathbf{t}_p$ seguido por la parte aceptada de una de las sugerencias propuestas por el sistema (a), más el texto (k) introducido por el usuario. Ahora este prefijo, $\mathbf{t}_pak$, se convertirá en el nuevo $\mathbf{t}_p$, comenzando con ello un nuevo ciclo predictivo del sistema CAT.

Una forma simple de formalizar el proceso de sugerencias de hipótesis en el sistema CAT sería como sigue a continuación [6]. Dada una frase fuente $\mathbf{s}$ y un prefijo validado por el usuario de una frase traducida $\mathbf{t}_p$, la búsqueda de un sufijo para completar la traducción y que maximice la probabilidad a posteriori sobre todos los sufijos es:

$$\hat{\mathbf{t}}_s = \underset{\mathbf{t}_s}{\operatorname{argmax}} \Pr(\mathbf{t}_s \mid \mathbf{s}, \mathbf{t}_p) \,. \quad (3)$$

Teniendo en cuenta que $\Pr(\mathbf{t}_p \mid \mathbf{s})$ no depende de $\mathbf{t}_s$, podemos escribir:

$$\hat{\mathbf{t}}_s = \underset{\mathbf{t}_s}{\operatorname{argmax}} \Pr(\mathbf{t}_p\mathbf{t}_s \mid \mathbf{s}) \,. \quad (4)$$

donde $\mathbf{t}_p\mathbf{t}_s$ es la concatenación de un prefijo dado $\mathbf{t}_p$ y un sufijo $\mathbf{t}_s$. Esta distribución conjunta puede ser modelada adecuadamente por SFSTs [10].

ITER-0	($\mathbf{t}_p$)	( )
ITER-1	($\hat{\mathbf{t}}_s$)	(*Haga clic para cerrar el diálogo de impresión*)
	(**a**)	(Haga clic)
	(**k**)	(**en**)
	($\mathbf{t}_p$)	(Haga clic en)
ITER-2	($\hat{\mathbf{t}}_s$)	(*ACEPTAR para cerrar el diálogo de impresión*)
	(**a**)	(*ACEPTAR para cerrar el*)
	(**k**)	(**cuadro**)
	($\mathbf{t}_p$)	(Haga clic en ACEPTAR para cerrar el cuadro)
FINAL	($\hat{\mathbf{t}}_s$)	(*de diálogo de impresión*)
	(**a**)	(*de diálogo de impresión*)
	(**k**)	(**#**)
	($\mathbf{t}_p \equiv \mathbf{t}$)	(Haga clic en ACEPTAR para cerrar el cuadro de diálogo de impresión)

Figura 1. Ejemplo de un sistema de interacción CAT para traducir al Español la frase en Inglés “Click OK to close the print dialog” extraida de un manual de impresoras. Cada paso comienza con un prefijo de salida préviamente fijado $\mathbf{t}_p$, para el cual el sistema sugiere un sufijo $\hat{\mathbf{t}}_s$. Entonces el usuario acepta parte de este sufijo (a) y teclea algunas teclas (k), con el fin de solucionar la parte equivocada de $\hat{\mathbf{t}}_s$. Esto produce un nuevo prefijo, compuesto por el prefijo de la anterior iteración y la parte aceptada y tecleada, (a,k), para ser utilizada como $\mathbf{t}_p$ en el siguiente paso. El proceso termina cuando el usuario introduce el caracter especial “#”. En la traducción final, $\mathbf{t}$, todos los textos que han sido tecleados por el usuario han sido subrayados.

La solución a este problema de maximización ha sido realizado en dos fases. En la primera se extrae un grafo de palabras del SFST dada una frase fuente $\mathbf{s}$. En la segunda fase, la búsqueda del mejor o mejores sufijos se lleva a cabo de acuerdo a la aproximación de Viterbi sobre el grafo de palabras de la primera fase dado un prefijo $\mathbf{t}_p$ de una traducción.

3. INTERFAZ CAT

En esta sección describiremos cómo hemos comunicado la interfaz con el sistema de traducción asistida por ordenador (motor de traducción), las funciones que fueron necesarias para esta comunicación, las partes que la componen y por último el funcionamiento que tiene y las facilidades que proporciona al usuario.

3.1. Comunicación entre la interfaz y el motor de traducción

La interfaz fue desarrollada en Java, mientras que el motor de traducción en C++. Por ello, es necesario comunicar estos dos programas de alguna manera. La forma más directa de integrar código nativo en programas escritos en Java es a partir de JNI (Java Native Interface)[11]. Hay otras como los sockets que nos permitirían tener motores de traducción en código nativo sin la necesidad que estuvieran implementados en C. En este trabajo nos centraremos exclusivamente en la integración de código nativo a través de JNI y dejaremos para posteriores versiones con más de un motor de traducción la comunicación por sockets.

Las funciones que son necesarias en este proceso de comunicación son las que se expecifican a continuación:

- *init*: Realiza la carga del traductor en el motor de traducción.
- *setSource*: Se le indica la frase fuente al motor de traducción.
- *setTarget*: Se le indica la traducción aceptada por el usuario hasta el momento al motor de traducción.
- *suggest*: Soluciones alternativas que el motor de traducción proporciona como sugerencia de traducción dada una frase fuente y la traducción aceptada hasta el momento.

Estas funciones estarán tanto en Java como en C. En Java como funciones de una clase que cargará una librería dinámica que contendrá la parte de C. Y en C como un .h y .c expecíficos para que las funciones en C sean “entendibles” por el Java.

3.2. Componentes de la interfaz

Las partes que componen la interfaz y que se pueden observar en la Figura 2 son las siguientes:

- *Menús*: desde donde se pueden realizar tareas como cambiar de traductor, abrir/cerrar un fichero, guardar las traducciones a fichero, etc.
- *Frases Anteriores*: son las frases fuente junto con su traducción (en el caso que la tenga) que se encuentran antes de la frase fuente que se está traduciendo en este momento (frase fuente actual) en el fichero que se ha cargado o las que se han guardado antes que la frase fuente actual en el caso que no se realicen traducciones desde fichero.
- *Frases Posteriores*: son las frases fuente junto con su traducción (en el caso que la tenga) que se encuentran después de la frase fuente que se está traduciendo en este momento (frase fuente actual) en el fichero que se ha cargado o las que se han guardado después que la frase fuente actual en el caso que no se realicen traducciones desde fichero.

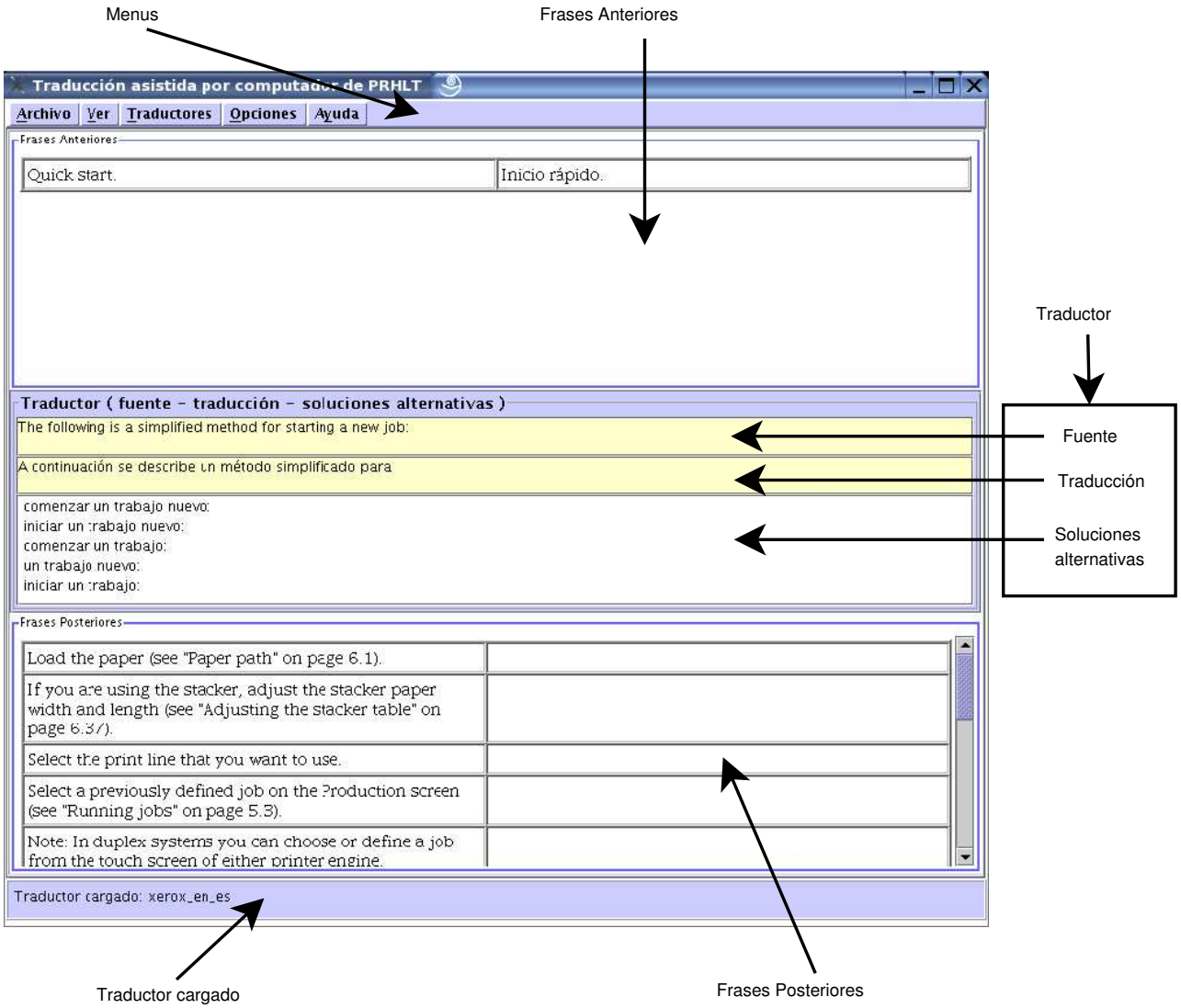


Figura 2. Componentes de la interfaz

- *Traductor*: es la parte en la interfaz en donde se visualiza siempre el proceso de traducción.
  - *Fuente*: es la frase a traducir en este momento (actual).
  - *Traducción*: es tal y como está en un momento dado la traducción de la frase fuente (prefijo aceptado).
  - *Soluciones alternativas*: son las traducciones (sufijos) que se proporcionan al usuario como posibles traducciones de la frase fuente.
- *Traductor cargado*: es un campo que nos muestra en cada momento el traductor que utiliza el motor de traducción para proporcionar las soluciones alternativas.

3.3. Funcionamiento

El proceso de traducción que se lleva a cabo sigue los siguientes pasos:

1. Inicializar (cargar) el traductor[3].
2. Pasar al motor de traducción la frase fuente.
3. Pasar al motor de traducción la frase traducida aceptada hasta el momento[4].
4. Pedir al motor de traducción las soluciones alternativas para la traducción de una frase fuente dada teniendo en cuenta la frase traducida hasta el momento[5].

[3]Este paso sólo se realiza al comienzo de la aplicación y cuando el usuario seleccione un cambio de traductor.

[4]Al principio se le pasa la cadena vacía, ya que no se tiene traducción.

[5]Devuelve 5 opciones.

ITER-0	Frase traducida	( )
ITER-1	Soluciones alternativas	(*A continuación se describe un método muy simple para comenzar un trabajo nuevo:*) (*A continuación se describe un método muy simple para comenzar un trabajo:*) (*El continuación se describe un método muy simple para comenzar un trabajo nuevo:*) (*La continuación se describe un método muy simple para comenzar un trabajo nuevo:*) (*Continuación se describe un método muy simple para comenzar un trabajo nuevo:*)
	Prefijo seleccionado	(A continuación se describe un método)
ITER-2	Soluciones alternativas	(*muy simple para comenzar un trabajo nuevo:*) (*muy simple para comenzar un trabajo:*) (*muy simple para comenzar un:*) (*muy simple para comenzar un trabajo nuevo.*) (*muy simple para comenzar un trabajo nuevo*)
	Modificación	(*simplificado para*)
FINAL	Soluciones alternativas	(*comenzar un trabajo nuevo:*) (*iniciar un trabajo nuevo:*) (*comenzar un trabajo:*) (*un trabajo nuevo:*) (*iniciar un trabajo:*)
	Frase traducida	(A continuación se describe un método simplificado para comenzar un trabajo nuevo:)

Figura 3. Ejemplo de interación de la interfaz con el usiario para traducir del castellano la frase en inglés “The following is a simplified method for starting a new job:” extraida de un manual de impresoras. En cada paso se especifican las soluciones alternativas proporcionadas por el sistema y las operaciones de selección y modificación que realiza el usuario. El proceso termina cuando el usuario cambia de frase. En la traducción final, última *Frase traducida*, todos los textos que han sido tecleados por el usuario han sido subrayados.

5. Interacción con el usuario[6].
   - Elección como traducción una de las soluciones alternativas.
   - Elección de parte de una traducción.
   - Modificación de la traducción.
6. Volver a paso 3 después de cada una de las interacciones anteriores realizadas por el usuario.

En la Figura 3 se puede ver un ejemplo de este proceso.

Algunas de las funcionalidades se especifican a continuación.

3.3.1. Carga de varios traductores

La interfaz permite realizar traducciones de diferentes tareas y en diversas direcciones. Esto se lleva a cabo cargando el traductor correspondiente en cada caso. Por ejemplo, se tiene un traductor de manuales de impresoras del inglés al castellano, otro sobre partes meterológicos del euskera al castellano y viceversa, etc. En la interfaz se mostrará en la parte inferior el traductor que está cargado para que el usuario sepa en cada momento sobre qué tarea puede realizar traducciones.

3.3.2. Abrir ficheros

Hay dos maneras de realizar traducciones. La primera es introduciendo directamente la frase fuente en el campo correspondiente (modo directo) y la otra es cargar un fichero de frases y seleccionar una para traducir (modo desde fichero). Este proceso se realiza de la misma manera que en otras aplicaciones. Junto con la interfaz hay unos ficheros de prueba para cada traductor, para probarlos sólo hay que seleccionar el que corresponda en cada caso.

[6]Puede realizar una o varias de estas opciones.

3.3.3. Aumento/disminución de la letra

A los campos afectados durante el proceso de traducción se les puede aumentar o disminuir la letra al tamaño que le sea cómodo al usuario. Dicha función se encuentra en el menú Ver y permite aumentar, disminuir o volver al tamaño original de letra.

3.3.4. Modos de traducción: de fichero/directo

Como ya se ha comentado en apartados anteriores, se permite realizar traducciones o bien de un modo directo o bien de un modo desde fichero. El proceso de traducción es el mismo para ambos casos. Lo único que varía es la manera de selecionar una frase a traducir. En el primer caso se introduce en el campo de frase fuente y cuando se quiere cambiar de fuente se selecciona la opción Nueva Fuente del menú Archivo. En el otro caso (de fichero) se selecciona una de las frases anteriores o posteriores.

3.3.5. Guardar traducciones

En el modo directo, primero hay que decirle al sistema que esa traducción la queremos guardar (Archivo/Guardar traducción) y luego decirle que la guarde en fichero (Archivo/Guardar traducciones como...). En el modo desde fichero, lo único que hay que hacer es que la guarde a fichero.

3.3.6. Cambio de foco

Estas opciones del menú se crearon para facilitar el proceso de traducción al usuario. Estas opciones tienen

asociadas unas teclas rápidas para agilizar el cambio de foco.

3.3.7. Ayuda teclas rápidas

Se han asociado una serie de teclas rápidas a ciertas utilidades de la interfaz para facilitar su utilización. Éstas se pueden consultar en la ayuda en la interfaz.

4. CONCLUSIONES Y TRABAJO FUTURO

Se ha creado una interfaz para la ayuda a la traducción basada en la ya existente en el proyecto TT2 [4]. Esta nueva interfaz, a diferencia de la anterior permite cambiar el tranductor en tiempo de ejecución, permitiendo de esta manera realizar traducciones sobre diferentes tareas sin necesidad de salir del programa.

Es una interfaz fácil de utilizar y en la que el proceso de traducción está siempre en el mismo punto de la interfaz. Ésto hace que el usuario siempre sepa dónde le van a aparecer las traducciones y las opciones que se le proponen.

Otra funcionalidad extra es la de poder aumentar y disminuir la letra de los campos utilizados durante la traducción dependiendo del gusto del usuario o para permitir una mejor visualización en demos en donde se tienen que visualizar desde bastante distancia.

Además también se permite realizar traducciones y guardarlas sin éstas tener que estar en un fichero, es decir, el usuario puede introducir una frase de la tarea del transductor cargado en cada momento y poder obtener una traducción de ella y su posterior almacenamiento en disco. Esta interfaz también permite dejar frases sin terminar de traducir y continuar con ellas en otro momento.

En trabajos futuros se pretende introducir nuevas funcionalidades como sería el caso de permitir cambiar el motor de traducción o permitir cargar diferentes textos planos y ella misma que se encargue de segmentarlos en frases para poder realizar el proceso de traducción de una manera más rápida. En estos momentos se permite cambiar de traductor, pero el motor de traducción es siempre el desarrollado en el marco del proyecto TT2 y las frases desde fichero se le tienen que pasar ya segmentadas. Además, se pretende crear una interfaz que sea accesible desde la web para mayor difusión del sistema.

5. BIBLIOGRAFÍA

[1] J. Civera, J. M. Vilar, E. Cubel, A. L. Lagarda, S. Barrachina, E. Vidal, F. Casacuberta, D. Picó, y J. González, "From machine translation to computer assisted translation using finite-state models," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP04)*, Barcelona, 2004.

[2] E. Vidal y F. Casacuberta, "Learning finite-state models for machine translation," in *Grammatical Inference: Algorithms and Applications. Proceedings of the 7th International Coloquium ICGI 2004*, vol. 3264 of *Lecture Notes in Artificial Intelligence*, pp. 16–27. Springer, Athens, Greece, October 2004, Invited conference.

[3] Enrique Vidal, Francisco Casacuberta, y Pedro García, "Grammatical inference and automatic speech recognition," in *New Advantages and Trends in Speech Recognition and Coding*, A. Rubio, Ed., vol. 147 of *NATO-ASI Series F:Computer and Systems Sciences*, pp. 174–191. Springer-Verlag, 1995.

[4] Atos Origin, Instituto Tecnológico de Informática, RWTH Aachen, RALI Laboratory, Celer Soluciones and Société Gamma and Xerox Research Centre Europe, "TransType2 - Computer Assisted Translation. Project Technical Annex.," 2001.

[5] E. Macklovitch, "TransType2: The Last Word," in *Proc. of LREC'06*, 2006.

[6] J. Civera, A.L. Lagarda, E. Cubel, F. Casacuberta, E. Vidal, J.M. Vilar, y S. Barrachina, "A Computer-Assisted Translation Tool based on Finite-State Technology," in *Proc. of EAMT'06*, 2006, pp. 33–40.

[7] Philippe Langlais, George Foster, y Guy Lapalme, "Unit completion for a computer-aided translation typing system," *Machine Translation*, vol. 15, no. 4, pp. 267–294, 2000.

[8] F. Casacuberta, H.Ñey, F. J. Och, E. Vidal, J. M. Vilar, S. Barrachina, I. Garcia-Varea, D. Llorens, C. Martinez, S. Molau, F.Ñevado, M. Pastor, D. Pico, y A. Sanchis, "Some approaches to statistical and finite-state speech-to-speech translation," *Computer Speech and Language*, vol. 18, pp. 25–47, 2004.

[9] F. Casacuberta y E. Vidal, "Machine translation with inferred stochastic finite-state transducers," *Computational Linguistics*, vol. 30, no. 2, pp. 205–225, 2004.

[10] J. Civera et al., "A syntactic pattern recognition approach to computer assisted translation," in *Advances in Statistical, Structural and Syntactical Pattern Recognition*, A. Fred, T. Caelli, A. Campilho, R. P.W. Duin, y D. de Ridder, Eds., Lecture Notes in Computer Science, pp. 207–215. Springer-Verlag, 2004.

[11] Sheng Liang, *The Java Native Interface: Programmer's Guide and Specification*, Addison Wesley Longman, 1999, ISBN: 0201325772.

INVENTARIO DE FRECUENCIAS FONÉMICAS Y SILÁBICAS DEL CASTELLANO ESPONTÁNEO Y ESCRITO

Antonio Moreno Sandoval[1], Doroteo Torre Toledano[1,2] Natalia Curto[1] y Raúl de la Torre[1]

{antonio.msandoval,doroteo.torre}@uam.es {natalia,raul}@maria.lllf.uam.es

[1]LLI, [2]ATVS, Universidad Autónoma de Madrid, SPAIN.

RESUMEN

Este artículo presenta dos inventarios de frecuencias – fonémico y silábico – del castellano obtenidos a partir del corpus C-ORAL-ROM, que recoge español oral espontáneo en distintos contextos y registros. Estos inventarios se han desarrollado mediante un transcriptor fonológico y silábico cuyos resultados para el corpus C-ORAL-ROM han sido en su mayor parte revisados manualmente. Los inventarios incluyen la frecuencia absoluta de aparición de los diferentes fonemas y sílabas. Estos datos se han examinado junto a los extraídos de un corpus comparable de texto escrito y se ha hallado evidencia de que los inventarios de frecuencias obtenidos hasta ahora, basados fundamentalmente en textos, no describen adecuadamente el castellano oral espontáneo.

1. INTRODUCCIÓN

El primer inventario de frecuencias de fonemas para el castellano fue el elaborado por Zipf y Rogers [7] en 1939, sirviéndose de la descripción fonológica de Navarro Tomás. Posteriormente se han realizado varios estudios sobre esta cuestión, que aparecen recogidos en la Tabla 1. Dicha tabla indica el número total de fonemas/letras considerados para el cálculo de frecuencias así como el tipo de corpus empleado (oral o escrito).

Autores	nº fon/let.	Tipo
Zipf y Rogers (1939)	5.000	escrito
Navarro Tomás (1946) [2]	20.000	Escrito
Guirao y Borzone (1972) [3]	62.980	Escrito
Quilis y Esgueva (1980) [4]	160.000	Oral
Rojo (1991) [5]	3.641.915	Escrito
UAM (2006)	1.200.000	Oral

Tabla 1. *Estudios de frecuencias de los fonemas del castellano.*

Este trabajo ha sido finaciado parcialmente por el MEC-CICYT (TIN2004-07588-C03-02) y la Comunidad Autónoma de Madrid y la Universidad Autónoma de Madrid (05/TIC/001).

Los estudios relativos a las sílabas son aún más escasos: Álvarez, Carreiras y De Vega [7] y Alameda y Cuetos [1] son los más recientes. Los primeros autores consideraron 41.592 sílabas. Los últimos emplearon 3.930.954 en sus cálculos. En ambos casos, las fuentes utilizadas fueron textos escritos sin transcribir fonológicamente.

En el presente estudio se ha manejado un total de 1.244.411 fonemas y 558.982 sílabas.

La novedad de esta investigación viene dada por el empleo como fuente de un corpus de habla espontánea, diez veces mayor del utilizado por Quilis y Esgueva [4] y con un gran número de hablantes. El LLI-UAM ha compilado los dos corpus de español oral más importantes: CORLEC[9] y C-ORAL-ROM [10]. Este último ha constituido la base para nuestro estudio. Este corpus contiene más de 348.000 palabras (incluyendo algunas marcas prosódicas), de 192 textos transcritos. En total, 429 hablantes distintos y más de 42 horas de grabación. El corpus se halla dividido en tres grandes grupos: informal (165.210 palabras), formal (70.924) y medios de comunicación (97.170). Un pequeño subcorpus de 14,760 palabras recoge conversaciones telefónicas. La calidad de las transcripciones viene avalada por una validación interna (cada texto fue verificado por al menos 3 lingüistas) y externa (ELDA). La alineación de sonido y transcripción supone la garantía última de que la transcripción se ajusta a la grabación.

La sección 2 describe la metodología general seguida para la obtención del inventario y la sección 3 está dedicada a describir el transcriptor que se ha empleado. La sección 4 recoge los resultados más importantes e incluye una comparación de los corpus escrito y oral. La sección 5 presenta las conclusiones que hemos obtenido de los resultados.

2. METODOLOGÍA

El corpus C-ORAL-ROM incluye la transcripción ortográfica de todas las grabaciones pero no la fonémica ni la silábica. Éstas han sido obtenidas de manera semi-automática.

Para la compilación del inventario del corpus oral se han dado los siguientes pasos:

1. El punto de partida fue un transcriptor desarrollado previamente (sección 3).
2. Basándose en el corpus CORLEC, los investigadores del LLI identificaron posibles problemas del transcriptor.
3. A partir de los resultados de esta investigación se realizaron mejoras en las reglas del transcriptor.
4. Se obtuvo una transcripción fonémica y silábica preliminar de C-ORAL-ROM.
5. Las transcripciones se revisaron manualmente: todas ellas fueron revisadas por al menos un lingüista y el 60% de ellas por dos personas distintas. Con estas revisiones, se volvieron a mejorar las reglas y las excepciones del transcriptor.
6. Finalmente, se llevó a cabo una transcripción definitiva que ha sido la base para los inventarios que aquí presentamos.

3. DESARROLLO DEL TRANSCRIPTOR

El transcriptor empleado en este trabajo utiliza un mecanismo de reglas de reescritura dependientes del contexto y excepciones, con la palabra como unidad. En primer lugar, la palabra se busca en una lista de excepciones que contiene transcripciones fonológicas y silábicas para casos especiales (normalmente extranjerismos). Si la palabra se halla en la lista se toma su correspondiente transcripción. Si la palabra no figura en la lista se aplica una serie de reglas de reescritura dependientes del contexto para obtener la transcripción fonológica. Después se derivan las sílabas a partir de las vocales mediante otro conjunto de reglas. Por último, otro conjunto de reglas interviene para determinar si las sílabas son acentuadas o no dependiendo de las reglas acentuales y ortográficas que rigen el empleo de la tilde. La transformación de la representación ortográfica de una palabra a su transcripción fonológica se basa en reglas de reescritura dependientes del contexto con el siguiente formato:

signo → [context-i] nuevo(s)-signo(s) [context-d]

en el que [context-i] y [context-d] son opcionales y pueden incluir cualquier número de signos (que pueden representar letras o fonemas dependiendo de la regla). Signo es la letra o fonema que debe reescribirse y nuevo(s) -signo(s) puede ser cero, uno o más signos que representan letras o fonemas (según la regla). Estas reglas se aplican una por una en un orden predeterminado. Dada la regularidad de la correspondencia entre letras y sonidos en español, bastó con 50 reglas para obtener buenos resultados. A esto también contribuyó nuestra decisión de utilizar un conjunto mínimo de 23 fonemas y de tomar en cuenta únicamente las transcripciones fonológicas canónicas, obviando variantes regionales y reducciones. Una limitación importante de esta transformación es el hecho de que la unidad de transcripción empleada es la palabra y no oraciones enteras con lo que fenómenos fonológicos intraoracionales no pueden tratarse.

Una vez que la palabra se ha transformado en secuencia de fonemas, cada vocal se marca inicialmente como una sílaba. A continuación, unas reglas determinan qué pares de vocales pertenecen a la misma sílaba y cuáles a sílabas diferentes. A continuación, ocho reglas añaden a la sílaba que corresponda las consonantes que aparecen antes y después de la(s) vocal(es). Si al final de este proceso hay consonantes no asignadas a ninguna sílaba, se informa de un error de silabificación, lo que sucedió fundamentalmente con palabras extranjeras y con acrónimos, que se incluyeron en la lista de excepciones.

Cuando la palabra ha sido transcrita fonológicamente y silabificada, otro conjunto de reglas asigna acento a una de las sílabas de la palabra según las convenciones ortográficas del español. En este punto la limitación de tomar la palabra como unidad es de nuevo importante, dado que es frecuente en español oral el agrupar varias palabras (p.e. el nombre y palabras funcionales como el artículo) con una única sílaba acentuada, lo que nuestro transcriptor no puede reflejar.

Para desarrollar el inventario hemos afinado las reglas y las excepciones del transcriptor, basándonos en las correcciones manuales de las transcripciones automáticas del corpus C-ORAL-ROM. Debe señalarse que sólo un 2% de las palabras transcritas automáticamente contenían un error de transcripción fonémico o silábico. Para el presente trabajo hemos omitido la información acentual y hemos tenido en cuenta únicamente la transcripción fonológica y la silabificación. Por lo tanto dos sílabas o fonemas que difieren únicamente en el acento se consideran equivalentes. Las excepciones corresponden sobre todo a palabras extranjeras y acrónimos con los cuales el transcriptor produjo errores de silabificación.

4. RESULTADOS

Con el fin de facilitar una comparación entre las frecuencias obtenidas a partir del corpus escrito y las del oral, seleccionamos aleatoriamente 500.000 palabras del corpus de una agencia de noticias (EFE) de 150 millones de palabras. El procedimiento de selección tomaba una palabra de cada 300. De esta forma contamos con un corpus escrito representativo con el que comparar nuestro corpus oral.

Fonemas	Español oral				Español escrito			
	Lexicón		Corpus		Lexicón		Corpus	
	Fr. Absoluta	Fr. Relat.	Fr. Absoluta	Fr. Relat.	Fr. Absoluta	Fr. Relat.	Fr. Absoluta	Fr. Relat.
a	23294	13.87	152664	12.27	46488	13.48	323783	12.89
b	5036	2.99	31126	2,50	12513	3.63	64170	2.55
θ	3623	2.15	18940	1.52	7469	2.16	50301	2.00
ʧ	592	0.35	3744	0.30	1043	0.30	4463	0.18
d	7521	4.48	54284	4.36	14479	4.20	136187	5.42
e	18337	10.92	188196	15.12	34510	10.00	320140	12.74
f	1630	0.97	6217	0.50	4381	1.27	23042	0.92
g	1995	1.19	11359	0.91	4108	1.19	26138	1.04
i	14623	8.71	89799	7.22	31626	9.17	190756	7.59
x	1548	0.92	7681	0.62	3031	0.88	19362	0.77
k	6981	4.16	55863	4.49	13466	3.90	95427	3.80
l	5627	3.35	56107	4.51	14633	4.24	137148	5.46
m	5438	3.24	39278	3.15	9847	2.85	69445	2.76
n	11394	6.78	87775	7.05	23366	6.77	178012	7.09
ɲ	320	0.19	2427	0.19	1451	0.42	7729	0.31
o	15399	9.17	129208	10.38	31187	9.04	234238	9.32
p	4582	2.73	34135	2.74	7899	2.29	68687	2.73
r	1783	1.06	5236	0.42	6561	1.90	25016	0.99
ɾ	10992	6.55	63702	5.12	23603	6.84	155632	6.19
s	12453	7.42	100881	8.11	23998	6.96	184085	7.33
t	9168	5.46	56287	4.52	17201	4.99	108398	4.31
u	4601	2.74	39146	3.14	9091	2.64	76390	3.04
ʎ	959	0.57	10356	0.83	2905	0.84	13307	0.53
TOTAL	167896	100	1244411	100	344856	100	2511856	100

Tabla 1. *Frecuencia de los fonemas españoles.*

El procedimiento de extracción del inventario ha sido el mismo para los dos corpus.

Primero se extraen dos listas de palabras de cada corpus: el conjunto de las formas que aparecen y este mismo conjunto más el número de apariciones de cada forma. Es decir, en el primero estarían sólo los "types" y en el segundo tanto "types" como "tokens". El transcriptor se aplica a ambas listas y se obtiene respectivamente un léxico y un corpus fonológicos con las apariciones de cada palabra.

A continuación se silabifican las formas y se obtienen dos conjuntos de sílabas fonológicamente transcritas: las correspondientes al léxico y las correspondientes al corpus. Las sílabas de los dos conjuntos se cuentan y se ordenan por frecuencia, de mayor a menor. Con esto averiguamos la distribución de las sílabas en el léxico y la distribución real en el corpus.

En este punto obtenemos una tabla de sílabas para el léxico y otra para el corpus, datos de las apariciones de cada sílaba, su frecuencia relativa al total y su distribución (Tabla 2). Ahora se pueden examinar las distintas estructuras de sílaba y hallar su frecuencia en el léxico y en el corpus. Por limitaciones de espacio presentamos únicamente las 10 sílabas más frecuentes. El orden de presentación es la frecuencia del corpus oral.

El último paso supone el recuento de fonemas y el cálculo de la frecuencia total de cada unidad. A continuación se repite el proceso tomando en cuenta el contexto silábico. Dada la frecuencia de aparición de cada fonema se pueden hallar las probabilidades de su presencia en cualquier combinación. La Tabla 1 muestra los resultados de los 23 fonemas.

La Figura 1 muestra la distribución de las sílabas en el corpus escrito y en el oral. Un resultado interesante y significativo es que las primeras 100 sílabas representan más del 80% del corpus oral. Las primeras 650 sílabas cubren más del 99 % del corpus.

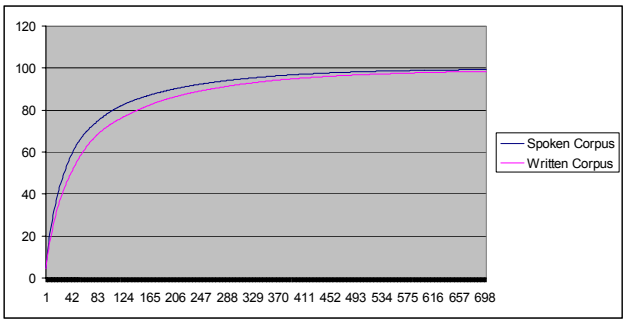


Figura 1. *Distribución de las sílabas en los corpus oral y escrito. La figura muestra la frecuencia relativa acumulada de las sílabas en español, ordenadas en orden decreciente.*

Corpus oral						Corpus escrito					
Lexicón			Corpus			Lexicón			Corpus		
.a.	2506	3,53	.a.	27606	4,94	.a.	4220	3,15	.de.	46748	4,49
.do.	1854	2,61	.ke.	21070	3,77	.ta.	2918	2,18	.a.	37021	3,55
.ta.	1621	2,28	.de.	19638	3,51	.do.	2613	1,95	.la.	27138	2,61
.te.	1406	1,98	.es.	13703	2,45	.ka.	2392	1,78	.ta.	17885	1,72
.ka.	1314	1,85	.i.	13102	2,34	.ti.	2264	1,69	.ke.	17704	1,70
.de.	1235	1,74	.no.	12781	2,28	.te.	2263	1,69	.en.	17203	1,65
.ti.	1182	1,66	.te.	10620	1,89	.ra.	2138	1,60	.do.	16840	1,62
.to.	1026	1,44	.el.	10282	1,84	.na.	1936	1,45	.te.	16610	1,59
.ko.	1019	1,43	.la.	10281	1,84	.de.	1828	1,37	.na.	15872	1,52
.ra.	992	1,39	.do.	10172	1,82	.ko.	1828	1,37	.ma.	15463	1,48

Tabla 2. *Las 10 sílabas más frecuentes en castellano.*

Por último mostramos el orden de distribución silábica por tipos (distintas combinaciones de vocal y consonante que aparecen en las sílabas en castellano)

Tipo de sílaba	Frecuencia Relativa	Tipo de sílaba	Frecuencia relativa
.CV.	51,35	.CVV.	3,37
.CVC.	18,03	.CVVC.	3,31
.V.	10,75	.CCV.	2,96
.VC.	8,60	.CCVC.	0,88

Tabla 3. *Distribución de frecuencias por tipos de sílaba en el corpus oral.*

5. CONCLUSIONES Y TRABAJO FUTURO

Éste es el primer inventario de frecuencias de fonemas y sílabas del castellano que emplea un corpus oral y otro escrito de tamaños comparables, y que emplea los mismos criterios y herramientas para segmentar las unidades.

Dos conclusiones importantes pueden derivarse de los datos:

1. El corpus escrito y el oral arrojan diferentes frecuencias. El orden de algunas unidades y el porcentaje de uso son diferentes. Esto es especialmente notable en el caso de las vocales /a, e, o/. Por lo tanto, el entrenamiento con modelos de lengua basados en textos escritos producirá peores resultados que el entrenamiento con textos orales.
2. Unas pocas sílabas en castellano permiten cubrir una parte significativa de un texto. De esto se sigue que el empleo de sílabas en lugar de fonemas como unidades en el desarrollo de tecnologías del lenguaje para el español parece prometedor. Nuestro equipo continuará investigando la segmentación en sílabas para el entrenamiento de sistemas de reconocimiento de español espontáneo.

Una vez que hayamos estimado la relevancia estadística de estos resultados con respecto a otro corpus oral similar, como nuestro corpus CORLEC, podremos ofrecer las garantías de un trabajo empírico sobre un número importante de datos orales espontáneos. Téngase en cuenta, que la obtención de un corpus oral de gran tamaño es una tarea muy costosa y que los corpus empleados están entre los más grandes disponibles para cualquier lengua.

10. BIBLIOGRAFÍA

[1] Alameda, J.R. & F. Cuetos (1995) *Diccionario de frecuencias de las unidades lingüísticas del castellano*. Servicio de publicaciones de la Universidad de Oviedo.

[2] Álvarez, C.J., M. Carreiras & M. De Vega (1992) "Estudio estadístico de la ortografía castellana: (1) la frecuencia silábica" *Cognitiva* 4, pp.75-105.

[3] Guirao, M. & A. Borzone de Manrique (1972) "Fonemas, sílabas y palabras del español de Buenos Aires" *Filologia*, XVI, pp.135-165.

[4] Navarro Tomás, T. (1946) "Escala de frecuencia de los fonemas españoles" *Estudios de fonología española*. Syracuse, pp.15-30

[5] Quilis, A. & M. A. Esgueva Martínez (1980)"Frecuencia de fonemas en el español hablado"*Lingüística Española Actual*, 2.

[6] Rojo, G. (1991) "Frecuencia de fonemas en el español actual" en Brea, M. & F. Fernández Rei (Coords.) *Homenaxe ó profesor Constantino García.* Universidade de Santiago. pp.451-457.

[7] Zipf, G. K. & J. M. Rogers (1939) "Phonemes and Variphones in four present-day romance Languages and Classical Latin from the viewpoint of dynamic Philology" Archives Néerlandaises de Phonétique Expérimentale" 15, pp. 111-147.

[8] Alameda, J.R. & F. Cuetos (1995) *Diccionario de frecuencias de las unidades lingüísticas del castellano*. Servicio de publicaciones de la Universidad de Oviedo.

[9] Marcos Marín, F. (1992) “El Corpus Oral de Referencia de la Lengua Española contemporánea” Project Report. Madrid. Publisher in ftp://ftp.lllf.uam.es/pub/corpus/oral.

LANGUAGE MODELING FOR VERBATIM TRANSLATION TASK

Maxim Khalilov and José A.R. Fonollosa

TALP Research Center, Universitat Pollitècnica de Catalunya, Barcelona

ABSTRACT

In this paper we present the first results towards finding the better TC-STAR[1] 2006 verbatim transcription system configuration by means of improving the quality of language model performance.

There is a present lack of research devoted to special techniques of verbatim translation, therefore we have made an attempt to improve translation accuracy by combining the Final Text Edition (FTE) system with supplementary verbatim corpus. Our work was focused on finding the best combination of the baseline (FTE) and verbatim language models for Spanish-English and English-Spanish language pairs. In order to improve the overall system performance standard n-gram based statistical machine translation (SMT) system was supplemented with a log linear combination of some additional feature functions and linguistically motivated word reordering technique.

In the final part of the study we report the results of the baseline system translation accuracy in comparison with the FTE-verbatim interpolated language model systems for various proportions of the language models linear combination.

1. INTRODUCTION

Statistical approach to the machine translation is comparatively recent area of scientific study, however, due to its promising potential it is growing relatively fast during last years.

A great advance in terms of translation system accuracy was done moving from the first systems based on the noisy channel approach model realizing word-to-word translation [1] to the phrase-based systems dealing with aligned bilingual corpora and implementing translation of the bilingual units [2,3]. However, this change entails correlation of the bilingual translation model, as well as additional language model, to the morphological nature of the corpus. Verbatim language peculiarities, such as non-grammatical or bad structured input, are not duly reflected in the regular translation systems. Hence, applying bilingual and language models properly calculated in accordance with semantically and grammatically corrected phrase alignment to the spontaneous speech input data, may cause translation errors. Because of the common point of view that SMT systems are robust to the non-grammatical input data, the majority of the up to date developed text-to-text translation systems address the problem of the pre-edited text translation, whereas systems of recognized speech translation like translation of the automatic speech recognizer output were set aside.

In this work we try to adapt the classical FTE system to the verbatim task by optimizing and adapting target language model which is an integrated part of the statistical translation system.

The paper is organized as follows. In Section 2 the n-gram translation model is briefly outlined, as well as additional models. Section 3 describes the verbatim task specific character and framework of the TC-STAR 2006 evaluation (www.elda.org/en/proj/tcstar-wp4). The performed experiments, results and discussions are described in the Section 4, while conclusions and future work are detailed in the Section 5.

2. STATISTICAL MACHINE TRANSLATION

As already mentioned, first SMT models were based on the noisy-channel paradigm [1], modeling translation of the target language sequence as argmax operation as described by the following equation:

$$\hat{e} = \arg\max_{e} P(e)\, P(f \mid e) \qquad (1)$$

where e refers to the sequence in target language and f to the sequence in the source language, and, consequently, translation model probability $P(f|e)$ and target language model probability $P(e)$.

2.1. N-gram translation model

The noisy channel approach was applied to the word-to-word alignment, where both models are estimated independently. The translation accuracy has been improved by switching to the phrase-based approach, in parallel, the n-gram based approach has appeared, operating with bilingual units extracted from the aligned bilingual corpus, referred to as tuples [4]. Tuples are extracted from a word-to-word aligned corpus according to certain constraints [5].

[1] Technology and Corpora for Speech to Speech Translation (http://www.tc-star.org)

The tuple n-gram translation model determines joint probability of the source and target language units as shown in the equation (2):

$$p(e,f) = \prod_{k=1}^{K} p(t_k \mid t_{k-N+1}, \ldots, t_{k-1}) \qquad (2)$$

where t_k refers to the k-*th* tuple of the given bilingual sentence pair segmented to *K* tuples, *N* refers to n-gram order.

Given a bilingual corpora, GIZA++ toolkit was used to generate word-to-word alignments in source-to-target and target-to-source directions [6], then tuples are extracted from these alignments.

Recently, the source channel model has been supplemented by the maximum entropy approach [7], implementing the posterior probability *p(e|f)* definition as a log-linear combination of the set of feature functions [8]. Speech-to speech translation task and finite state transducer model underlie this approach [4,9]. This technique allows to simplify feature models combinations under the translation hypothesis determination, as described below (3):

$$\hat{e}^I = \arg\max_{e^I} \left\{ \sum_{m=1}^{M} \lambda_m h_m(f_1^J, e_1^I) \right\} \qquad (3)$$

where the feature functions h_m refer to the system models as bilingual translation model, target language model, etc. and λ_m to weights corresponding to these models.

The weight coefficients normally are to be optimized to provide largest value of scoring function (BLEU score in the system under consideration [10]).

2.2. Additional feature models

Following the maximum entropy approach, the baseline system implements the log-linear combination of the translation model and the following feature models:

2.2.1. Target language model

The first additional feature is not obligatory for the systems based on n-gram approach in contrast to word-based systems is used to improve system translation quality. The model is computed according to the following equation:

$$P_{LM}(t_k) \approx \prod_{n=1}^{K} p(w_n \mid w_{n-N+1}, \ldots, w_{n-1}) \qquad (4)$$

where t_k refers to the partial translation hypothesis, w_n to the n-*th* word in this partially translated sentence.

2.2.2. Word penalty model

This feature was implemented to compensate system's striving for short output sentences. This phenomenon is due to the target language model. Technically, the penalization depends on the total number of words contained in the partial translation hypothesis, and can be found as follows:

$$P_{WP}(t_k) = \exp(number\ of\ words\ in\ t_k) \qquad (5)$$

2.2.3. Source-to-target lexicon model

This model uses word-to-word IBM model 1 probabilities [11] to estimate lexical weights of each tuple, as follows:

$$P_{IBM1}((e,f)_n) = \frac{1}{(I+1)^J} \prod_{j=1}^{J} \sum_{i=0}^{I} p\left(e_n^i \mid f_n^i\right) \qquad (6)$$

where f_n^j and e_n^i are the *j-th* and *i-th* words in the source and target parts of the tuple $(e,f)_n$, being *J* and *I* the corresponding total number words in each side of it. Giza++ word-to-word source-to-target alignment was used.

2.2.4. Target-to-source lexicon model

The same as the previous model but for opposite direction, backward lexicon model. Giza++ word-to-word target-to-source alignment was used.

2.2.5. POS target language model

The feature represents an n-gram model of the POS tags training corpus. English POS tagger TNT [12] and for Spanish Freeling [13] were used for monolingual corpora tagging [14].

2.3. Reordering

For Spanish to English a linguistically motivated word reordering technique was applied in order to decrease the number of errors caused by the difference in word order between two languages. The idea of the algorithm is to detect blocks of alignments which, if swapped, produce a monotone translation, then to classify the alignments blocks to groups and identify the consecutive blocks in the source corpora to swap them. Detailed description of the reordering procedure can be found in [15, 16]. The technique was applied to all the source corpora and the reordered training set was realigned to build the final translation system.

2.4. Decoding

MARIE decoder was used as a search engine for the translation system, the details can be found in [17]. The decoder implements a beam-search algorithm with

pruning capabilities. Versions starting with 1.2 support POS tags target language models, thereby all five additional feature models described above were taken into account.

2.5. Optimization

Optimization of the weight coefficients of the scoring function is based on the simplex optimization method [18]. Given development set and references, the log-linear combination of the weights is adjusted to maximize score function (see eq. 3) according to the highest BLEU score. Experiments on log-linear combination of BLEU and NIST scores are planned for the future work.

3. TC-STAR EVALUATION FRAMEWORK

Translations provided by the baseline system has been evaluated in the framework of the TC-STAR 2006 evaluation campaign. Our institution participated in the Spanish-English and Spanish-English language directions of FTE, verbatim and ASR (Automatic Speech Recognition output) shared tasks.

3.1. Task description

Verbatim transcription includes spontaneous speech phenomena (hesitations, false-starts, half-words, etc). FTE is a manually corrected text slightly different from the verbatim text. TC-STAR 2006 verbatim evaluation was performed in true case and with punctuation marks.

The difference between FTE and verbatim texts can be seen from the example below:

FTE**:** *I am starting to know what Frank Sinatra must have felt like*
Verbatim: *I'm I'm I'm starting to know what Frank Sinatra must have felt like*

3.2. Corpora

The data provided for shared tasks are European Parliament Plenary Sessions (EPPS) database for English-Spanish and Spanish-English language pairs.

Additionally, monolingual EPPS Verbatim Transcription corpora of the EPPS was used for English and Spanish specific verbatim language modeling. Development and test sets have 2 references for both languages. First 500 phrases of the provided official development corpora for both direction have been used for to maximize score function.

Basic corpora statistic can be found in the Tables 1 and 2. Training corpora are unique for all the tasks, development and test corpora vary.

EPPS	Spanish	English
Training set		
Sentences	1.3 M	1.3 M
Words	36.57 M	34.9 M
Vocabulary	153 K	107 K
Development set		
Sentences	500	500
Words	15 K	12 K
Vocabulary	2.5 K	2.3 K
Test set		
Sentences	699	1.155
Words	31 K	30 K
Vocabulary	3.9 K	4 K

Table 1. *EPPS corpora (M stands for millions, K stands for thousands).*

EPPS-Vbt	Spanish	English
Training set		
Sentences	70 K	73 K
Words	512 K	781 K
Vocabulary	20 K	17 K

Table 2. *EPPS-Verbatim corpora.*

4. EXPERIMENTS AND RESULTS

The mixing procedure was implemented with *n-gram* tool from the SRI Language Modeling toolkit [19], allowing to read the second model for interpolation purposes and adjust weight coefficients when interpolating with the main model. We considered six alternative configurations for each target language. They include various weights of the minor verbatim model from 0 (FTE model only) to 1 (verbatim model only) with step 0.2.

The BLEU scores obtained on the development corpus as a result of the simplex optimization procedure (refer to Dev) were investigated. Tables 3 and 4 represent BLEU score of Spanish-English and English-Spanish translation experiments under the baseline system and systems using interpolated language models. Graphical representation of the obtained results can be found in the Figures 1 and 2.

Maximizing this value, the best performance point is provided by the system corresponding to 0.1 FTE – 0.9 VBT configuration for Spanish to English translation task and to 0.9 FTE – 0.1 VBT for English to Spanish task. It allows gaining 0.47 BLEU points on the Test for Spanish to English translation and 0.27 BLEU points for English to Spanish task comparing to the baseline system.

Moreover, we investigated the behavior of the BLEU score obtained on the Test corpus considering the same system configurations (Tables 5 and 6) and optimized models weights (refer to Test). Results also can be found in the Figures 3 and 4. It can be seen that despite absolute maxima achieved on the Test and Dev vary, the shapes of the curves are similar.

In the framework reported in the paper, we did not conduct any experiments focused on the translation model training based on the interpolated models, which can be an important point of investigation in future.

5. CONCLUSIONS AND FUTURE WORK

In this paper we presented a brief study of the possible ways of verbatim texts translation improvement by means of applying modified language model to the translation system. The baseline translation system, represents translation system used for TC-STAR 2006 evaluation and implements some new features, namely additional tagged target language model and lexically motivated reordering technique.

It is seen from the reported results that the described technique works for both of the considered languages and corresponding translation tasks, for English to Spanish and Spanish to English translations. Despite the slight difference on the translation accuracy was observed, the smoothed shapes of the translation BLEU score curves for Test and Dev corpora are similar.

Additional verbatim language model can be also included as a feature to the log-linear combination of the set of functions with a corresponding weight, as shown in (3).

Further work includes applying of new additional feature optimization, namely the algorithm adjusting weight coefficients so that to maximize a log-linear combination of NIST and BLEU over the development set and to apply the described technique in combination with phrase-based approach [20] which could imply further BLEU improvement due to different algorithm of taking advantage of each feature including the target language model.

6. ACKNOWLEDGMENTS

This work has been partially funded by the European Union under the integrated project TC-STAR – Technology and Corpora for Speech to Speech Translation – (IST-2002-FP6-506738, http://www.tc-star.org) and by the Spanish Government under grant AP20042835 (FPU grant).

7. REFERENCES

[1] P. Brown, S. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. "The mathematics of statistical machine translation: parameters estimation". Computational Linguistics, 19(2):263-311.

[2] P. Koehn, F. J. Och, and D. Marcu. 2003. "Statistical phrase-based translation". Proc. of the 2003 Meeting of the North American chapter of the ACL, Edmonton, Alberta.

[3] R. Zens, F. J. Och, and H. Ney, "Phrase-based statistical machine translation" in KI – 2002: Advanced in artificial intelligence, M. Jarke, J. Koehler, and G. Lakemeyer, Eds. Springer Verlag, September 2002, vol. LNAI 2479, pp. 18-32.

[4] A. de Gispert, and J. B. Mariño. 2002. "Using X-grams for speech-to-speech translation". Proc. Of the 7th Int. Conf. on Spoken Language Processing.

[5] J. M. Crego, J. B. Mariño, A. de Gispert. 2004. "Finite-state-based and phrase-based statistical machine translation". Proc. Of the 8th Int. Conf. on Spoken Language Processing, :37-40.

[6] F. J. Och and H. Ney. 2000, "Improved statistical alignment models". Proc. of the 38th Ann. Meeting of the ACL, Hong Kong, China, October.

[7] A. Berger, S. Della Pietra, and V. Della Pietra, "A maximum entropy approach to natural language processing", Computational Linguistics, vol. 22, no.1, pp. 39-72, March 1996

[8] F. J. Och and H. Ney. 2000, "Improved statistical alignment models". Proc. of the 38th Ann. Meeting of the ACL, Hong Kong, China, October.

[9] E. Vidal, "Finite-state speech-to-speech translation", Proc. of 1997 IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 111-114, 1997.

[10] K. Papineni, S. Roukos, T. Ward, and Wei Jing Zhu. 2002. "Bleu: a method for automatic evaluation of machine translation". Proc. of the 40th Ann. Conf. of the ACL, Philadelphia, PA, July.

[11] F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Zhen Jin, and D. Radev, "A smorgasbord of features for statistical machine translation", Proc. of the Human Language Technology Conference, HLT-NAACL' 2004, pp. 161-168, May, 2004.

[12] T. Brants. 2000. "Tnt – a statistical part-of-speech tagger" Proceedings of the Sixth Applied Natural Language Processing.

[13] X. Carreras, I. Chao, L. Padró, and M. Padró. 2004. "Freeling: An open-source suite of language analyzers". 4th Int. Conf. on Language Resources and Evaluation, LREC'04.

[14] M. Popović and H. Ney. "POS-based Word Reorderings for Statistical Machine Translation". 5th International Conference on Language Resources and Evaluation (LREC), pages - to appear, Genova, Italy, May 2006.

[15] M. R. Costa-jussà, J. M. Crego, A. de Gispert, P. Lambert., M. Khalilov, R. E. Branchs, J. B. Mariño, J. A. R. Fonollosa. 2006. "TALP Phrase-based statistical translation system for European language pairs". NAACL 2006 Workshop on Statistical Machine Translation. Pages – to appear. 2006

[16] M. R. Costa-jussà and J. A. R. Fonollosa. 2006. "Statistical Machine Reordering". Proceedings of EMNLP 2006. pp. 70-76.

[17] J. M. Crego, J. B. Mariño, and A. de Gispert. 2005. "An Ngram-based statistical machine translation decoder". Proc. of the 9th Int. Conf. on Spoken Language Processing, ICSLP'05.

[18] J.A. Nelder and R. Mead. 1965. "A simplex method for function minimization". The Computer Journal, 7:308-313

[19] A. Sholcke. 2002. "SRILM: an extensible language modelling toolkit". Proc. of the Int. Conf. on Spoken Language Processing: 901-904, Denver, CO, September.

[20] R. Zens, F. J. Och, and H. Ney, "Phrase-based statistical machine translation" in KI – 2002: Advanced in artificial intelligence, M. Jarke, J. Koehler, and G. Lakemeyer, Eds. Springer Verlag, September 2002, vol. LNAI 2479, pp. 18-32.

FTE	Vbt	BLEU Dev
Baseline (1.0 FTE-0.0 Vbt)		50.53
0.9	0.1	50.66
0.8	0.2	50.84
0.5	0.5	50.76
0.2	0.8	50.92
0.1	0.9	50.99
0.0	1.0	50.97

Table 3. *Results of Spa-to-Eng translation (Dev).*

FTE	Vbt	BLEU Dev
Baseline (1.0 FTE-0.0 Vbt)		44.05
0.9	0.1	44.39
0.8	0.2	44.01
0.5	0.5	43.90
0.2	0.8	43.52
0.1	0.9	43.17
0.0	1.0	42.03

Table 4. *Results of Eng-to-Spa translation (Dev).*

FTE	Vbt	BLEU Test
Baseline (1.0 FTE-0.0 Vbt)		52.24
0.9	0.1	52.49
0.8	0.2	52.83
0.5	0.5	52.81
0.2	0.8	52.87
0.1	0.9	52.72
0.0	1.0	52.63

Table 5. *Results of Spa-to-Eng translation (Test).*

FTE	Vbt	BLEU Test
Baseline (1.0 FTE-0.0 Vbt)		44.19
0.9	0.1	44.46
0.8	0.2	44.48
0.5	0.5	44.75
0.2	0.8	44.06
0.1	0.9	43.62
0.0	1.0	42.71

Table 6. *Results of Eng-to-Spa translation (Test).*

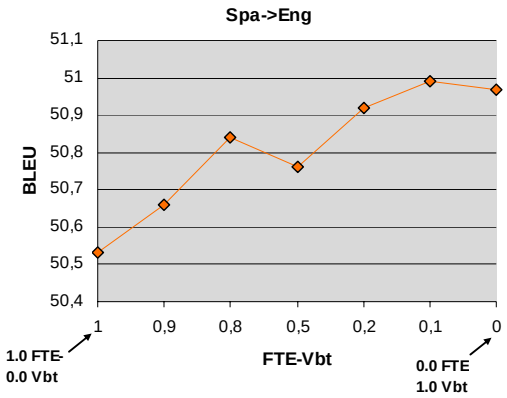


Figure 1. *Results of Spa to Eng translation (Dev).*

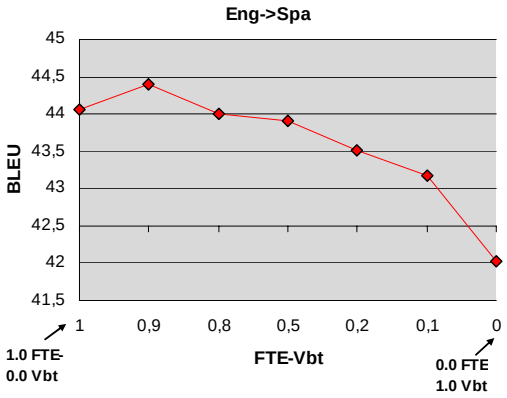


Figure 2. *Results of Eng to Spa translation (Dev).*

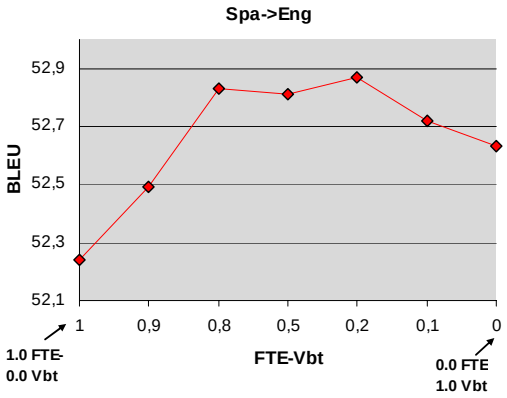


Figure 3. *Results of Spa to Eng translation (Test).*

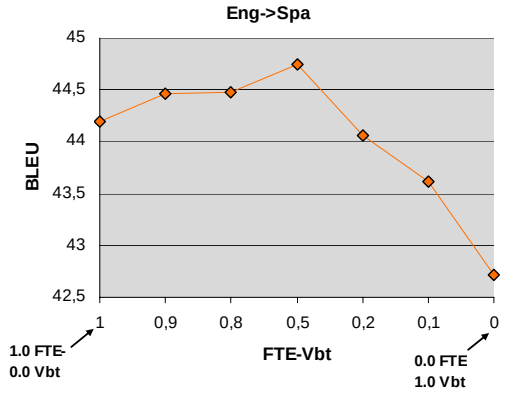


Figure 4. *Results of Eng to Spa translation (Test).*

MODELOS DE LENGUAJE BASADOS EN CATEGORÍAS SEMÁNTICAS EN UN SISTEMA DE DIÁLOGO DE HABLA ESPONTÁNEA EN CASTELLANO

*Raquel Justo, M. Inés Torres**

Universidad del País Vasco
Dpto. de Electricidad y Electrónica
48080 Leioa, España
webjublr@lg.ehu.es, manes@we.lc.ehu.es

Lluis Hurtado

Universidad Politécnica de Valencia
Dpto. de Sistemas Informáticos y Computación
46071 Valencia, España
lhurtado@dsic.upv.es

RESUMEN

El principal objetivo de este trabajo es comprobar si un modelo de lenguaje basado en categorías semánticas puede mejorar el rendimiento de un sistema de Reconocimiento Automático del Habla integrado en un sistema de diálogo de habla espontánea en castellano. Los experimentos se han llevado a cabo generando diferentes modelos de lenguaje basados en categorías semánticas. Se ha tratado de que estos modelos incorporen conocimiento relativo a las relaciones entre palabras además de la información asociada a las categorías. Para ello se han utilizado diferentes aproximaciones y los mejores resultados se han obtenido para un modelo híbrido que es una combinación lineal de un modelo basado en palabras y un modelo basado en categorías.

1. INTRODUCCIÓN

Los sistemas de diálogo se encuentran entre las aplicaciones de mayor interés en el ámbito de las tecnologías del habla. La finalidad de estos sistemas es interaccionar oralmente con los usuarios a fin de proporcionar determinados servicios como acceso a la información o control de funcionalidad de una máquina. Diversos ejemplos de este tipo de sistemas y las descripciones de los mismos pueden encontrarse en [1, 2, 3]

En un sistema de diálogo intervienen principalmente los módulos que se enumeran a continuación: Módulo de Reconocimiento Automático del Habla (RAH), Módulo de comprensión, Módulo de gestión de diálogo, Módulo de generación de respuestas y Módulo de Síntesis de Voz. Estos módulos cooperan para generar las intervenciones del sistema con el usuario [4].

El módulo de reconocimiento captura la señal acústica pronunciada por el usuario y proporciona la transcripción ortográfica al resto de los módulos del sistema de diálogo que se encargarán de interpretarla y generar la repuesta adecuada. De esta forma, un módulo de reconocimiento con bajo rendimiento, incapacita la evaluación global del sistema de diálogo, puesto que no es posible que el sistema de diálogo genere una respuesta adecuada a partir de una frase mal reconocida. Es por ello que este trabajo se centra en el módulo de RAH y más concretamente en el Modelo de Lenguaje (ML) utilizado en él. El módulo de comprensión, por su parte, recoge la salida del módulo de RAH y extrae el significado de las palabras reconocidas expresándolo en términos de un lenguaje semántico especificado para la tarea [5]. Si la salida del módulo de RAH proporciona una frase reconocida que aporte información semántica además de la transcripción ortográfica, el proceso de reconocimiento y comprensión podría integrarse en un solo paso.

El lenguaje natural entre personas se basa en una gran diversidad de conocimiento a priori que permite realizar presuposiciones y simplificar el lenguaje que se utiliza. Este hecho dificulta en gran medida el trabajo de los sistemas que tratan con el habla natural y carecen de esa información adicional, como por ejemplo el módulo de RAH. Por este motivo la utilización de un ML apropiado que se adapte a los requisitos de la tarea y que sea capaz de capturar la estructura de las frases pronunciadas por los locutores, cobra una gran importancia. Actualmente, se utilizan ML estadísticos en el proceso de reconocimiento. Para estimar de forma robusta los parámetros de este tipo de modelos se necesitan grandes cantidades de datos de entrenamiento. Sin embargo, en el caso de los sistemas de diálogo no existe mucho material de entrenamiento disponible, debido en gran medida a la dificultad de adquisición del mismo. Una de las vías para compensar la escasez o dispersión de los datos de entrenamiento se basa en la categorización del vocabulario de la aplicación en un conjunto más pequeño de clases [6].

En este trabajo la categorización se ha realizado utilizando un criterio semántico. De esta forma, se podría recuperar la información semántica asociada a las categorías en el proceso de reconocimiento, integrando en un solo paso la obtención de la transcripción ortográfica de la frase y la interpretación semántica de la misma. Este tipo de categorías semánticas están estrechamente relacionadas con la tarea que nos ocupa. En este caso la tarea consiste en consultas telefónicas acerca de horarios y precios

*Este trabajo ha sido parcialmente subvencionado por el proyecto CICYT TIN2005-08660-C04-03 y la Universidad del País Vasco 9/UPV 00224.310-15900/2004

de trenes de largo recorrido, pronunciadas en castellano por usuarios potenciales del sistema.

El principal objetivo de este trabajo es estudiar la influencia de esa categorización semántica en un ML que se encuentra integrado en el módulo de RAH de un sistema de diálogo. A partir de esta categorización se generan diferentes ML-s basados en las categorías obtenidas. Estos modelos se integran en un sistema de RAH y se evaluan en términos de WER.

En la sección 2 se describen los diferentes modelos basados en categorías que se han utilizado en este trabajo. La sección 3 detalla las características de la tarea y el corpus. En la sección 4 se describen los experimentos realizados y se presenta la evaluación de las propuestas descritas en la secciones previas. Finalmente, en la sección 5 se presentan las conclusiones principales extraídas de los experimentos, así como las propuestas para trabajos futuros

2. MODELOS DE LENGUAJE BASADOS EN CATEGORÍAS SEMÁNTICAS

Los ML basados en categorías recogen la información asociada a las relaciones entre grupos de palabras "olvidando" las relaciones entre palabras particulares. Esta pérdida de información deriva habitualmente en una reducción del rendimiento del sistema de RAH. Por este motivo, existen diversos trabajos en los que se intenta generar modelos que combinen ambas fuentes de conocimiento, obteniendo así un modelo enriquecido [7]. En este trabajo se ha realizado esta tarea mediante tres métodos diferentes que se detallan más abajo.

2.1. Modelos basados en k-gramas de categorías

En primer lugar se han generado ML basados en k-gramas de categorías. Este tipo de modelos se genera en dos pasos. En primer lugar, a partir de un corpus categorizado, se aprende un modelo de lenguaje basado exclusivamente en las relaciones entre categorías. Para ello se utilizan gramáticas k-Explorables en sentido estricto (k-EEE) [8] que son la aproximación gramatical de los bien conocidos modelos de k-gramas. Las categorías utilizadas están constituídas por palabras o secuencias de palabras que cumplen una función semántica concreta. Así, de la misma manera que se aprenden las relaciones entre categorías, se aprenden también modelos de lenguaje basados en palabras para cada una de las categorías definidas, de forma que no se pierde toda la información asociada a las relaciones entre palabras.

Los ML k-EEE vienen representados por Autómatas de Estados Finitos Estocásticos (AEFE). La integración, en un sistema de RAH, de este tipo de ML k-EEE basados en categorías se realiza en dos niveles: por un lado se considera el autómata que modela las relaciones existentes entre las categorías y por otro los autómatas correspondientes a cada una de las categorías. En el primer nivel, el más genérico, se transita mediante las etiquetas de las categorías de un estado a otro del autómata. Cada estado de este autómata se corresponde con una categoría específica, para la cual existe un autómata de palabras asociado. En el segundo nivel, el más específico, se transita por estos autómatas de palabras. En el proceso de reconocimiento se integran los autómatas de ambos niveles en un único autómata que se va construyendo al vuelo, en función de los caminos que se mantienen activos cuando se realiza la búsqueda de la hipótesis más probable.

En este trabajo se han generado dos ML de estas características. En el primero se consideran bigramas de categorías y bigramas de palabras dentro de las categorías (bi-bi). En el segundo se consideran trigramas de categorías y bigramas de palabras dentro de las categorías (tri-bi).

2.2. Modelo híbrido

Por otra parte se ha considerado un modelo híbrido. Este modelo trata de aprovechar la información relativa a las relaciones entre palabras particulares y la asociada a las relaciones entre categorías, mediante una combinación lineal de ambos modelos [7]. Es decir, se trata de una combinación lineal de un modelo de categorías como el descrito en el apartado previo y un modelo de k-gramas de palabras convencional. Se han generado dos ML híbridos:

1. Una combinación lineal de un modelo de trigramas de palabras con un modelo de bigramas de categorías (hib-bi).
2. Una combinación lineal de un modelo de trigramas de palabras con un modelo de trigramas de categorías (hib-tri).

En este caso, los dos modelos que combina el modelo híbrido se toman en cuenta de forma independiente, otorgándole a cada uno el peso que viene dado por el parámetro de la combinación lineal.

2.3. Modelos basados en MGGI

La metodología Morphic Generator Grammatical Inference (MGGI) es una metodología de inferencia gramatical que se ha utilizado con éxito en el campo del modelado de lenguaje [9]. Básicamente consiste en lo siguiente:

- Las muestras de entrenamiento son etiquetadas utilizando una función de etiquetado g.
- Se aprende un lenguaje regular 2-Explorable en sentido estricto utilizando las muestras etiquetadas. El resultado se puede representar como un autómata finito determinista. El alfabeto de este modelo es el alfabeto etiquetado, no el original.
- Aplicando un homomorfismo se procede a desetiquetar el modelo aprendido en el paso anterior. En

nuestro caso el proceso de desetiquetado se consiguió definiendo cada palabra etiquetada como una categoría cuyo único miembro era la propia palabra desetiquetada.

El resultado final es un autómata no determinista que admite cadenas sobre el alfabeto original. Los modelos MGGI son modelos más grandes que los 2-EEE para las mismas muestras de entrenamiento y el lenguaje que aceptan es un subconjunto del lenguaje aceptado por los 2-EEE. En realidad, el lenguaje inferido utilizando la metodología MGGI fluctúa entre el lenguaje 2-Explorable en sentido estricto inferido a partir de las muestras de entrenamiento sin etiquetar y el aceptor de prefijos aprendido con esas mismas muestras, dependiendo de la función de etiquetado g utilizada. En el caso que nos ocupa la función de etiquetado consiste en añadir a cada palabra la etiqueta de la categoría semántica a la que pertenecía en esa frase. Una variante de la metodología MGGI es la denominada tri-MGGI. La única diferencia respecto a la original consiste en utilizar modelos 3-Explorables en sentido estricto en lugar de los 2-Explorables de la metodología original. En este trabajo se han generado dos modelos basados en esta técnica: mggi y trimggi (utilizando la técnica tri-MGGI)

3. CARACTERÍSTICAS DE LA TAREA Y CORPUS

En este trabajo se ha utilizado una base de datos hombre-máquina en castellano. Este corpus fue adquirido dentro del marco del proyecto DIHANA [10] y consta de diferentes intervenciones en las que los usuarios solicitan información sobre horarios y precios de trenes de largo recorrido mediante el teléfono. Las características de este corpus se detallan en la tabla 1

DIHANA	
diálogos	900
locutores	225
nº frases total	6229
nº frases entrenamiento	4888
nº frases test	1341
vocabulario	718
OOV (palabras no vistas)	95
nº palabras	47219

Tabla 1. Características del corpus DIHANA

4. RESULTADOS EXPERIMENTALES

En este trabajo se han llevado a cabo un conjunto de experimentos preliminares en los que se han generado ML que tratan de aprovechar la información asociada a las relaciones entre palabras y entre categorías. Para generar un ML basado en categorías, en primer lugar se requiere un conjunto de categorías seleccionado a partir de un criterio de categorización previamente estipulado. En este caso, las categorías se han obtenido a partir de un criterio semántico y coinciden con los conceptos semánticos especificados para la tarea que utiliza el módulo de comprensión. Se han utilizado, en concreto, 31 categorías en las que se han clasificado todas las palabras del vocabulario de la tarea. A continuación se enumeran algunas de las categorías seleccionadas:

- **afirmación:** “sí”, “vale”, “está bien”, . . .
- **negación:** “no”, “nada”, “nada más”, . . .
- **hora_llegada:** “a qué hora llega”, . . .
- **hora_salida:** “tengo que salir”, “a qué hora sale”, “de salida”, . . .
- **ciudad_origen:** “de madrid”, “desde bilbao”, . . .
- **ciudad_destino:** “a girona”, “a valladolid”, . . .

Los experimentos se han llevado a cabo utilizando el corpus descrito en el punto anterior. Los modelos generados se representan mediante autómatas que han sido integrados en un sistema de RAH. Este sistema consiste finalmente en un único autómata estocástico en el que se integran los modelos acústicos, el modelo léxico y los modelos de lenguaje basados en palabras y categorías. Mediante el algoritmo de Viterbi se realiza una búsqueda en haz de la hipótesis más probable a través de este autómata general.

En la tabla 2 se muestran los resultados de WER obtenidos para los diferentes ML generados. En primer lugar se han considerado dos ML basados en palabras como referencia. Por otra parte se han generado dos ML basados en k-gramas de categorías (bi-bi y tri-bi) y otros dos ML híbridos. Estos modelos híbridos se generan como una combinación lineal de un modelo basado en trigramas de palabras y un modelo basado en bigramas (hib-bi) o trigramas (hib-tri) de categorías. Finalmente se han generado ML basados en MGGI (mggi y trimggi)

WER (%)		
pal.	**bigramas**	20.34
	trigramas	19,05
categorías	**bi-bi**	20.35
	tri-bi	20.44
	hib-bi	18.78
	hib-tri	18.77
	mggi	19.01
	trimggi	18.96

Tabla 2. Resultados de WER para los diferentes modelos basados en palabras y categorías. El conjunto de categorías utilizado es el mismo en todos los casos y los modelos que se han utilizado son los descritos en el apartado 2

La tabla 2 muestra que los mejores resultados se obtienen para el modelo híbrido. Los valores de WER obtenidos tanto para el hib-bi como para el hib-tri son mejores que el resto de los modelos basados en categorías y además mejoran el rendimiento de los ML basados en palabras. Los modelos basados en MGGI proporcionan resultados de WER ligeramente mejores que los modelos basados en palabras y se encuentran en un punto intermedio entre éstos últimos modelos y los modelos híbridos. Finalmente, los modelos basados en k-gramas de categorías (bi-bi y tri-bi) ofrecen resultados peores que el resto de los modelos basados en categorías, y muy similares a los modelos basados en palabras. En cambio, según se aprecia en trabajos previos [11], cuando se utiliza este tipo de modelos considerando unigramas de palabras dentro de las categorías, los valores de WER son considerablemente mayores que en los ML basados en palabras. Es decir, el hecho de considerar bigramas de palabras dentro de las categorías produce que los resultados de WER mejoren y sean muy próximos a los obtenidos mediante ML basados en palabras.

5. CONCLUSIONES Y TRABAJO FUTURO

Los experimentos realizados muestran que incluyendo información semántica en los modelos se puede mejorar ligeramente el rendimiento del reconocedor.

Por otra parte, mediante los experimentos realizados se ha tratado de evaluar diferentes ML que integran información relativa a dos fuentes de conocimiento. Es decir, información relativa a las relaciones entre palabras y a las relaciones entre categorías. Los experimentos muestran que la mejor manera de combinar ambas fuentes de conocimiento es la combinación lineal de las mismas, es decir, la aproximación proporcionada por el modelo híbrido. Sin embargo, dado que las diferencias entre los resultados obtenidos no son muy significativas habría que realizar una experimentación más extensa para estudiar la potencia de los modelos propuestos. Se podría ampliar la experimentación utilizando diferentes combinaciones de k-gramas en los modelos basados en categorías, e incluso experimentar con valores de k mayores que 3.

Por otra parte, la utilización de ML basados en categorías semánticas nos permite extraer la categoría semántica de las palabras reconocidas mediante el módulo de RAH. De esta forma los procesos de reconocimiento y comprensión podrían quedar integrados en un solo paso. En un trabajo futuro se quiere obtener tasas de error de las frases reconocidas en lo que a la categoría semántica se refiere. Por otra parte sería interesante comparar la información semántica proporcionada por el reconocedor con la obtenida mediante un módulo de comprensión cuando el proceso de reconocimiento y comprensión se encuentran desacoplados.

6. BIBLIOGRAFÍA

[1] V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. Hazen, y L. Hetherington, "Jupiter: A telephone-based conversational interface for weather information," in *IEEE Trans. on Speech and Audio Proc.*, 2000, pp. 8(1):85–96.

[2] L. Lamel, S. Rosset, J.L. Gauvain, S. Bennacef, M. Garnier-Rizet, y B. Prouts, "The LIMSI ARISE System," vol. 31, no. 4, pp. 339–354, Aug 2000.

[3] S. Seneff y J. Polifroni, "Dialogue management in the mercury flight reservation system," in *Proc. ANLP-NAACL 2000 Satellite Workshop*, 2000, pp. 1–6.

[4] E. Gianchin y S. McGlashan, "Corpus-Based Methods in Speech Processing," *Kluwer Academic, ch. Spoken Language Dyalogue Systems*, pp. 67–117, 1997.

[5] D. Vilar, M. J. Castro, y E. Sanchis, "Connectionist classification and specific stochastic models in the understanding process of a dialogue system," in *03*, Geneva (Switzerland), Sept. 2003, vol. 1, pp. 645–648.

[6] T. R. Niesler y P. C. Woodland, "A variable-length category-based n-gram language model," in *IEEE ICASSP-96*, Atlanta, GA, 1996, IEEE, vol. I, pp. 164–167.

[7] J.M. Benedí y J.A. Sánchez, "Estimation of stochastic context-free grammars and their use as language models," *Computer Speech and Language*, vol. 19, no. 3, pp. 249–274, 2005.

[8] P. Garcia y E. Vidal, "Inference of k-testable languages in the strict sense and application to syntactic pattern recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 9, pp. 920–925, 1990.

[9] E. Segarra y L. Hurtado, "Construction of Language Models using Morfic Generator Grammatical Inference MGGI Methodology," in *97*, Rhodes (Greece), Sept. 1997, pp. 2695–2698.

[10] DIHANA project, "Dialogue System for Information Access Using Spontaneous Speech in Different Environments," Comisión Interministerial de Ciencia y Tecnología TIC2002-04103-C03-03, 2005, http://www.dihana.upv.es.

[11] R. Justo, I. Torres, y J.M. Benedí, "Category-based language models in a spanish sponken dialogue system," in *XXII Congreso de la Sociedad Española de Procesamiento del Lenguaje Natural*, 2006, pp. 19–24.

UNA EVALUACIÓN EXHAUSTIVA DE SISHITRA, UN PARADIGMA HÍBRIDO EN TRADUCCIÓN AUTOMÁTICA

Jesús González[1], *A. Giménez*[1], *Jorge González*[1], *A. L. Lagarda*[1], *J. R. Navarro*[2], *L. Eliodoro*[2], *V. Félix*[2], *P. Peris*[1], *F. Casacuberta*[1]

[1]Departamento de Sistemas Informáticos y Computación
Universitad Politécnica de Valencia
{jegonzalez, agimenez, jgonzalez, alagarda, pperis, fcn}@dsic.upv.es

[2]Instituto Tecnológico de Informática
Universitad Politécnica de Valencia
{jonacer, leliodoro, victorf}@iti.upv.es

RESUMEN

SisHiTra representa un marco de computación en el que se combinan técnicas deductivas e inductivas para el desarrollo de sistemas de traducción automática de dominio abierto entre idiomas parejos. El uso de máquinas de estados finitos para modelar todas las fuentes de conocimiento ha permitido el diseño de un software muy eficiente. El rendimiento de esta aproximación se demuestra sobre el par de lenguas español y catalán, consiguiendo un compromiso muy interesante entre precisión y eficiencia computacional.

1. INTRODUCCIÓN

El objetivo de la *Traducción Automática* (TA) es desarrollar aplicaciones que permitan la traducción de textos sin intervención humana. Sin embargo, las tecnologías actuales de traducción no son capaces de cubrir las demandas de una traducción de alta calidad entre cualquier par de idiomas.

Las aproximaciones basadas en el conocimiento, en las que el conocimiento lingüístico experto se formaliza de un modo computable han tenido cierto éxito al abordar una traducción sin ningún tipo de restricciones. Sin embargo, las técnicas basadas en corpus, en las que los modelos estadísticos se infieren automáticamente a partir de ejemplos de textos, han alcanzado resultados muy competitivos en tareas semánticamente restringidas.

Existen modos diferentes de representar el conocimiento lingüístico, y, en particular, los transductores de estados finitos [1, 2, 3] han demostrado ser muy eficientes tanto en la implementación de sistemas deductivos como inductivos. Asimismo, las técnicas basadas en modelos de estados finitos han permitido el desarrollo de herramientas útiles para el procesamiento del lenguaje natural[4, 5, 6, 7, 8].

SisHiTra representa un marco de computación en el que, a través de un conjunto de modelos de estados finitos, se combinan técnicas deductivas e inductivas para el desarrollo de sistemas de TA de dominio abierto entre idiomas parejos [9].

El principal objetivo de SisHiTra es el de proporcionar traducciones de alta calidad con propósitos diseminativos. Por supuesto, es el objetivo de cualquier sistema de TA; sin embargo, en nuestro caso, es un aspecto especialmente importante que se ha tenido en cuenta en el diseño de cada etapa. Por lo tanto, consideramos que una traducción óptima es aquella que no parece el resultado de una traducción, sino que parece, más bien, fruto de un proceso de generación directa en el idioma destino. Esto no suele ser un problema para un traductor profesional, aunque es crucial en un sistema automático. Por ejemplo, un ser humano resuelve ambigüedades semánticas con cierta facilidad, mientras que en TA suelen representar un serio problema. Como consecuencia, la evaluación de la calidad de SisHiTra se hace en términos de la distancia entre las hipótesis de traducción y sus respectivas referencias, cuya optimalidad lingüística ha sido determinada a priori por una serie de expertos.

Por otro lado, el uso de máquinas de estados finitos para modelar todas las fuentes de conocimiento ha permitido un diseño muy eficiente del sistema. Así, la evaluación de la productividad de SisHiTra se ha realizado a través del tiempo de respuesta que proporciona al usuario.

Por último, el rendimiento de este paradigma híbrido se demuestra sobre el par de lenguas español y catalán, consiguiendo un compromiso muy interesante entre precisión y eficiencia computacional.

Este trabajo ha sido parcialmente respaldado por la "Agència Valenciana de Ciència y Tecnologia" bajo la subvención GRUPOS03/031 y por el proyecto español ITEFTE (TIC2003-08681-C02-02).

2. MARCO ESTADÍSTICO

Sea $\mathcal{W} = \{w_1, w_2, \ldots, w_N\}$ el vocabulario de entrada y sea $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ el de salida. El problema de la TA puede formularse estadísticamente como sigue: dada una frase $\mathbf{s} = \mathbf{s}_1, \ldots, \mathbf{s}_L \in W^*$, se debe buscar la secuencia de elementos de salida $\hat{\mathbf{t}} \in C^*$ que maximice la probabilidad a posteriori[1]:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} \Pr(\mathbf{t}|\mathbf{s}) \; . \tag{1}$$

Empleando la regla de Bayes y dado que el proceso de maximización es independiente de la frase de entrada **s**, la Ecuación (1) se puede reescribir como:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} \Pr(\mathbf{t}) \Pr(\mathbf{s}|\mathbf{t}) \; . \tag{2}$$

En esta ecuación, las probabilidades contextuales (o modelo de lenguaje), $\Pr(\mathbf{t})$, representan todas las secuencias posibles de elementos de la salida, mientras que las probabilidades de emisión (o modelo de transferencia), $\Pr(\mathbf{s}|\mathbf{t})$, establecen la relación entre los vocabularios de entrada y salida.

Esta ecuación puede resolverse adecuadamente bajo ciertas condiciones aceptables: en primer lugar, se asume una traducción monótona elemento a elemento, proporcionando hipótesis de la misma longitud que la frase de entrada. Este hecho implica la utilización de un modelo de fertilidad 1. Por lo tanto, se pueden incorporar ciertas asunciones propias de los modelos de Markov para simplificar el problema.

Por una parte, se asume que las probabilidades contextuales para un elemento determinado dependen exclusivamente de los *n* elementos inmediatamente anteriores. Por otra parte, podemos asumir que las probabilidades de emisión únicamente dependen del símbolo de salida correspondiente. Así, desde un punto de vista generativo, las hipótesis se producirían monótonamente de izquierda a derecha, obteniéndose exactamente un símbolo de salida por cada elemento de la entrada.

Para los modelos de Markov de primer orden (bigramas), el problema se reduce a resolver la siguiente ecuación:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}_1,\ldots,\mathbf{t}_L} \left(\prod_{1\ldots L} P(\mathbf{t}_i|\mathbf{t}_{i-1}) P(\mathbf{s}_i|\mathbf{t}_i) \right) \tag{3}$$

Los parámetros de esta ecuación se pueden representar mediante un *Modelo Oculto de Markov* (en inglés *Hidden Markov Model*) en el que los estados y los símbolos de salida están unívocamente asociados. Las probabilidades contextuales, $P(\mathbf{t}_i|\mathbf{t}_{i-1})$, son las probabilidades de transición entre estados y las probabilidades del modelo de transferencia, $P(\mathbf{s}_i|\mathbf{t}_i)$, se pueden ver como distribuciones de probabilidad entrada-salida. El problema de maximización que se describe en la Ecuación (3) se resuelve por medio del algoritmo de Viterbi[10].

[1]Por simplicidad, $\Pr(X = x)$ y $\Pr(X = x \mid Y = y)$ se denotan como $\Pr(x)$ y $\Pr(x \mid y)$

2.1. Desambiguación probabilística

El objetivo de la desambiguación sintáctica es el de decidir la categoría léxica a la que pertenece un elemento dentro de un contexto determinado. Esta categorización no sólo es útil para reducir la ambigüedad sobre la traducción, sino también para otros propósitos lingüísticos, como, por ejemplo, la detección de sintagmas, indispensable para llevar a cabo las concordancias pertinentes.

La desambiguación sintáctica se realiza a través de la aproximación estadística descrita en la Ecuación 3, en la que el vocabulario de salida se refiere al conjunto de categorías léxicas.

Además de la desambiguación sintáctica, la TA precisa desambiguar semánticamente los elementos de la entrada antes de convertirlos a la lengua destino. Los métodos de desambiguación semántica tratan de determinar el significado implícito de un elemento dentro de un contexto. De nuevo, SisHiTra emplea modelos estadísticos para esta tarea.

Del mismo modo que en el caso sintáctico, la desambiguación semántica se lleva a cabo mediante la aproximación estadística descrita en la Ecuación 3.

Los modelos estadísticos están ganando popularidad por diversos motivos, especialmente por su facilidad de generación frente a los métodos deductivos. Las técnicas probabilísticas aprenden automáticamente a partir de corpus, evitándose el proceso de producir conocimiento lingüístico, con el ahorro temporal y económico que ello conlleva. Sin embargo, la obtención de los corpus para el entrenamiento de los modelos tampoco es trivial.

Por una parte, los modelos empleados en la desambiguación sintáctica necesitan un corpus monolingüe y segmentado. A cada segmento se le asigna una categoría. Por otra parte, los modelos para la desambiguación semántica en SisHiTra requieren corpus paralelos, es decir, corpus en los que el texto se encuentre emparejado biunívocamente entre ambas lenguas. A tal fin, se recolectaron diversos corpus a partir de algunas publicaciones electrónicas bilingües (periódicos, textos oficiales, etc.), y se paralelizaron mediante diversos algoritmos de alineamiento.

3. ARQUITECTURA DEL SISTEMA

SisHiTra es un paradigma de computación para el desarrollo de sistemas de traducción automática de dominio abierto entre idiomas parejos. Una versión previa de la arquitectura de SisHiTra se puede encontrar en [11].

Se han utilizado una serie de metodologías innovadoras, basadas en modelos de estados finitos, para representar las diferentes fuentes de conocimiento. Se han empleado transductores estocásticos como estructuras de datos para los accesos al diccionario, así como en la comunicación entre módulos. En los procesos de desambiguación, se han aplicado los conocidos Modelos de Markov[12].

SisHiTra sigue una arquitectura modular basada en máquinas de estados finitos, por lo que proporciona un

entorno de trabajo homogéneo y eficiente. El proceso de traducción de un idioma fuente a un idioma destino se realiza en 4 fases, combinando elementos tanto de naturaleza lingüística como estadística (véase la Figura 1).

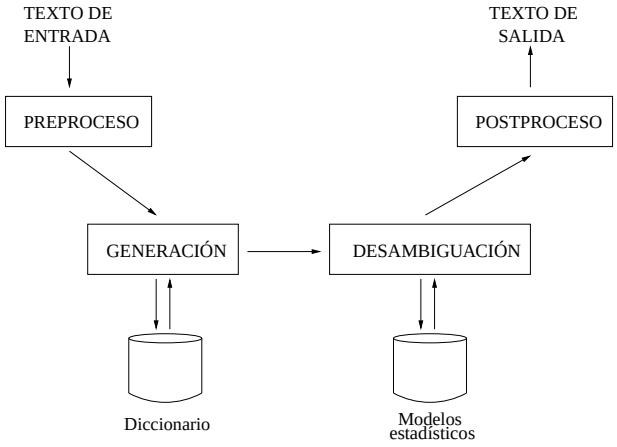


Figura 1. Arquitectura del sistema

La información parcial durante las etapas de traducción también se representa por medio de modelos de estados finitos. La arquitectura de SisHiTra se estructura del siguiente modo:

Preproceso: se encarga de dividir el texto de entrada en frases, y éstas, en elementos indivisibles (signos de puntuación, números, abreviaturas, nombres propios, palabras generales, etc.).

Generación: a través de una serie de accesos al diccionario, se produce un grafo sintáctico que representa todos los posibles análisis de la frase de entrada junto con todas sus posibles traducciones. En la Figura 2 se observa un diseño muy simple de la base de datos que constituye el diccionario, que para el par de lenguas que nos ocupa, dispone de un total de 80544 entradas.

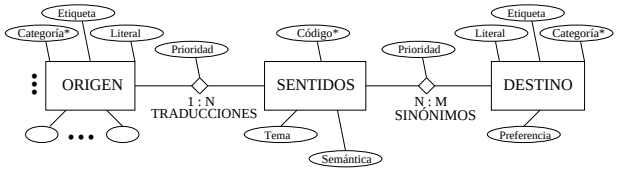


Figura 2. Diseño del diccionario

El uso de una aproximación basada en modelos de estados finitos resulta fundamental para una respuesta en tiempo real del sistema.

Desambiguación: este módulo realiza un proceso completo de desambiguación usando modelos estadísticos. En primer lugar, se desambigua morfosintácticamente a través de un proceso de etiquetado PoS, seleccionando la ruta sintáctica más probable dentro del grafo de análisis. Esto implica una segmentación de la frase de entrada, seleccionando una categoría léxica para cada segmento. A continuación, se realiza una desambiguación semántica, también por medio de métodos basados en el contexto.

Postproceso: aplica varias transformaciones basadas en reglas para transformar las frases de salida (no expresadas todavía en lenguaje natural) en frases correctas del idioma destino. En primer lugar, se realiza una concordancia de género y número a nivel de sintagma nominal. Seguidamente, se flexionan las palabras que admiten flexión (verbos, adjetivos o sustantivos), teniendo en cuenta algunos de sus rasgos lingüísticos, tales como género, número o persona. En tercer lugar, se realizan contracciones de palabras, elisiones de vocales, y se implementa correctamente el uso de pronombres. Por último, se realiza una conversión a texto plano en un lenguaje natural legible y sensible a mayúsculas.

Los módulos de SisHiTra se han desarrollado siguiendo dos esquemas diferentes: por un lado, los módulos basados en reglas (preproceso y postproceso) se han implementado en *Flex*, una herramienta muy útil para generar programas que realizan búsquedas de expresiones regulares en textos; por otro lado, los módulos de análisis (generación y desambiguación) se han escrito en *C*, siguiendo una estrategia de decodificación basada en el algoritmo de Viterbi.

4. MOTIVACIONES LINGÜÍSTICAS

Aquí nos limitaremos a mostrar un ejemplo de cómo se ha manejado la información lingüística para obtener una traducción de alta calidad. Para ilustrarlo, hemos seleccionado un aspecto lingüístico puntual que requiere una solución diferenciada en cada lengua. En concreto, nos referimos al comportamiento del complemento directo animado en español y catalán. A tal fin, reseñaremos la solución ofrecida por nuestro sistema.

Tanto en español como en catalán, el complemento directo se inserta directamente (de ahí el nombre) sin mediación de ningún tipo de partícula. Así, decimos:

Compré los tomates para el gazpacho.
Vaig comprar els tomàquets per al gaspatxo. (4)

y tanto en una lengua como en la otra, nuestros *tomates* aparecen ligados directamente al verbo.

Pero, cuando este complemento se refiere a un ente animado (por ejemplo, una persona), en español se introduce mediante la preposición *a*. Por contra, en catalán es indiferente que sea o no animado, ya que continúa conectándose al verbo (haciendo algunas salvedades) sin ningún tipo de elemento.

Ayer vimos a tu madre.
Ahir vam veure la teua mare. (5)

La solución que ofrece SisHiTra para esta frase es exactamente la que aparece en el ejemplo (5).

La elaboración de la regla lingüística que permite traducir correctamente estos casos es relativamente fácil. Tan sólo hay que etiquetar automáticamente aquellos verbos que son transitivos (es decir, que requieren un complemento directo) y de generar una regla que suprima la preposición *a* en catalán, cuando coincidan en la misma secuencia. Ahora bien, puede suceder que se den casos como:

Vimos a las tres de la tarde a tu madre.

Vam veure a les tres de la vesprada la teua mare. (6)

donde concurren un verbo transitivo y dos complementos introducidos por la preposición *a*. ¿Cómo puede decidir el traductor automático qué preposición debe suprimir? De hecho, la traducción resultante tiende a eliminar la preposición del primer complemento:

Vam veure les tres de la vesprada la teua mare. (7)

La verdadera dificultad reside, justamente, en identificar el complemento animado. También debemos contemplar que éste, en un uso vivo de la lengua, puede aparecer en un orden relativamente libre. Aún más, puede tratarse del complemento indirecto de un verbo transitivo (que a su vez posea un complemento directo introducido, por ejemplo, por la partícula que). En definitiva, se trata de considerar en abstracto todos los posibles contextos de concurrencia.

Si profundizamos un poco más en este fenómeno, observaremos que en él hay implícitas dos vertientes lingüísticas que tradicionalmente aparecen separadas. Por un lado, desde un punto de vista morfosintáctico, la presencia de un verbo transitivo que necesita completar su significación por medio de un complemento. Y por otro, semánticamente, que dicho complemento se refiera a una entidad animada. En cuanto al primero, la identificación automática de las categorías, su etiquetado, etc. o la inclusión de información de este tipo (y de muchos otros) en los diccionarios electrónicos, no suponen ninguna novedad. Sin embargo, en relación al segundo, la dificultad reside justamente en ser capaces de identificar la significación (animado vs no animado) del complemento.

Por nuestra parte, desde una perspectiva cognitivista, consideramos que ambos aspectos forman parte de un mismo todo y, por eso mismo, no pueden separarse. En ese sentido, los elementos que integran nuestro diccionario, se definen tanto por sus rasgos morfosintácticos como semánticos. Por ejemplo, el complemento directo que constituían nuestros *tomates* del ejemplo (4), poseen una etiqueta que señala su categoría gramatical (y, por tanto, su comportamiento sintáctico) y la subclase a la que pertenece dentro de ésta.

Así, decimos que se trata de un nombre[13] (N–), dado que pertenece a una clase léxica abierta, flexiva, etc., que prototípicamente es núcleo de un sintagma que, a su vez, puede ejercer las funciones de sujeto, de complemento, etc. Además decimos que es un nombre común (NC-), ya que presenta los rasgos de [+extensión], [+intensión] y [+referencialidad inherente]. En este caso, como en otros, hemos considerado que no nos es necesaria más información para nuestro propósito. Si ahora pasamos a la mencionada *madre*, comprobaremos que también aparece definida como un nombre común, pero ahora animado (NCA), porque para el par de lenguas con las que trabajamos este rasgo semántico sí que es pertinente para resolver dicha particularidad sintáctica.

Lo que tratamos de decir con todo ello es que la información semántica no sólo puede formalizarse (aunque sea muy someramente) sino que, además, puede resultar muy útil para tratar de resolver aspectos sintácticos. Es evidente que previamente hay que decidir y seleccionar qué rasgos semánticos pueden contribuir a una mejor resolución en la traducción entre un par de lenguas determinado. Creemos que esta digresión es necesaria para hacer ver que la voluntad de obtener traducciones automáticas de alta calidad precisa continuas reflexiones y revisiones sobre el trabajo realizado.

5. EVALUACIÓN

El rendimiento de toda aplicación de traducción comprende dos aspectos bien diferenciados y hasta cierto punto incompatibles: precisión y tiempo de respuesta. Ambos aspectos son importantes, ya que una traducción perfecta pero que requiera un tiempo excesivo de ejecución nos sirve de tan poco como una traducción instantánea pero completamente errónea. Debe llegarse a un equilibrio entre ambas aumentando la precisión lo máximo posible sin sacrificar la velocidad. Éste ha sido uno de los objetivos del nuevo diseño modular de SisHiTra y en los siguientes apartados veremos los resultados obtenidos.

Para evaluar el rendimiento de nuestra aproximación, disponemos de un corpus que consta de 240 pares de frases bilingües, extraídos de diversas fuentes, con una media de $18{,}3$ palabras por frase, contabilizando un total de 4387 palabras.

5.1. Otros sistemas

interNOSTRUM[14] es un sistema clásico de traducción indirecta por transferencia morfológica avanzada, semejante a la empleada en los sistemas comerciales para PC. Tiene una estructura modular, muy similar en muchos aspectos a la de SisHiTra, donde el análisis morfológico, el *PoS-tagging*, los diccionarios bilingües, y la información de formato cooperan todos juntos para proporcionar traducciones aproximadas del español al catalán en tiempo real. Se puede procesar texto en cualquiera de los siguientes formatos: ANSI, HTML, y RTF.

SALT[2] es un sistema de TA completamente basado en el conocimiento que posee un método interactivo que minimiza errores, añadiendo naturalidad a las traducciones

[2] http://www.cult.gva.es/salt/salt_programes_salt2.htm

entre español y catalán. Además, es un corrector ortográfico muy potente que detecta barbarismos, perífrasis o locuciones incorrectas, combinaciones incorrectas de pronombres, errores de concordancia, etc. Finalmente, es un programa de autoaprendizaje que enseña catalán a partir de los propios errores del usuario, con cientos de consejos sobre léxico, gramática, el uso apropiado de mayúsculas o minúsculas, signos de puntuación, etc.

AutomaticTrans[3] es un servidor de traducción fácilmente integrable en una red TCP/IP para procesar una gran cantidad de datos (miles de ficheros diarios) con excelente calidad. Es una plataforma de traducción que permite la selección explícita de un estilo linguístico, consiguiendo así una comunicación homogénea por la que es bien conocido. Con AutomaticTrans, que es más barato que las aproximaciones tradicionales a la traducción (pero no de peor calidad), los documentos se traducen simultáneamente a varios idiomas.

Comprendium[4] también es un sistema de TA comercial, desarrollado por la empresa Translendium SL, que se encuentra en Barcelona y forma parte del grupo europeo Braintribe, líder en Gestión de Contenidos de Empresa multilingües. La tecnología de Comprendium es líder mundial en TA tanto por el número de pares de idiomas así como por la funcionalidad ofrecida por el sistema. Éste consiste en un motor de traducción, con una estructura modular de gramáticas y léxicos, el cual realiza un análisis morfosintáctico del texto de entrada para luego traducirlo al idioma destino. El motor puede conectarse a una serie de módulos de memorias de traducción así como a un editor profesional de diccionarios. El usuario puede acceder al mismo a través de un servidor multiusuario, bien desde un cliente web, o bien desde una aplicación profesional monousuario. Se pueden crear distintas configuraciones del producto según las necesidades del usuario. Comprendium es el sucesor de INCYTA, sistema pionero en TA entre español y catalán.

5.2. Precisión

El WER es una medida objetiva de la calidad de la traducción que calcula la distancia de edición a nivel de palabra entre una hipótesis de traducción y una traducción de referencia predefinida. La distancia de edición calcula el número de sustituciones, inserciones y borrados que son necesarios para trasformar la hipótesis de traducción en la traducción de referencia. El número acumulado de errores para todas las frases de test se divide entre el total de palabras del mismo, y el porcentaje resultante nos muestra el número medio de palabras incorrectas. Los resultados de WER para SisHiTra son similares a los obtenidos por el resto de sistemas analizados, tal y como vemos en la Tabla 1.

Una desventaja del WER es que sólo compara la hipótesis de traducción con una única traducción de referencia.

[3]http://www.automatictrans.es

[4]http://www.translendium.com

Sistema	WER
interNOSTRUM	12.6
SALT 3.0	12.2
AutomaticTrans	**10.3**
Comprendium	11.5
SisHiTra	11.8

Tabla 1. WER para varios sistemas de traducción

Esto no ofrece ningún margen a traducciones correctas con un estilo de escritura diferente. Por lo tanto, para evitar este problema, evaluamos nuestro sistema mediante el MWER, que considera varias traducciones de referencia para una misma frase de test, calcula la distancia de edición de la hipótesis con todas ellas, y retorna el mínimo WER obtenido. El MWER ofrece una medida más realista que el WER ya que permite una mayor variabilidad en el estilo de traducción.

Utilizaremos un total de 3 referencias, incorporando una serie de traducciones generadas por lingüistas a partir de la referencia original.

Sistema	MWER
interNOSTRUM	6.5
SALT 3.0	6.1
AutomaticTrans	5.2
Comprendium	5.7
SisHiTra	**3.3**

Tabla 2. MWER para varios sistemas de traducción

Los resultados de MWER para SisHiTra son los mejores de entre los cinco sistemas analizados, tal y como vemos en la Tabla 2.

5.3. Tiempo de respuesta

El tiempo de respuesta es el tiempo transcurrido desde que el usuario lanza una petición hasta que el sistema produce el resultado correspondiente.

Para estimar el tiempo de respuesta de SisHiTra, hemos medido el tiempo empleado por cada módulo para procesar el corpus de evaluación. Así, sumando los tiempos empleados por módulo, obtenemos una estimación sobre el tiempo de respuesta de nuestro sistema, considerando despreciables los tiempos internos de comunicación entre módulos.

Los experimentos se realizaron en un computador con un procesador Pentium 4 a 3.2 GHz y 2 GB de memoria. Hemos repetido cada prueba 20 veces con el fin de que los resultados estén dotados de la máxima fiabilidad posible, calculando la media y la desviación típica de los tiempos obtenidos (véase la Tabla 3).

De este modo, dividiendo el total de palabras del corpus de evaluación, 4387, por el tiempo medio observado para su procesamiento, $0{,}8745$ segundos, podremos obtener una estimación de la velocidad de nuestro sis-

	Tiempo	
Módulo	Media	Desv. Típica
Preproceso	0.0180	0.0041
Generación	0.2605	0.0147
Desambiguación	0.1815	0.0067
Postproceso	0.4145	0.0051
Total	0.8745	0.0188

Tabla 3. Tiempos de respuesta en segundos

tema, $\frac{4387}{0{,}8745} \approx 5017$ palabras/segundo, considerado como tiempo real. Tan sólo interNOSTRUM reporta una velocidad comparable de 5000 palabras por segundo. El resto de sistemas presentan una productividad inferior.

6. CONCLUSIONES

En este artículo, hemos realizado una evaluación exhaustiva de SisHiTra, un paradigma de computación en el que se combinan técnicas deductivas e inductivas para el desarrollo de sistemas de traducción automática de dominio abierto entre idiomas parejos.

El rendimiento de esta aproximación se ha demostrado sobre el par de lenguas español y catalán, comparando los resultados obtenidos frente a otros sistemas de traducción automática. El uso de máquinas de estados finitos para modelar todas las fuentes de conocimiento ha permitido un diseño muy eficiente del sistema, logrando un compromiso muy interesante entre precisión y eficiencia computacional.

Como trabajo futuro, pretendemos desarrollar un prototipo inverso de traducción, es decir, catalán⇒español, a partir del diccionario actual. Por otro lado, y dentro del marco lingüístico, hemos iniciado manualmente el marcaje y la revisión de las entradas del diccionario. A la vez, estamos perfilando una serie de reglas lingüísticas considerando todos los contextos posibles, para ajustar los resultados. Además, los principales aspectos a mejorar son:

- Tiempo de respuesta, optimizando el módulo de postproceso cuya carga es excesiva porcentualmente.
- El desarrollo de una serie de aplicaciones de usuario que permitan la explotación masiva de las características de SisHiTra.
- Emplear SisHiTra sobre otras lenguas.

7. BIBLIOGRAFÍA

[1] L. Karttunen, "Citation of unpublished documents," Tech. Rep., XEROX Palo Alto Research Center, 1993.

[2] Emmanuel Roche y Yves Schabes, Eds., *Finite-State Language Processing*, Bradford Book. MIT Press, Cambridge, Massachusetts, USA, 1997.

[3] Emmanuel Roche, "Finite state transducers: parsing free and frozen sentences," pp. 108–120, 1999.

[4] M. Mehryar, "Finite-state transducers in language and speech processing," 1997.

[5] Mehryar Mohri, Fernando Pereira, y Michael Riley, "The design principles of a weighted finite-state transducer library," *Theoretical Computer Science*, vol. 231, no. 1, pp. 17–32, 2000.

[6] Kemal Oflazer, "Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction.," *Computational Linguistics*, vol. 22, no. 1, pp. 73–89, 1996.

[7] Emmanuel Roche y Yves Schabes, "Deterministic part-of-speech tagging with finite-state transducers," *Computational Linguistics*, vol. 21, no. 2, pp. 227–253, 1995.

[8] F. Casacuberta, H. Ney, F. J. Och, E. Vidal, J. M. Vilar, S. Barrachina, I. Garcia-Varea, C. Martinez D. Llorens, S. Molau, F. Nevado, M. Pastor, D. Pico, y A. Sanchis., "Some approaches to statistical and finite-state speech-to-speech translation," *Computer Speech and Language*, vol. 18, pp. 25–47, 2004.

[9] J. González, A. L. Lagarda, J. R. Navarro, L. Eliodoro, A. Giménez, F. Casacuberta, J. M de Val, y F. Fabregat, "SisHiTra: a Spanish-to-Catalan hybrid machine translation system.," in *5th SALTMIL Workshop on Minority Languages*, Genoa, Italy, 23 May 2006, pp. 69–73.

[10] A. Viterbi, "Error bounds for convolutional codes and an asymtotically optimal decoding algorithm," *Annals of the New York Academy of Sciences*, vol. IT-13, pp. 260–269, 1967.

[11] José R. Navarro, Jorge González, David Picó, Francisco Casacuberta, Joan M. de Val, Ferran Fabregat, Ferran Pla, y Jesús Tomás, "Sishitra: a hybrid machine translation system from spanish to catalan.," in *EsTAL*, 2004, pp. 349–359.

[12] A. Sanchis, D. Picó, J.M. del Val, F. Fabregat, J. Tomás, F. Casacuberta, y E. Vidal, "A morphological analyser for machine translation based on finite-state transducers," in *MT Summit VIII*, September 2001, pp. 305–309.

[13] M.J. Cuenca, *Sintaxi fonamental: les categories gramaticals*, Empúries, 1996.

[14] M. Forcada, A. Garrido, R. Canals, A. Iturraspe, S. Montserrat-Buendia, A. Esteve, S. Ortiz Rojas, H. Pastor, y P.M. Pérez, "The spanish-catalan machine translation system internostrum," *0922-6567 - Machine Translation*, vol. VIII, pp. 73–76, 2001.

N-BEST REORDERING IN STATISTICAL MACHINE TRANSLATION

Germán Sanchis[*], *Francisco Casacuberta*[*]

[*]Departamento de Sistemas Informaticos y Computación
Universidad Politécnica de Valencia
Camino de Vera, s/n. 46071 Valencia, Spain
{gsanchis,fcn}@dsic.upv.es

ABSTRACT

As statistical machine translation (SMT) systems strive to improve the translation quality they are able to deliver, the word reordering problem is being unveiled as a major problem that must be addressed, whenever these systems are to be improved. While most works published focus their results in corpora involving English, Chinese and Arabic, such a translation problem can also be found within Spain itself: its origin being unknown, Basque presents a very peculiar word order, which is very different to most other european languages, and specially very different to Spanish word order. Because of this fact, SMT systems not including some sort of word reordering yield unsatisfactory results, involving serious training problems, when confronted with the Basque-Spanish task. Although some efforts have been made towards including word or phrase reordering in the decoding algorithms, these approaches usually imply a computational overhead that obliges the designers of such algorithms to assume sub-optimal restrictions, which often lead to a significant dimish in the translation quality. In this work, we present a reordering method based on the extraction and exploitation of monotized corpora, which prove to be specially useful for the language pairs presenting severe word reorderings. Our system has been tested on the Basque Tourist task, where very promissing results have been obtained.

1. INTRODUCTION

SMT systems have proved in the last years to be an efficient way of building machine translation systems which, with little need of human supervision, are able to deliver a translation quality very similar, if not better, than most commercial machine translation systems, which are usually characterized by having a very important development effort behind each pair of languages they intend to translate.

Statistically, the machine translation problem can be defined as, given a sentence s from a certain source language, the search for an adequate sentence $\hat{t}$ that maximises the posterior probability:

$$\hat{t} = \underset{t}{\operatorname{argmax}}\, Pr(t|s)$$

However, this probability is decomposed, in the vast majority of cases, into two different probabilities, the first being the target statistical language model and the second one being the translation model. Hence, the previous equation is transformed by means of the Bayes' theorem to obtain the following equation:

$$\hat{t} = \underset{t}{\operatorname{argmax}}\, Pr(t) \cdot Pr(s|t)$$

More intuitively, the translation model $Pr(s|t)$ will capture word relations between both languages, and is normally based on stochastic dictionaries and alignment models, whereas the language model $Pr(t)$ will award a higher probability to well-formed sentences from the target language.

Monotonous SMT systems are nowadays the most used in research for Machine Translation. The reason is simple, but conclusive: non-monotonous models are computationally too expensive, needing restrictions to be applied for them to work. Being necessary, these restrictions are quite often applied in training and search, leading to a substantial reduction of the model's performance.

Because of this reason, most SMT systems that have been developped in the last years, including the vast majority of the state of the art systems, either implement a Phrase-Based model [1, 2] or Weighted Finite State Transducers for Machine Translation [3, 4]. These systems are inherently monotonous, and are usually estimated using word-level aligned corpora, hence often incurring in ordering related errors. Although they try to allow some sort of reordering, such reorderings are quite often very limited, and are not able to account for the more wild reorderings that take place in languages from very different origins. This does not only imply that the output sentence will present a grammatically incorrect word order, but, moreover, the parameters of the systems trained can be estimated incorrectly, by assuming that a given input phrase is the translation of a certain output phrase, leading this assumption in many cases to incorrect translations.

The problem of word reordering has been tackled with already since the origin of what is now a days known as machine translation: Berger et al. [5] already introduced in their alignment models what they called distortion models, in an effort towards including in their SMT system a solution for the reordering problem. However, these distortion models are usually implemented within the decoding algorithms and imply serious computational problems, leading ultimately to restrictions being applied to the set of possible permutations of the output sentence. Hence, the search performed turns sub-optimal, and an important loss in the representational power of the distortion models takes place.

The most nave - but effective - approach to the reordering problem would be to allow for arbitrary word reorderings, and choose the one that obtains the highest score in some reordering scoring model. However, when allowing all possible word permutations the search has been proved to be NP-hard [6].

In our work we present a novel approach to solve the reordering problem, based on the work of Zens, Matusov and Kanthak [7, 8, 9], who introduce the idea of monotonizing a corpus, i.e. using the alignments produced by the IBM models to reorder the input sentence s and produce a new bilingual pair, composed by the reordered input sentence s' and the output sentence t, whose translation is monotonous. Once this is achieved, any random monotonous translation model may be trained without the problems derived from word reorderings. In this paper, the monotonized corpus will be used to train a Phrase Based model.

However, a crucial problem has to be addressed when trying to learn models from monotonized corpora: in search time, the output sentence is not available, hence the need for a model that will be able to learn how to reorder efficiently a given input sentence. Our approach copes this problem by learning a very simple reordering model and generating an n-best list of reordering hypothesis. After this set of hypothesis has been translated, only the best scoring sentence with respect to the translation model is kept as final output sentence.

The next section briefly describes some of the latest efforts made towards solving this problem. Then, in section 3, we will explain our approach, based on an n-best list of best reordering hypothesis. In section 4 we will describe the experiments performed with our systems and the results obtained. Finally, in section 5 we will present the conclusions that can be elucidated from the experiments described, as well as the work that is currently in progress.

2. BRIEF REVIEW OF EXISTING APPROACHES

When facing the reordering problem for Machine Translation, three main possibilities exist: output sentence reordering, input sentence reordering and the reordering of both, the latter being as of yet unexplored.

- Let s be a source sentence, t a target sentence.
- Let C be a cost matrix, such that c_{ij} = cost of aligning s_j to t_i.
- Let $\{s^r\}$ be the set of all possible permutations of s.
  1. compute alignment $A_D(j) = \underset{i}{\operatorname{argmin}}\, c_{ij}$
  2. $s' = \{s^r | \forall j : A_D(j) \leq A_D(j+1)\}$
  3. recompute (reorder) C, obtaining C'.
  4. set $A'_I(i) = \underset{j}{\operatorname{argmin}}\, {c'}_{ij}$.
  5. Optional: Compute the minimum-cost *monotonic* path through the cost matrix C'.

Figure 1. Algorithm for obtaining a monotonic alignment by reordering the source sentence.

2.1. Output sentence reordering

Two main approaches have been tested in this direction, the first one being originally developed by J.M.Vilar et al. [10], and more recently taken up again at the TALP Research Center [11]. The idea behind this approach is to monotonize the most probable non-monotonous alignment patterns and add a mark in order to be able to remember the original word order. This being done, a new output language has been defined and a new language and translation model can be trained, being the translation process now monotonous.

Another approach more recently tested by [12] involves learning weighted finite state transducers that account for local reorderings of two or three positions, allowing each word to jump a maximum of one or two positions. They applied these models for phrase reordering, but the training of the models did not yield statistically significant results with respect to the introduction of the models with fixed probabilities.

2.2. Input sentence reordering and corpora monotonization

The main idea behind this approach developed at the RWTH-Aachen [7, 8, 9], is to avoid the non-monotonous translation problem by reordering the input sentence in such a way that the translation model will not need to account for possible word reorderings. To achieve this, alignment models are used, in order to establish which word order should be the appropriate for the translation to be monotonous, and then the input sentence is reordered in such a manner that the alignment is monotonous. The algorithm performing such a reordering is described in Figure 1.

However, this approach has an obvious problem, since the output sentence is not available in search time and the sentence pair cannot be monotonized. The nave solution, test on all possible permutations of the input sentence, has already been discussed earlier, being NP-hard [6], as $J!$ possible permutations can be obtained from a sentence of length J. Hence, the search space must be restricted.

On their paper, Kanthak et al. [9] describe four constraints.

- IBM constraints, described in Berger et al. [5]. At each moment, the first k yet uncovered word positions can be chosen to be the next word, being an uncovered word position one such position that has not yet been chosen.
- Inverse IBM constraints. In this case, any word can be chosen, except if $k-1$ words on positions $j' > j$ have already been chosen, in which case j must be the next word. This is specially useful for language pairs in which we need to translate words at the end of the sentence, and then translate the rest nearly monotonically.
- Local constraints. Here, the next word must be contained within the first k positions, starting from the first yet uncovered position, and counting both covered and not covered positions. This constraint is useful for language pairs in which words are only shifted a few positions.
- ITG constraints. Being inspired by the bilingual bracketing of inverse transduction grammars (ITG) [13], these constraints have experimentally proven to be useful in Machine Translation. In this approach, the sentence is parsed into a binary tree. In the nodes of such a tree, we can choose either to invert the original order, or leave it as is. Because of its nature, ITG constraints forbid combinations such as (3, 1, 4, 2) or (2, 4, 1, 3).

3. THE REORDERING MODEL AND N-BEST REORDERINGS

However, and although the constraints described above did yield interesting results, the search space containing the permutations still remains huge, paying a very expensive computational price for the relatively small benefit obtained.

In addition, the reordering constraints described in the work by Kanthak et al. [9] disregards the crucial information that can be extrated from the monotonized corpora, for it is within this type of corpus that lies the information needed to train a reordering model which, ideally,would be capable of reordering the input sentences in such a manner that the translation following would be monotonous. Although such a model is bound to introduce error into the overall system, but the benefits obtained may well be worth it.

The main idea behind our approach is that monotonized alignments define a new source "*language*", and hence a reordered language model can be trained with the reordered input sentences s'. Such a reordered language will have the same vocabulary as the source language, but the word ordering of the target language, and hence a reordering model can be trained from the monotonized corpus. Moreover, the reordering model will most likely not need to depend on the output sentence, whenever the word-by-word translation is accurate enough.

Hence, the reordering problem can be defined as follows:

$$s' = \underset{s^r}{\operatorname{argmax}}\, Pr(s^r) \cdot Pr(s|s^r)$$

where $Pr(s^r)$ is the reordered language model, and $Pr(s|s^r)$ is the reordering model. Being this problem very similar to the translation problem but with a very constrained translation table, it seems only natural to use the same methods developed to solve the translation problem to face the reordering problem. Hence, in this paper we will be using an exponential model as reordering model, defined as:

$$Pr(s|s') \approx exp(-\sum_i d_i)$$

where d_i is the distance between the last reordered word position and the current candidate position.

However, and in order to reduce the error that will introduce a reordering model into the system, we found that it is very useful to compute an n-best list of reordering hypothesis and translate them all, selecting then as final output sentence the one which obtains the highest probability according to the models $Pr(t) \cdot Pr(s|t)$. Ultimately, what we are actually doing with this procedure is to contrain the search space of permutations of the source sentence as well, but taking into account the information that monotonized alignments entail, together with a much stronger restriction of the search space than previous approaches, reducing significantly the computational effort needed.

4. TRANSLATION EXPERIMENTS

4.1. Corpus characteristics

Our system has been tested on the Basque-Spanish translation task, a tough problem with no trivial solution in which reordering plays a crucial role.

The corpus chosen for this experiment is the *Tourist* corpus [14], which is an adaptation of a set of Spanish-German grammars generating bilingual sentence pairs [15] in such languages. Hence, the corpus is semi-synthetic. In this task, the sentences describe typical human dialogs in the reception desk of a hotel, and they have been mainly extracted from tourist guides. However, because of its design, there is some asymmetry between both languages,

Table 1. Characteristics of the Tourist corpus.

		Spanish	Basque
Training	Sentences	38940	
	Different pairs	20318	
	Words	368314	290868
	Vocabulary	722	884
	Average length	9.5	7.5
Test	Sentences	1000	
	Test independent	434	
	Words	9507	7453
	Average length	9.5	7.5

and a concept being expressed in several manners in the source language will always be translated in the same manner in the target language. Because of this, the target language is meant to be simpler than the source language. Since the input language during the design of the corpus was Spanish, the vocabulary size of Basque should be smaller. Actually, however, the vocabulary size of Basque is bigger than that of Spanish, and this is due to the agglomerative nature of the Basque language. The corpus has been divided into two separate subsets, a bigger one for training and a smaller one for test. The characteristics of this corpus can be seen in Table 1.

4.2. System evaluation

The SMT system developed has been automatically evaluated by measuring the following rates:

WER *(Word Error Rate)*: The WER criterion is similar to the edit distance for Speech Recognition, computing the minimum number of editions (substitution, insertion and deletion operations) needed to convert the translated sentence into the sentence considered ground truth. This measure is because of its nature a pessimistic one. For example, input sentences may allow many different translation, but the WER criterion will penalize all those translations which hold a difference with respect to the translation considered ground truth.

PER *(position-independent WER)*: The PER criterion is similar to WER, but word order is ignored. This criterion accounts for the fact that a system may produce an acceptable translation differing only in word order. This sentence may even be grammatically correct, but will be nevertheless penalized when using the WER criterion.

BLEU *(BiLingual Evaluation Understudy) score*: This score measures the precision of unigrams, bigrams, trigrams, and 4-grams with respect to a whole set of reference translations, with a penalty for too-short sentences [16]. It must be noted that BLEU measures accuracy, not error rate, which means that the higher the BLEU score, the better.

Table 2. Results for Spanish to Basque translation.

	Baseline translation	Reordered translation
WER	19.5%	10.9%
BLEU	81.0%	87.1%
PER	6.2%	4.9%

4.3. Experimental setup and translation results

We used the reordering technique described above to obtain an n-best reordering hypothesis list and translate them, keeping the best scoring one. In this case, n was set to 5.

First, the bilingual pairs were aligned using IBM model 4 by means of the GIZA++ toolkit [17]. After this, the alignments were monotonized in the fashion described in [9] and a new alignment was recalculated, determining the new monotonous alignment between the reordered source sentence and the target, and a reordered source sentence language model was built. Phrase extraction was performed by using the Thot toolkit [18].

For the next step, the reordering model, we used the reordering model built in the toolkit Pharaoh. This was done by including in the translation table only the words contained in the vocabulary of the desired source language, and allowing the toolkit to reorder the words by taking into account the language model and the phrase-reordering model it implements, which is an exponential model. Since in this case, the phrases are just words, what results is an effective implementation of an exponential word-reordering model, just as we wanted.

Once the 5 best reordering hypothesis had been calculated, we translated them all by using the toolkit Pharaoh once again, and kept just the best scoring translation, where the score is determined as the product of the (inverse) translation model and the language model.

The results of this setup can be seen in Table 2. As a baseline, we took the results of translating the same test set, but without the reordering pipeline, i.e. just using GIZA++ for aligning, Thot for phrase extraction and Pharaoh for translating.

In these results it can be seen that the reordering pipeline established does have significant benefits on the overall quality of the translation, almost achieving even a relative improvement of 50% of the Word Error Rate. Furthermode, it is interesting to point out that even in the case of the PER criterion the results obtained are better. At first sight, this might seem odd, since the PER criterion does not take into account possible ordering errors within the sentence, which is a main problem that every reordering technique tries to solve. However, we found that this improvement is due to the fact that reordering the source sentence allows for better phrases to be extracted.

5. CONCLUSIONS AND FUTURE WORK

A reordering technique has been implemented, taking profit of the information that monotonized corpora provide. By doing so, better quality phrases can be extracted and the overall performance of the system improves significantly in the case of a pair of languages which present heavy reordering complications.

This technique has been applied to translate a semi-synthetic corpus which deals with the task of Spanish-Basque translation, and the results obtained prove to be statistically signficant and show to be very promissing. Moreover, the technique we propose in this paper is learnt authomatically, without any need of linguistic annotation or manually specified syntatic reordering rules, which means that out technique can be applied to any language pair without need for any additional development effort.

Both reordered corpora and reordering techniques seem to have a very important potential for the case of very different language pairs, which are the most difficult translation tasks.

As future work, we are planning on obtaining results with other non-synthetic, richer and more complex corpora, as may be other Spanish-Basque corpora or corpora involving language pairs such as Arabic, Chinese or Japanese. In addition, we are planning on developping more specific reordering models, which will be more suitable for this task than the exponential model described here.

Acknowledgements

This work has been partially supported by Spanish project TIC 2003-08681-C02-02 and by the MEC scholarship AP2005-4023.

6. BIBLIOGRAPHY

[1] P. Koehn, F.J. Och, y D. Marcu, "Statistical phrase-based translation," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Edmonton, Canada, 2003, vol. 1, pp. 48–54.

[2] F.J. Och y H. Ney, "The alignment template approach to statistical machine translation," *Computational Linguistics*, vol. 30, no. 1, pp. 417–449, 2004.

[3] F. Casacuberta y E. Vidal, "Machine translation with inferred stochastic finite-state transducers," *Computational Linguistics*, vol. 30, no. 2, pp. 205–225, 2004.

[4] S. Kumar y W. Byrne, "A weighted finite state transducer implementation of the alignment template model for statistical machine translation," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Edmonton, Canada, 2003, vol. 1, pp. 63–70.

[5] A.L. Berger, P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, J.R. Gillet, A.S. Kehler, y R.L. Mercer, "Language translation apparatus and method of using context-based translation models," in *United States Patent 5510981*, 1996.

[6] K. Knight, "Decoding complexity in word-replacement translation models," *Computational Linguistics*, vol. 25, no. 4, pp. 607–615, 1999.

[7] R. Zens, H. Ney, T. Watanabe, y E. Sumita, "Reordering constraints for phrase-based statistical machine translation," in *COLING '04: The 20th Int. Conf. on Computational Linguistics*, Geneva, Switzerland, 2004, pp. 205–211.

[8] E. Matusov, S. Kanthak, y H. Ney, "Efficient statistical machine translation with constrained reordering," in *In Proceedings of EAMT 2005 (10th Annual Conference of the European Association for Machine Translation)*, Budapest, Hungary, 2005, pp. 181–188.

[9] S. Kanthak, D. Vilar, E. Matusov, R. Zens, y H. Ney, "Novel reordering approaches in phrase-based statistical machine translation," in *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, Ann Arbor, Michigan, 2005, pp. 167–174.

[10] J.M. Vilar, E. Vidal, y J.C. Amengual, "Learning extended finite-state models for language translation," in *Proc. of Extended Finite State Models Workshop (of ECAI'96)*, Budapest, August 1996, pp. 92–96.

[11] A. de Gispert y J.B. Mario, "Experiments in word-reordering and morphological preprocessing for transducer-based statistical machine translation," in *Proc. of 2003 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU'03)*, USA, 2003.

[12] S. Kumar y W. Byrne, "Local phrase reordering models for statistical machine translation," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, Vancouver, Canada, 2005, pp. 161–168.

[13] D. Wu, "Stochastic iversion transduction grammars and bilingual parsing of parallel corpora," *Computational Linguistics*, vol. 23, no. 3, pp. 377–403.

[14] A. P´erez, F. Casacuberta, M.I. Torres, y V. Guijarrubia, "Finite-state transducers based on k-tss grammars for speech translation," in *Proceedings of Finite-State Methods and Natural Language Processing (FSMNLP 2005)*, Helsinki, Finland, September 2005, pp. 270–272.

[15] E. Vidal, "Finite-state speech-to-speech translation," in *Int. Conf. on Acoustics Speech and Signal Processing (ICASSP-97), proc., Vol.1*, Munich, Germany, 1997, pp. 111–114.

[16] Papineni, A. Kishore, S. Roukos, T. Ward, y W. Jing Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Technical Report RC22176 (W0109-022)*, IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY, 2001.

[17] F. J. Och y H. Ney, "Improved statistical alignment models," Hongkong, China, October 2000, pp. 440–447.

[18] D. Ortiz, I. Garca-Varea, y F. Casacuberta, "Thot: a toolkit to train phrase-based statistical translation models," in *Tenth Machine Translation Summit*, pp. 141–148. Asia-Pacific Association for Machine Translation, Phuket, Thailand, September 2005.

OPTIMIZACIÓN DEL PROCESO DE GENERACIÓN DE GRAMÁTICAS PARA RAH BASADAS EN REGLAS Y APLICACIÓN EN UN SISTEMA DE DIÁLOGO

Zoraida Callejas Carrión, Ramón López-Cózar Delgado

Dpto. Lenguajes y Sistemas Informáticos, ETS Ingenierías Informática y Telecomunicaciones, Universidad de Granada, {zoraida, rlopezc}@ugr.es

RESUMEN

En este artículo presentamos una técnica para la actualización de gramáticas para RAH basadas en reglas cuyo vocabulario esté almacenado en una o varias bases de datos. La técnica, denominada CGBD (Creación de Gramáticas Basada en Disparadores), tiene como finalidad actualizar automáticamente el vocabulario de dichas gramáticas mediante un mecanismo de disparadores (triggers) de bases de datos, conforme se producen cambios en el contenido de éstas. Describimos asimismo cómo esta técnica ha sido implementada mediante la herramienta de Generación Automática de Gramáticas (GAG). Los resultados de evaluación, obtenidos usando un sistema de diálogo experimental, muestran que la técnica mejora significativamente la percepción que tienen los usuarios de prueba respecto a la velocidad de interacción del sistema. En comparación con otras técnicas existentes, la técnica CGBD ha sido preferida en la mayoría de las ocasiones, especialmente a partir de un valor umbral de 300.000 palabras en el vocabulario.

1. INTRODUCCIÓN

Las gramáticas para reconocimiento automático del habla (RAH) permiten alcanzar una gran precisión en aplicaciones con un vocabulario y tipo de construcciones gramaticales muy bien delimitados. Es el caso de los sistemas de diálogo diseñados para proveer información, que normalmente tienen en cuenta vocabulario concerniente a las distintas consultas posibles (p.e. horarios de trenes [1], de aviones [2], o información en exposiciones y museos [3]).

Sin embargo, en algunas aplicaciones el vocabulario aceptado por las gramáticas puede ser desconocido a priori por estar almacenado en una (o varias) bases de datos (p.e. nombres de profesores de una universidad) en contraposición con aquellas en que el vocabulario es conocido desde el momento de diseño del sistema (p.e. confirmaciones y negaciones). En la literatura podemos encontrar diversas aproximaciones a la construcción de gramáticas para RAH basadas en reglas, que podemos clasificar en dos grandes grupos: el primero crea las reglas de forma estática, mientras que el segundo lo hace dinámicamente. Ambos enfoques poseen ventajas y desventajas. Un problema del enfoque estático es que las reglas no se actualizan cuando se producen cambios en las bases de datos y por tanto las frases aceptadas pueden quedarse obsoletas. En el caso de gramáticas dinámicas, una desventaja es que la extracción del vocabulario y la creación de las reglas puede traer consigo tiempos de respuesta del sistema muy grandes. Para solventar estos problemas, presentamos una técnica que crea las reglas estáticamente con vocabulario desconocido a priori, y las actualiza empleando un mecanismo de disparadores cuando se producen cambios en las bases de datos. La versión preliminar de esta técnica fue presentada en [4], en el presente artículo describimos la versión completa y los resultados de su evaluación con usuarios reales.

El artículo está organizado de la siguiente forma. En la sección 2 presentamos el estado del arte de la creación de gramáticas basadas en reglas. En la sección 3 presentamos nuestro enfoque para llevar a cabo esta tarea y describimos la implementación empleada para probar su efectividad. En la sección 4 describimos el sistema de diálogo oral experimental en el que hemos evaluado la técnica y los resultados obtenidos. Finalmente, la sección 5 presenta las conclusiones y propone algunas posibilidades de trabajo futuro.

2. TÉCNICAS EXISTENTES PARA LA CREACIÓN DE GRAMÁTICAS BASADAS EN REGLAS

2.1. Creación estática

La forma más sencilla de construir gramáticas para el RAH basadas en reglas consiste en crear dichas reglas de forma estática. Además, es la forma más rápida si el vocabulario permanece inalterado. De hecho, la mayoría de los entornos comerciales para el desarrollo de sistemas de diálogo oral proveen de herramientas para la creación y prueba de gramáticas basadas en reglas (p.e. IBM WebSphere, Voxeo o Nuance). Sin embargo, cuando el vocabulario no es conocido a priori sino que se encuentra almacenado en una o varias bases de datos, la creación estática puede provocar inconsistencias entre los contenidos de ésta y el vocabulario incluido en las gramáticas.

2.2. Creación dinámica

En la literatura encontramos diversas técnicas para crear gramáticas para RAH dinámicamente cuando el vocabulario aceptado por la aplicación es desconocido a priori. Una primera técnica construye las gramáticas dinámicamente en tiempo de procesamiento [5]. Esta técnica ofrece flexibilidad puesto que las gramáticas generadas de esta manera siempre están actualizadas con los últimos cambios en los datos. Sin embargo, este método puede implicar una carga computacional enorme, que se traduce en un incremento del tiempo de ejecución. En aplicaciones con bases de datos muy grandes, este tiempo puede ser excesivo, provocando que el sistema sea considerado por los usuarios "lento" en su tiempo de respuesta.

Una segunda técnica crea las gramáticas antes de que tenga lugar el proceso de reconocimiento automático del habla. Estas gramáticas se actualizan siempre antes de la ejecución del sistema independientemente de los cambios en las bases de datos [6]. Este método no implica un incremento en el tiempo de reconocimiento, con lo que el incremento en el tiempo de espera es menor desde el punto de vista del usuario, pero sigue habiendo un retardo al comienzo de la ejecución que se ve incrementado al hacerlo el tamaño de las bases de datos.

En ambos casos, la aplicabilidad de la técnica depende del tamaño del vocabulario. Como se ha discutido en diversos trabajos previos (p.e. [7]) hay tres tiempos de ejecución básicos a ser considerados:

- El límite para que le usuario considere que la aplicación se realiza en tiempo real es de 0.1 segundos.
- De 0.1 a 1.0 segundos el usuario nota un retardo pero su flujo de pensamiento no se interrumpe.
- De 1.0 a 10.0 segundos, el usuario todavía presta atención al diálogo pero para retardos mayores es necesario darle alguna información con un mensaje del sistema y/o una melodía para indicarle que el sistema se encuentra procesando la respuesta [8].

Hemos medido en nuestro laboratorio el tiempo de ejecución necesario para construir una gramática de una única regla de gramática a partir de una columna específica de una base de datos, teniendo en cuenta el número de palabras existentes en la misma. Como se muestra en la Figura 1, el límite 0.1 se alcanza con 10.000 palabras, el límite de 1 segundo con 100.000 palabras, y el límite máximo (10 segundos) se alcanza con vocabularios entre 300.000 y 400.000 palabras.

Por ejemplo, para un vocabulario de un millón de palabras obtenemos un retardo de 30 segundos. Por tanto, para vocabularios grandes (más de 300.000 palabras) el usuario se percata de la sobrecarga introducida y se necesita una técnica de retroalimentación que le mantenga informado de los tiempos de espera. Desafortunadamente, estas técnicas suelen causar una mala impresión a los usuarios, que normalmente las encuentran muy irritantes [9]. Además, un retardo en la respuesta del sistema lo más pequeño posible, definido éste como el tiempo transcurrido entre que el usuario termina de hablar y el sistema responde [10], es vital para minimizar los costes de interacción. Esto, unido con la maximización del éxito en la tarea, es vital para mejorar el grado de satisfacción de los usuarios y se emplea como parámetro clave en la evaluación de sistemas de diálogo, tal y como se propone en el marco PARADISE [10].

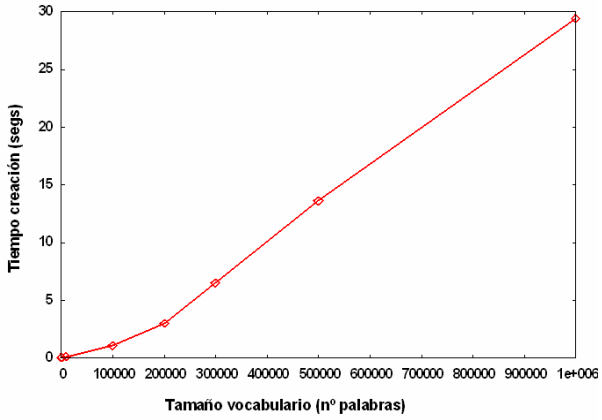


Figura 1. *Latencia en el proceso de creación dinámica de gramáticas para RAH basadas en reglas.*

3. TÉCNICA CGBD

Para evitar los problemas comentados en la sección anterior, proponemos una nueva técnica denominada CGBD (Creación de Gramáticas Basada en Disparadores) para optimizar el proceso de creación de gramáticas para RAH basadas en reglas. La idea de la misma es actualizar el vocabulario de dichas de forma automática conforme se producen cambios en las bases de datos utilizadas, empleando para ello un mecanismo de disparadores. De esta forma, no hay retardo de creación en tiempo de ejecución y el usuario siente que está interactuando con un sistema rápido.

La mayor ventaja de esta técnica en comparación con otras que encontramos en la literatura (p.e. [11]) es que es general y por tanto adecuada para crear gramáticas que puedan emplearse en cualquier sistema de diálogo que extraiga vocabulario de bases de datos.

El proceso de creación de dichas gramáticas se lleva a cabo en tres pasos:

i) Extracción de información de las bases de datos. Las gramáticas se generan antes de que el proceso de RAH se lleve a cabo.

ii) Construcción de las gramaticales a partir de los datos extraídos. Los datos se disponen en

formato JSGF (Java Speech Grammar Format) o ABNF (Augmented Backus-Naur Form).

iii) Actualización de las gramáticas con los cambios efectuados en las bases de datos. Para asegurar que las gramáticas están actualizadas permanentemente incluso cuando cambia el contenido de las bases de datos, nuestra técnica emplea un mecanismo basado en "disparadores" (triggers) de bases de datos. Éstos se disparan cuando se producen cambios en los campos de las bases de datos a partir de los cuales se extraen las palabras para las reglas de las gramáticas.

El grupo de trabajo W3C Voice Browser especifica en [12] que las reglas debe ser re-definibles en tiempo de ejecución. Para ello, apuntan diversos mecanismos. Entre ellos, subrayamos la división del espacio de reglas en estáticas y dinámicas. Empleando la técnica CGBD, la estructura de frase se crea estáticamente conteniendo referencias a reglas dinámicas, que pueden almacenarse en el mismo fichero o en uno externo, como ilustra el siguiente ejemplo:

#ABNF 1.0 iso-8859-1;
root $consulta;
public $consulta = Me gustaría hablar con $profesor;
public $profesor = [Don | Doña] $<Path de la regla para el nombre> $<Path de la regla para el primer apellido> $<Path de la regla para el Segundo apellido >;

Empleando esta gramática, los usuarios pueden expresar su intención de hablar con un profesor diciendo su nombre y apellidos. Las reglas $nombre y $apellidos se construirían con el vocabulario extraído de una base de datos en la que se almacenasen los nombres de profesores. Por tanto, estas reglas pueden incluirse en el mismo archivo donde se encuentra toda la gramática o crearse en ficheros independientes como en el ejemplo anterior.

3.1. Implementación

La herramienta de Generación Automática de Gramáticas (GAG), que desarrollamos en un trabajo previo ([4]), realiza la creación de gramáticas de forma automática. Para este trabajo, le hemos añadido la funcionalidad de la técnica CGBD empleando PHP, HTML, JavaScript y PostGreSQL. La herramienta se emplea como un módulo adicional del sistema de diálogo oral UAH (que describimos en la sección 4.1). La herramienta dispone de una interfaz sencilla de utilizar para permitir al diseñador del sistema escoger los campos de la base de datos que se emplearán para extraer el vocabulario. El proceso para crear reglas empleando esta herramienta se divide en tres etapas:

- En primer lugar, la herramienta GAG solicita al usuario el nombre de la base de datos, el nombre del ordenador servidor, el nombre de usuario y la contraseña para acceder a la base de datos donde se encuentra almacenada la información. También solicita el nombre de la regla de gramática y el tipo de gramática a crear (JSGF o ABNF).
- En segundo lugar, se visualizan los campos de las tablas de la base de datos en un menú *drop-down*. Los campos aparecen en el menú en el mismo orden en que se encuentran en la tabla correspondiente, pero el diseñador puede seleccionarlos en cualquier orden para crear la regla de la gramática. Cuando se selecciona un campo, un número aparece automáticamente junto al campo indicando el orden de selección. Por ejemplo, en una aplicación académica el diseñador del sistema podría seleccionar el nombre de profesor y seguidamente su apellido obteniendo la regla: <profesor> = ("Sergio García"|…| "Miguel Moreno"). Alternativamente, podría seleccionar ambos casos en orden inverso, es decir, primero el apellido y después el nombre, en cuyo caso la regla obtenida sería: <profesor> = "García Sergio" |…| "Moreno Miguel").
- Por ultimo, después de la selección de campos el diseñador debe introducir el nombre del archivo donde desea almacenar la regla de gramática. Puede elegir, además, entre crear la regla como parte de una gramática existente en su mismo fichero, o bien, crear un único fichero con una nueva gramática cuya única regla sea esa.

La herramienta GAG implementa el mecanismo de disparadores CGBD para mantener las reglas de las gramáticas actualizadas con los últimos cambios realizados en las bases de datos. Para ello, al final de la creación de cada gramática el diseñador del sistema indica si desea que se actualice automáticamente el vocabulario en la regla con cambios en la base de datos. En caso afirmativo, se crean dinámicamente disparadores que se activan cuando los valores correspondientes se actualizan, borran o insertan. Por ejemplo, si se crea una regla para los nombres y apellidos de profesores, el disparador incluirá "María" como una nueva palabra en la regla si se introduce en el campo "Nombre" de la tabla "Profesor". También borra la palabra "María" de la regla si se borra de la tabla, y cambia "María" por "Mónica" si el campo se actualiza con este nuevo valor.

Como los disparadores indican los valores antiguos y nuevos del campo de la tabla, sólo se realizan los cambios pertinentes en las reglas de las diversas gramáticas en lugar de generarlas por completo de nuevo cada vez que ocurra cualquier cambio, lo que incluiría un retardo equivalente al de crearlas

dinámicamente (como se ha comentado en la sección 2.2). El proceso completo de actualización automática se ilustra en la Figura 2, donde se esquematiza una arquitectura distribuida en la que el RAH se realiza en un ordenador con que dispone de una tarjeta de interfaz telefónica Intel Dialogic D/41JCT-LS, mientras que el gestor de diálogo y la herramienta GAG se ejecutan en el ordenador servidor, mientras que las bases de datos se almacenan en un equipo diferente.

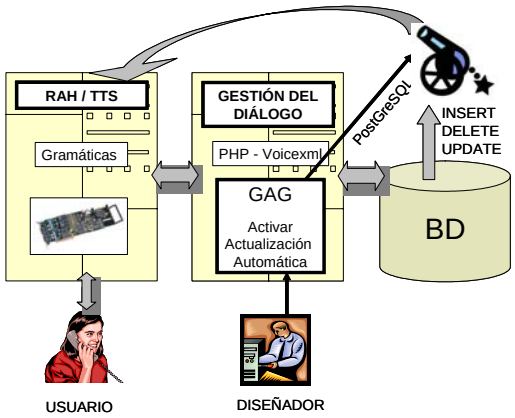


Figura 2. *Actualización automática de gramáticas con la herramienta GAG*

4. EXPERIMENTOS

Para obtener resultados experimentales usando la técnica CGBD hemos empleado el sistema de diálogo oral UAH (Universidad Al Habla) [13]. En esta sección describimos brevemente el sistema y a continuación discutimos los resultados obtenidos en la evaluación. Realizamos dos tipos de experimentos. En primer lugar, determinamos el grado de satisfacción de los usuarios conforme varía el tamaño del vocabulario, teniendo en cuenta únicamente el retardo introducido por el proceso de creación de las gramáticas. En segundo lugar, observamos el efecto producido por la técnica en la interacción de usuarios reales con el sistema UAH Nuestro objetivo, en este caso, es conocer el grado de aceptación del funcionamiento del sistema, de una forma global.

4.1. El sistema de diálogo UAH (Universidad al Habla)

El sistema UAH ha sido desarrollado en nuestro laboratorio para proporcionar acceso telefónico a la información disponible en la página web de nuestro departamento, permitiendo de esta manera el acceso a esta información a usuarios invidentes. El sistema se encuentra en funcionamiento desde Julio de 2005 en nuestro laboratorio prestando un servicio de información telefónica acerca de nuestro Departamento y asimismo, de forma limitada, acerca del proceso de auto-matrícula en nuestra Universidad.

En el momento de la escritura de este artículo el sistema está en proceso de evaluación. Para ello grabamos las llamadas de los usuarios, clasificamos las frases pronunciadas teniendo en cuenta los *prompts* del sistema, y almacenamos en ficheros las hipótesis de reconocimiento obtenidas (frases reconocidas). Además, obtenemos las opiniones de los usuarios acerca de distintos aspectos de la interacción con el sistema a través de formularios web[1]. De esta forma, hemos construido un corpus de pronunciaciones de frases que podemos analizar para obtener medidas objetivas del funcionamiento del sistema. Además, gracias a los formularios web podemos obtener también medidas subjetivas de evaluación del sistema [14].

Las gramáticas para RAH que usa el sistema se crean de formas diversas dependiendo del tipo de vocabulario y del momento de creación del mismo. En total empleamos cuatro métodos de creación, atendiendo a que el vocabulario sea conocido o desconocido en tiempo de diseño. Es decir, consideramos por una parte vocabulario que no cambia nunca, y por otra, vocabulario que puede ser cambiante por ser actualizado en una o varias bases de datos. En función del tipo de vocabulario realizamos la creación de las reglas de las gramáticas de forma estática o dinámica. La creación dinámica es preferible cuando el vocabulario de la gramática es conocido a priori, o en caso de ser desconocido, es lo suficientemente pequeño como para permitir la creación en tiempo de ejecución sin introducir un gran retardo. Empleando la técnica propuesta en este artículo, si el vocabulario es muy grande, usamos la herramienta GAG para crear automáticamente las reglas de las gramáticas. Para ello, la estructura de las frases se almacena en una gramática general que hace referencia a reglas externas que contienen el código generado dinámicamente (como se describe en la sección 3.1).

4.2. Resultados

En los experimentos realizados comparamos dos enfoques distintos para la creación de gramáticas para RAH basadas en reglas a partir de datos extraídos de bases de datos: creación dinámica (método tradicional) y creación usando la técnica propuesta (CGBD).

Para realizar una comparativa lo más ilustrativa posible de ambas técnicas, empleamos vocabularios que contenían desde 1 a 10^6 palabras (como los mostrados en la Figura 1). Sin embargo, el tamaño del vocabulario manejado en UAH es inferior a las 4.000 palabras, con lo que en muchas ocasiones se tuvieron que emplear vocablos de relleno generados automáticamente en la base de datos desde la cual se extrae el vocabulario para generar las reglas de las gramáticas de RAH.

Con la configuración anterior, pedimos a 30 usuarios de prueba que emplearan el sistema dos veces.

[1] El lector puede participar en la evaluación del sistema accediendo a la siguiente URL: http://teide.ugr.es/UAH/InstruccionesUAH.html

En primer lugar con la técnica de generación dinámica, pidiendo que expresaran su grado de satisfacción (en una escala de 1 a 5 puntos) acerca de la velocidad de interacción con el sistema; y en segundo lugar, empleando la técnica CGBD, que eligieran entre ésta y la primera estrategia utilizada. Los resultados de los experimentos anteriores son los mostrados en la Figura 3, donde se aprecia que el grado de satisfacción de los usuarios respecto a la creación dinámica de las gramáticas decae conforme aumenta el tamaño del vocabulario. En cambio, la tendencia es opuesta cuando se usa la técnica propuesta en este artículo.

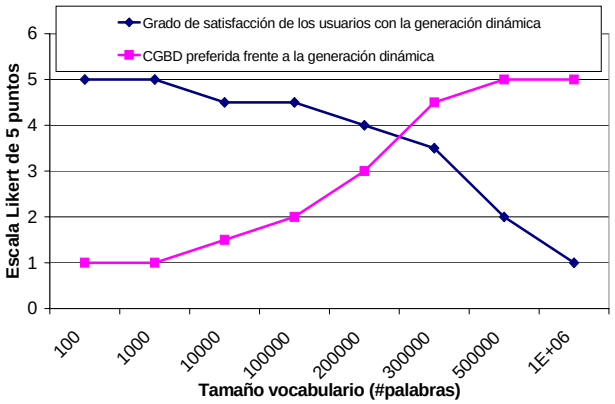


Figura 3. *Valoración de técnica la técnica CGBD frente a la creación dinámica de gramáticas*

Es claramente destacable la correspondencia entre el grado de satisfacción de los usuarios y las restricciones de tiempo comentadas en la sección 2.2. Como podemos observar en la Figura 3, para vocabularios menores de 10.000 palabras (que conllevan un tiempo de generación con la herramienta GAG menor de 0.1 segundos) los usuarios no son conscientes del retardo del sistema cuando se usa la técnica de generación dinámica. También podemos observar que existe un punto en el que ambas líneas se cruzan: cuando el vocabulario tiene un tamaño de unas 300.000 palabras. La figura muestra que a partir de este punto, el tamaño del vocabulario es lo suficientemente grande como para hacer que la técnica propuesta sea claramente preferible a la creación dinámica, dado el retardo causado por ésta última.

En general, al emplear el módulo GAG en el sistema UAH tenemos que, independientemente del tamaño del vocabulario, el grado de satisfacción de los usuarios con la interacción (medida en una escala de Likert de 5 puntos) es en el 78% de los casos, mayor o igual a 3 (como muestra la Figura 4).

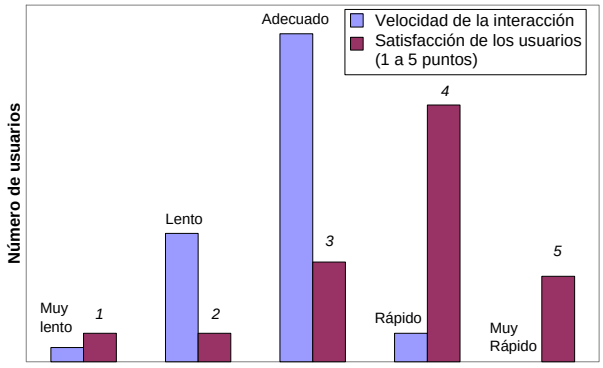


Figura 4. *Medidas subjetivas acerca de la velocidad de interacción y el grado de satisfacción de los usuarios.*

5. CONCLUSIONES Y TRABAJO FUTURO

Algunos sistemas de diálogo oral emplean gramáticas para RAH basadas en reglas cuyo vocabulario se extrae de bases de datos. Este método causa problemas si el tamaño de las bases de datos es grande, puesto que la interacción se ralentiza debido al tiempo requerido para crear las gramáticas en tiempo de ejecución. Para evitar este problema, otros sistemas de diálogo crean las reglas de las gramáticas de forma estática. Sin embargo, usando este método pueden surgir inconsistencias entre el vocabulario de las gramáticas y el contenido de las bases de datos conforme dicho contenido cambia.

Para resolver este problema de forma eficiente, hemos desarrollado una técnica que crea las reglas de las gramáticas antes de que comience el proceso de RAH, de forma tal que el vocabulario de dichas reglas se actualiza automáticamente conforme el contenido de las bases de datos cambia. Esta técnica es genérica y no está ligada a ningún tipo de base de datos o dominio de aplicación en particular. Por tanto, puede aplicarse para mejorar la velocidad de interacción de cualquier sistema de diálogo que utilice gramáticas para RAH basadas en reglas cuyo vocabulario deba ser extraído de bases de datos.

La técnica ha sido aplicada a un sistema de diálogo oral que proporciona información acerca de nuestro Departamento y, de forma limitada, acerca del proceso de auto-matrícula en nuestra Universidad.. Los resultados experimentales muestran que usando dicha técnica el grado de satisfacción de los usuarios que han participado en la evaluación es alto pues éstos consideran que la interacción se realiza a una velocidad adecuada.

Como trabajo futuro tenemos previsto realizar estudios más profundos de la apreciación de los usuarios respecto de la velocidad de interacción del sistema usando las dos técnicas de creación de gramáticas estudiadas: esto es, técnica tradicional (creación dinámica de gramáticas) frente a técnica CGBD. Nuestro interés se centrará, fundamentalmente, en determinar el efecto producido por el uso de una u

otra técnica junto con diversas técnicas de gestión de la iniciativa en el diálogo (esto es, iniciativa dirigida por sistema, dirigida por el usuario o mixta).

6. BIBLIOGRAFÍA

[1] D. Griol, F. Torres, L.F. Hurtado, E. Sanchis, E. Segarra, “Different approaches to the dialogue management in the DIHANA project”, SPECOM 2005

[2] S. Seneff, J. Polifroni, "Dialogue management in the mercury flight reservation system", Proc. ANLP-NAACL Sat. Workshop, Seattle, 2000.

[3] A. Ndiaye, P. Gebhard, M. Kipp, M. Klesen, M. Schneider, W. Wahlster, “Ambient Intelligence in Edutainment: Tangible Interaction with Life-Like Exhibit Guides”, Artificial Intelligence, Vol. 3814, pp. 104-113, 2005.

[4] Z. Callejas, R. López-Cózar, “Nueva Técnica de Generación Automática de Gramáticas para Sistemas de Diálogo”, Procesamiento del lenguaje natural, vol 35, pp. 205 - 212

[5] P. Truillet, O. Grisvard, B. Goujon, ”SCOPE – CARE II Innovative WP3 –R3 – Model of English”, European Organisation for the Safety of Air Navigation, 2004.

[6] J. Schalkwyck, L. Hetherington, E Story, “Speech recognition with Dynamic Grammars Using Finite-State Transducer”, Proc. Eurospeech, 1969-1972. Geneva, Switzerland, 2003.

[7] J. Nielsen, “Usability Engineering”, Ed. Morgan Kaufmann, San Francisco, 1994.

[8] L. Cerrato, “A comparison between feedback strategies in human-to-human and human-machine communication”, Proc. ICSLP, 557-560. Denver, USA, 2002.

[9] K. Mäkelä, E. Salonen, M. Turunen, J. Hakulinen, R. Raisamo, ”Evaluating the User Interface of a Ubiquitous Computing system Doorman”, Proc Ubicomp, Atlanta, USA, 2001.

[10] S. Möller, “Quality of Telephone-Based Spoken Dialogue Systems”, Ed. Springer, New York, 2005.

[11] M.F. Mc Tear, “Spoken Dialogue Technology: Toward the Conversational User Interface”, Ed. Springer, 2004.

[12] W3C Working Draft 23 Dec 1999, Grammar Representation Requirements for Voice Markup Languages. Available in http://www.w3.org/TR/voice-grammar-reqs/

[13] Z. Callejas, R. López-Cózar, “Implementing Modular Dialogue Systems: A Case of Study”, Proc. ASIDE, Aalborg, Denmark, 2005.

[14] R. López-Cózar, M. Araki, “Spoken, Multilingual and Multimodal Dialogue Systems: Development and Assessment”, Ed. Wiley, England, 2005.

PERSPECTIVAS DE LA TRADUCCIÓN AUTOMÁTICA CASTELLANO-GALLEGO MEDIANTE TÉCNICAS ESTADÍSTICAS Y POR TRANSFERENCIA

Gonzalo Iglesias Iglesias[1], *Leandro Rodríguez Liñares*[2], *Eduardo Rodríguez Banga*[3], *Francisco León Campillo Díaz* [3] *y Francisco Méndez Pazó*[3]

Centro Ramón Piñeiro para la Investigación en Humanidades (Santiago), E.T.S. Informática de Ourense (Universidad de Vigo), E.T.S. Telecomunicaciones (Universidad de Vigo)

RESUMEN

En esta comunicación presentamos nuestros trabajos en traducción automática entre las lenguas gallega y castellana. Los esfuerzos se centran en las dos principales líneas en que se divide este campo: traducción estadística y por transferencia.

1 INTRODUCCIÓN

En los últimos años, el avance de las tecnologías informáticas ha permitido la proliferación de productos basados en tecnologías del habla y del lenguaje natural. Entre las herramientas más útiles y revolucionarias que han aparecido figuran los traductores automáticos, que juegan un papel fundamental en la nueva forma de comunicarse que ha supuesto la aparición de Internet.

Especialmente importantes son, en este campo, los esfuerzos invertidos en el desarrollo de esta tecnología para las lenguas minoritarias, ya que amplía su visibilidad y constituye una garantía para su supervivencia.

En España podemos encontrar iniciativas de traducción automática. Podemos destacar, entre otros, productos ya maduros como el *OpenTrad Apertium* [3] y *Marie* [5].

Una buena introducción a la traducción automática puede encontrarse en [7]. En este artículo nos centraremos en la *traducción basada en reglas* o por transferencia y la *traducción estadística*.

Los traductores por transferencia se basan en la idea de que un conocimiento lingüístico lo suficientemente extenso de las dos lenguas involucradas debería permitir, mediante reglas, una traducción correcta.

La transferencia entre las estructuras de los lenguajes de origen y de destino puede tener lugar a niveles muy distintos de análisis lingüístico. El triángulo de Vauquois ilustra esta idea. En la base de este triángulo se situaría la denominada traducción directa (traducción palabra a palabra sin información lingüística contextual). En el vértice superior del triángulo tendríamos una representación conceptual (sintáctico, semántico, pragmático) de los idiomas, denominada interlingua. Entre estos dos extremos se encuentra la traducción por transferencia a distintos niveles, según la información empleada para crear las reglas entre lengua origen y lengua destino.

La *traducción estadística* ha cobrado auge desde la década de los noventa, a partir de los trabajos en el centro T.J. Watson de IBM ([1][2]). El modelo de traducción estadística más utilizado se basa en una analogía con un sistema de comunicaciones en que un mensaje codificado en la lengua destino se transmite a través de un canal ruidoso, con lo que se obtiene la versión distorsionada en el lenguaje origen. De acuerdo con este modelo, un sistema de traducción entre los lenguajes origen y destino equivale a decidir en base al mensaje recibido cuál es la secuencia de palabras transmitidas que con mayor probabilidad generaron dicho mensaje.

En esta comunicación presentamos dos traductores de castellano a gallego. Uno, basado en reglas; el segundo es un traductor estadístico. Los dos están aún en fase de desarrollo. El artículo se organiza de la siguiente manera: en la sección 2 describiremos el traductor basado en reglas; en la sección 3, se describe nuestro decodificador estadístico, inspirado en *Marie* [5]. Posteriormente, presentaremos algunos resultados relevantes y líneas futuras de trabajo.

2 TRADUCTOR AUTOMÁTICO BASADO EN REGLAS

La figura 1 representa las distintas etapas del traductor basado en reglas en el que el análisis se realiza frase a frase. En primer lugar, a cada frase del idioma origen se le aplica un preprocesado, con objeto de obtener el texto a traducir sin problemas de formato.

A continuación, en la segunda etapa se realiza un análisis de la frase para obtener las categorías morfosintácticas de cada palabra.

El análisis morfosintáctico mediante reglas y diccionarios no produce resultados unívocos. De ahí que sean necesarios criterios de desambiguación. Para este propósito se usan reglas lingüísticas fiables y un decisor estadístico basado en un modelo contextual y un modelo léxico.

El análisis morfosintáctico integra la lematización, que se puede entender como un proceso de abstracción de las variantes morfemáticas de la palabra mediante ciertos criterios o convenios. Así, el lema deducido es otra palabra que mantiene el valor semántico.

Una vez obtenido el lema del lenguaje origen se realiza la traducción al correspondiente lema del idioma

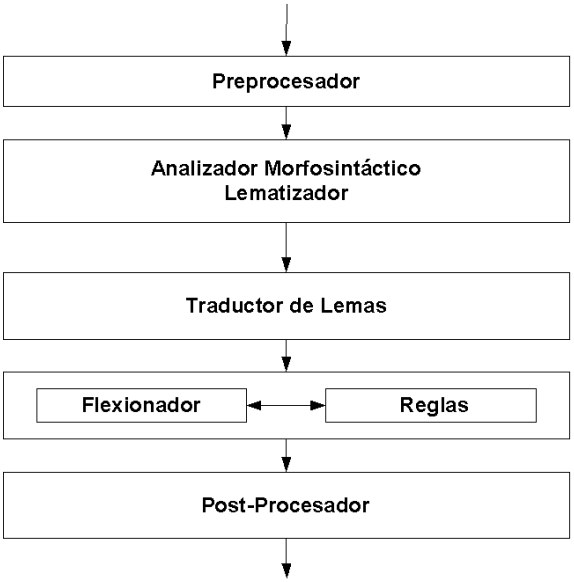


Figura 1. *Arquitectura del traductor.*

destino con la ayuda de un diccionario de traducción.

Finalmente el lema traducido es flexionado de acuerdo a características como género y número, por ejemplo, si se trata de un sustantivo, adjetivo o determinante. El proceso de flexión está supervisado por el módulo de reglas, que permite realizar además las transformaciones estructurales necesarias en la frase.

La traducción finaliza con un post-procesado, que devuelve el texto traducido en el formato inicial.

A continuación detallamos más detenidamente el funcionamiento de estos módulos.

2.1 El analizador morfosintáctico

Se trata de una etapa clave, ya que determina en gran medida el correcto funcionamiento del sistema. El analizador morfosintáctico empleado en este caso para el gallego y el castellano fue el desarrollado para nuestro conversor texto-voz Cotovía [10]. En la figura 3 se describe su estructura. En primer lugar, se asigna a cada palabra de la frase el conjunto de categorías que le pueden corresponder, por medio de diccionarios y reglas morfológicas sencillas.

A continuación, se reduce la ambigüedad aplicando un pequeño conjunto de reglas lingüísticas altamente fiables que solucionan los casos más sencillos. Posteriormente, se traducen las categorías a un conjunto de etiquetas más sencillo, y sobre ellas se emplea un modelo estadístico que combina la información resultante de una parte contextual y otra léxica para encontrar la secuencia de categorías morfosintácticas más probable para la frase de entrada. Finalmente se traducen las categorías resultantes a sus correspondientes en el conjunto de etiquetas original.

En los siguientes apartados se explican con más detalle las partes más importantes de dicho analizador.

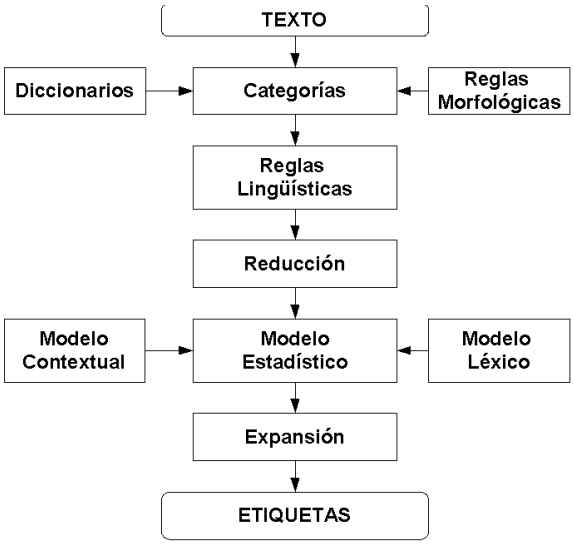


Figura 2. *Arquitectura del analizador morfosintáctico.*

2.1.1 El modelo contextual

El modelo contextual busca la categoría más probable para una palabra en función de las categorías de las palabras que la rodean. En este caso se recurre a la utilización de n-gramas de categorías gramaticales (n=5), determinando la categoría de cada palabra mediante el n-grama más probable cuando éste se halla centrado en dicha palabra.

2.1.2 El modelo léxico

El modelo léxico considera la probabilidad de que cada palabra concreta pertenezca a una categoría gramatical, independientemente del contexto que la rodee. Para calcular dichas probabilidades de forma fiable se necesitaría un corpus de entrenamiento excesivamente grande, por lo que se optó por el uso de *clases de ambigüedad* [6], ya que han demostrado ser una aproximación razonable [10] cuando no se dispone de suficiente texto etiquetado.

Esta estrategia consiste en agrupar palabras del corpus de entrenamiento en función del conjunto de

categorías que, a priori, se les pueden asignar. Por ejemplo, una clase de ambigüedad podría ser artículo-preposición-nombre. Dicha clasificación reduce de forma considerable el tamaño del espacio considerado, lo que supone que la estimación de las probabilidades sea más fiable.

2.1.3 El conjunto reducido de etiquetas

El número de categorías consideradas influye notablemente en el tamaño del corpus etiquetado que es preciso para obtener una estimación fiable de las probabilidades de los modelos contextual y léxico. Por otra parte, a medida que se vayan considerando menos categorías, el análisis generado irá perdiendo en poder descriptivo. La solución adoptada en este caso para evitar dicho problema consiste en considerar dos conjuntos diferentes de etiquetas: uno principal, y otro reducido.

El conjunto de etiquetas principal es más descriptivo e incluye información sobre género, número, persona, tiempos verbales, tipos de adverbios, tipos de determinantes..., mientras que el conjunto reducido agrupa las categorías del principal según su comportamiento gramatical en la oración (por ejemplo, artículos determinados e indeterminados se incluyen en la misma clase), respetando la información de género y número en aquellos casos en que tenga sentido.

De esta manera, el grupo de categorías principal puede ser tan amplio y descriptivo como se desee, debido a que en el modelo estadístico se construye sobre el conjunto de categorías reducido, que consta únicamente de 49 etiquetas [10]. Se requiere así de menor cantidad de texto etiquetado para el entrenamiento, lo que permite trabajar con un modelo de lenguaje más pequeño y manejable sin afectar a la capacidad descriptiva del análisis realizado.

2.1.4 El lematizador

El analizador morfosintáctico integra tanto en castellano como en gallego un lematizador. Antes de aplicarse el modelo estadístico, el problema de análisis y lematización se aborda por dos vías: mediante un diccionario de palabras y mediante un diccionario de derivaciones.

2.1.4.1 Diccionario de palabras.

Los diccionarios de palabras del analizador contienen información sobre el lema, directa o indirectamente, junto con sus posibles categorías. Por ejemplo, a un sustantivo como *actriz* el diccionario le asigna directamente el lema *actor*. Por otra parte, cualquier verbo constituye un buen ejemplo de búsqueda indirecta. Así, si en el análisis nos encontramos con la palabra *comieras*, se ha de determinar primero si esta palabra puede ser verbo, eliminando las desinencias y pronombres enclíticos, si los hay, y buscando la raíz *com-* en la lista de raíces verbales. Además, de nuestro diccionario de modelos de conjugación se obtiene que esta raíz está relacionada con el verbo *comer* mediante el modelo regular de la segunda conjugación. Un diccionario de desinencias proporciona la relación entre *-ieras* y dicho modelo regular y, como resultado, la categoría buscada: segunda persona singular del pretérito de subjuntivo.

2.1.4.2 Diccionario de derivaciones.

En caso de que no se pueda analizar y lematizar mediante los diccionarios de palabras, se acude al diccionario de derivaciones, que contiene información sobre los morfemas que permiten derivar una palabra de otra y cubre las ausencias de los diccionarios previos Así, se puede determinar típicamente si una palabra es un diminutivo, un aumentativo o un peyorativo; si la palabra en cuestión es un adverbio terminado en *-mente* o si se trata de una forma sustantivada de un verbo, etcétera.

El empleo del diccionario de derivaciones implica también la utilización del diccionario de palabras. Es decir: para que se considere una forma derivada debe verificarse la existencia de la palabra de la que procede.

2.1.5 Resultados

En [10] se realiza una descripción pormenorizada de resultados. Como muestra, en la tabla 1 presentamos los resultados del análisis sobre un texto en gallego. Algunas características de este texto se resumen en la tabla 2. La columna *ambiguas* se refiere a palabras con inicialmente más de dos etiquetas de categorías gramaticales asignadas; *etiquetas* representa el número medio de etiquetas por palabra; por último, *etiquetas ambiguas* representa el número medio de etiquetas en palabras donde inicialmente hay ambigüedad.

	Categoría	***Género***	***Número***
Texto	97.42%	97.96%	98.90%

Tabla 1. *Tasas de acierto del análisis.*

	Palabras	***Ambiguas***	***Etiquetas***	***Etiquetas Ambiguas***
Texto	24321	7528	1.68	3.08

Tabla 2. *Complejidad del texto de prueba.*

Por lo que respecta al castellano, todavía no se ha realizado una evaluación formal ya que el corpus disponible no es aún lo bastante amplio. En breve dispondremos de un corpus con una cobertura adecuada

y esperamos que los resultados serán similares a los obtenidos para gallego. Evidentemente, que el analizador morfosintáctico en castellano no sea aún todo lo preciso que debiera tiene una incidencia negativa en el traductor.

2.2 Diccionario de traducción

El traductor consta de un diccionario de lemas en sentido castellano → gallego (unas 20.000 entradas).

2.3 Flexionador

Se basa en los mismos diccionarios del analizador, optimizados adecuadamente para esta tarea. La flexión permite, a partir de un lema, generar la o las palabras que cumplan las características deseadas como pueden ser género, número, persona o tiempo verbal.

2.4 Módulo de reglas

El módulo de reglas procesa las palabras teniendo en cuenta la información morfosintáctica del entorno. De esta manera se pueden realizar las transformaciones necesarias para el idioma destino, en este caso el gallego. Este módulo controla también el flexionador; siempre a través de reglas en un diccionario.

La notación es bastante intuitiva y flexible. Cada regla consta de dos campos obligatorios: *condición* y *acción.* La acción tiene lugar si se verifica la condición, que puede contener uno o más elementos separados por espacios. Cada elemento de la condición puede ser una palabra, una categoría o un lema con una categoría. La *acción* indica la transformación que sufren las palabras involucradas. Una regla para transformar el antepretérito de subjuntivo castellano en pretérito de subjuntivo en gallego podría formularse de la siguiente manera (condición y acción se separan con una coma):

haber/VPS VT0SM,%1/VPS**

Esta regla busca el verbo *haber* en la forma pretérito de subjuntivo (*VPS*). La regla especifica que a continuación del verbo tiene que existir un participio (*VTOSM*). De cumplirse la condición, se sustituye con el lema del participio (referenciado con *%1*), flexionado en este caso a pretérito de subjuntivo en gallego.

De esta forma, el módulo de reglas recorre cada palabra de la frase y decide si se dan las condiciones en cada caso para aplicar alguna regla. Si es así, la aplica y, si no hay más reglas aplicables, avanza hasta la siguiente palabra. El orden en que se definen las reglas es relevante, ya que los cambios efectuados por una pueden ser modificados por otra.

Con el módulo de reglas desarrollado se resuelven adecuadamente muchos de los problemas más frecuentes que plantea la tarea de traducción. Exponemos a continuación algunos de los casos más relevantes.

2.4.1 Contracciones

Mientras que en castellano actual sólo se emplean dos contracciones, el número de contracciones en gallego es muy elevado. Así son muy frecuentes contracciones entre preposiciones y determinantes, preposiciones y pronombres, etc., además de los problemáticos conglomerados de pronombres átonos que según en qué caso van en posición enclítica o proclítica respecto al verbo. Como ejemplo sencillo, *de estas* en castellano debe traducirse a *destas* en gallego, que se podría especificar simplemente por la regla:

de estas, destas

Pero si deseamos resumir en una regla los determinantes demostrativos de la primera persona, escribiremos:

de este/DD,deste/CDD**

El * de la regla propaga en este ejemplo el género y el número desde la condición hasta la acción: con ello se resuelven cuatro casos con la misma regla.

2.4.2 Transformación de tiempos verbales

En gallego no existen los tiempos verbales compuestos, por lo que los tiempos compuestos del castellano deben transformarse en el tiempo gallego más adecuado.

A modo de ejemplo, las siguientes dos reglas: la primera resuelve la transformación de antecopretérito de indicativo en castellano, tiempo compuesto, a antepretérito en gallego, que es una forma simple. La segunda resuelve la regla de antepresente de indicativo a pretérito de indicativo.

haber/VCP VT0SM,%1/VAP**
haber/VR VT0SM,%1/VP**

Y de nuevo, el * propaga número y persona de la condición a la acción.

El típico problema de la perífrasis *ir a + infinitivo* se puede resolver también con la siguiente regla:

ir/V a VN,%0 %2

2.4.3 Enclisis

Los pronombres átonos en gallego sufren a menudo el proceso de enclisis con el verbo; salvo que alguna

partícula lo impida, como puede ser un pronombre interrogativo o un exclamativo. Por el contrario, en castellano es usual que los pronombres átonos vayan delante del verbo. Este hecho se controla nuevamente mediante una serie de reglas. Por ejemplo, para traducir adecuadamente formas del tipo *me dijiste* a *dixéchesme* basta con la siguiente regla:

me/PA V,%1 +me

Esta regla busca pronombre atóno *me* seguido de un verbo y lo sustituye por el mismo verbo seguido del pronombre. El símbolo + indica que debe realizarse además una operación de fusión con la palabra anterior. La fusión revisa la acentuación de la nueva palabra (*dixéchesme)*.

2.4.4 Expresiones y otros problemas.

De lo expuesto en las subsecciones precedentes es fácil deducir que las reglas permiten resolver la traducción de expresiones hechas o el artículo del posesivo en gallego.

Además, ofrecen una solución parcial para los casos en que el sustantivo en castellano y su correspondiente traducción en gallego tienen distinto género.

2.5 Preprocesador y postprocesador

Estos dos módulos se ocupan de mantener el formato de entrada en el texto de salida. Además de texto plano y sus características típicas (tabulaciones, retornos de carro, ciertos signos de puntuación, etc.), reconoce el formato *xml/html,* permitiendo un filtrado de datos no traducibles, bien sean las propias etiquetas, texto comentado o, típicamente, código *javascript*. De esta manera se mantiene íntegro el formato y la funcionalidad de cualquier *página web*.

3 TRADUCTOR ESTADÍSTICO

La traducción estadística parte de la idea, poco intuitiva lingüísticamente, de que la traducción puede ser vista como la transformación de una frase de un lenguaje a otro mediante un proceso estocástico. Por tanto, según este enfoque, traducir una frase f en el lenguaje origen consistirá en buscar una frase e en el lenguaje destino que maximice la fórmula:

$$\arg\max_{e} p(f/e) * p(e)$$

donde $p(f/e)$ representa el modelo de traducción y $p(e)$ corresponde al modelo de lenguaje destino.

Mientras que para entrenar el modelo de lenguaje destino es suficiente con un conjunto de texto de tamaño considerable en la lengua destino, para entrenar el modelo de traducción hace falta un corpus bilingüe revisado y alineado a nivel de frase.

3.1 Corpus de entrenamiento

Disponemos de un corpus de entrenamiento bilingüe creado a partir de texto periodístico. La cifra total de palabras presentes en este corpus es superior a un millón. Hemos utilizado el algoritmo LCS para alinear el corpus y rechazar pares de frases que no cumplen determinados umbrales de parecido.

Como se ha comentado con anterioridad, es deseable que el corpus bilingüe esté revisado manualmente. En nuestro caso, a falta de una revisión exhaustiva, hemos realizado una validación informal de material extraído del corpus de manera aleatoria, obtieniendo un resultado satisfactorio.

Por motivos de disponibilidad de material, hemos usado el material en lengua gallega presente en este corpus para la estimación del modelo de lenguaje destino.

3.2 Entrenamiento de los modelos

De entre todas las alternativas para la estimación del modelo de traducción, los denominados modelos IBM 1-5 [2] son probablemente los más populares. Estos modelos son válidos para lenguas origen y destino muy dispares, ya que la fertilidad o el reordenamiento de frases está contemplado. En este caso, el entrenamiento se realiza en fases de complejidad creciente, siendo el resultado de cada fase el punto de partida para la siguiente.

Nosotros hemos usado GIZA++ [8] para realizar el alineamiento del corpus de entrenamiento bilingüe, para a continuación extraer las unidades de traducción (tuplas) mediante el método expuesto en [4]. Para entrenar los modelos de N-gramas, hemos utilizado el conjunto de herramientas SRILM [9]. Estas herramientas también las hemos utilizado para la obtención del conjunto de N-gramas correspondiente al modelo de lenguaje destino.

	Modelo de lenguaje destino	***Modelo de traducción***
Unigramas	64858	74213
Bigramas	468925	484251
Trigramas	156250	166739

Tabla 3. *Número de N-gramas de los modelos.*

En la tabla 3 se pueden ver las características de los conjuntos de N-gramas obtenidos, tanto para el modelo de traducción como para el modelo de lenguaje destino.

3.3 Decodificador

Para la traducción, se utilizó una herramienta de elaboración propia que realiza una búsqueda basada en el algoritmo de Viterbi.

A partir de los N-gramas del modelo de traducción, se recorre la frase en el lenguaje origen de izquierda a derecha de tal modo que se exploran las posibles alternativas de traducción. Para evitar la aparición de un número desmesurado de posibilidades, en cada paso se realiza una poda N-best, quedándose el decodificador con las N mejores posibilidades. En nuestros experimentos, N=20 fue un valor que resultó ser adecuado para alcanzar un compromiso entre tamaño de las estructuras de datos y exhaustividad de la exploración.

El modelo de lenguaje destino es aplicado en nuestro traductor sobre las frases candidatas obtenidas con el modelo de traducción.

El algoritmo de búsqueda da lugar a traducciones monótonas, ya que no permite cubrir partes de la frase origen de manera desordenada. De todas maneras, la proximidad lingüística entre las lenguas origen y destino (castellano y gallego) hace que los reordenamientos lejanos sean innecesarios.

El decodificador fue implementado mediante una arquitectura cliente/servidor en el lenguaje de programación Java. La utilización de Java conlleva las ventajas de facilidad de programación (tanto algorítmica como de interfaces gráficas) y portabilidad, pudiendo ser utilizado en distintas plataformas o a través de Internet (como applet o servicio web).

A priori podría pensarse que la utilización de Java iba a reducir las prestaciones del decodificador; sin embargo, las pruebas realizadas demostraron que, una vez lanzado el servidor, los tiempos de ejecución para n=20 eran sólo ligeramente superiores a programas implementados en lenguaje C.

4 CONCLUSIONES Y LÍNEAS FUTURAS

En este artículo hemos presentado nuestros trabajos en traducción automática entre gallego y castellano. Son muchas las mejoras pendientes de realizar en ambos traductores, que todavía están en fase de desarrollo.

Además de todos los aspectos concretos que mejorarán las prestaciones de ambos traductores, en este momento, sólo se realiza una traducción castellano a gallego. En breve plazo desarrollaremos versiones que realicen traducción gallego → castellano y más a medio plazo entre estas dos lenguas e inglés.

5 BIBLIOGRAFÍA

[1] P. Brown,J. Cocke, S.A. Della Pietra, V.J. Della Pietra., F. Jelinek, J. Lafferty, R. Mercer y P. Roossin. "A statistical approach to machine translation". Computational Linguistics, vol 16-2, pp. 79-85. 1990.

[2] P. Brown, V.J. Della Pietra, S.A. Della Pietra y R. Mercer. "The Mathematics of Statistical Machine Translation: Parameter Estimation Computation". Computational Linguistics,vol 19-2, pp. 263-311. 1993.

[3] A.M. Corbí-Bellot, Mikel L. Forcada, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, Iñaki Alegria, Aingeru Mayor and Kepa Sarasola. "An open-source shallow-transfer machine translation engine for the romance languages of Spain." Proceedings of the European Association for Machine Translation, 10th Annual Conference, Budapest, pp. 79-86. 2005.

[4] J.M. Crego, J.B. Mariño, A. Gispert. "Finite-state-based and Phrase-based Statistical Machine Translation." Proc. of the 8th Int. Conf. on Spoken Language Processing, ICSLP'04, pp. 37-40. 2004.

[5] J.M. Crego, ,J.B. Mariño, Adrià Gispert. "An Ngram-based Statistical Machine Translation Decoder." 9th European Conference on Speech Communication and Technology. Interspeech, Lisboa. 2005.

[6] D. Cutting,,J. Kupiec, J. Pedersen y P. Sibun. "A practical part-of-speech tagger." Proceedings of the 3rd Conference on Applied Natural Language Processing (ANLP), pp. 133–140. 1992

[7] Ismael García. *Traducción Automática Estadística: Modelos de Traducción basados en Máxima Entropía y Algoritmos de Búsqueda.* Tesis Doctoral. Universidad Politécnica de Valencia, Departamento de Sistemas Informáticos y Computación. 2003

[8] F.J. Och, "An Efficient Method for Determining Bilingual Word Classes." Ninth Conf. of the Europ. Chapter of the Association for Computational Linguistics. pp. 71-76, EACL'99, Bergen, Norway. 1999.

[9] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit". Proc. Intl. Conf. Spoken Language Processing, Denver, Colorado. 2002.

[10] F. Méndez, F. Campillo, E. R. Banga y E. F. Rei, "Análisis morfológico estadístico en lengua gallega". Procesamiento del lenguaje natural nº 31, pp 71-76, 2003.

PROTOTIPO DE TRADUCCIÓN DE VOZ A LENGUA DE SIGNOS ESPAÑOLA

E. Ibáñez, A. Huerta, R. San-Segundo, L.F. D'Haro, F. Fernández, R. Barra
Grupo de Tecnología del Habla. Universidad Politécnica de Madrid

RESUMEN

Este artículo presenta los primeros experimentos de traducción de voz a Lengua de Signos Española (LSE). El sistema desarrollado está orientado aun dominio concreto de aplicación: frases pronunciadas por un funcionario que atiende a las personas que desean obtener o renovar el DNI (Documento Nacional de Identidad) o el pasaporte. El sistema traduce las frases pronunciadas por el funcionario a lengua de signos. El sistema está formado por 3 módulos principales: un reconocedor de voz, un módulo de traducción de lenguaje natural (traduce secuencias de palabras en secuencia de signos), y finalmente un módulo de representación de los signos mediante un agente animado en 3D. Los mejores resultados obtenidos ofrecen un error en la secuencia de signos del 25.2% y un BLEU (BiLingual Evaluation Understudy) del 0.64.

1. INTRODUCCIÓN

La lengua de signos presenta una gran variabilidad entre diferentes países e incluso en diferentes áreas de un mismo país. Por esta razón, desde 1960, la lengua de signos ha sido objeto de estudio en Estados Unidos [1][2][3], en Europa [4][5], Africa [6] y Japón.

En España, durante los últimos 20 años ha habido varias propuestas para normalizar la Lengua de Signos Española (LSE), pero ninguna de ellas ha sido aceptada por las personas sordomudas. Desde su punto de vista, estas propuestas tienden a limitar la flexibilidad de la lengua de signos. En 1991 M.A. Rodríguez [8] llevó acabo un análisis detallado de las diferencias entre la lengua de signos utilizada por las personas sordomudas y las propuestas de estandarización. Este estudio constituye uno de los principales estudios de la LSE y ha sido una referencia importante en este trabajo.

La traducción de habla natural ha sido y sigue siéndolo hoy en día una de las principales áreas de investigación en proyectos como C-Star, ATR. Vermobil, Eutrans, LC-Star, PF-Star y TC-Star. Todos estos proyectos, a excepción de TC-Start, son proyectos enfocados a la traducción de habla en dominios restringidos (como por ejemplo una agencia de viajes o turismo) con vocabularios de tamaño medio. Los mejores sistemas de traducción están basados en soluciones estadísticas [9], incluyendo técnicas basadas en ejemplos [10], traductores de estados finitos [11] y otras soluciones basadas en datos. Los importantes progresos conseguidos en traducción de habla se deben principalmente a factores como la aparición de medidas de error [12], mejora de la eficiencia de los algoritmos de entrenamiento [13], desarrollo de modelos dependientes del contexto [10] y algoritmos de generación eficientes[14],

El proyecto europeo eSIGN (Essential Sign Language Information on Government Networks) [15] constituye uno de los esfuerzos más importantes para el desarrollo de herramientas de apoyo a la generación de contenido para personas sordomudas. En este proyecto, uno de los principales resultados ha sido la generación de un avatar en 3D (VGuido) (capaz de representar cualquier signo de la lengua de signos) y un entorno visual para el desarrollo de animaciones fácilmente. Estas herramientas están orientadas a traducir contenido web. La lengua de signos es la lengua principal para muchas personas sordas (algunas de las cuales no comprenden correctamente la lengua escrita puesto que son personas sordas prelocutivas). El resultado del proyecto eSIGN está funcionando en páginas web oficiales de Alemania, Países Bajos y el Reino Unido.

En los últimos años ha aumentado el interés de varios grupos por la traducción automática en lengua de signos desarrollándose varios prototipos: basados en ejemplos [16], basados en reglas [17], frases completas [18] o soluciones estadísticas [19][20]. Este artículo presenta unos primeros experimentos sobre uno de los primeros sistemas de traducción de voz a signos y el primero para Lengua de Signos Española (LSE).

2. DESCRIPCIÓN DEL SISTEMA

El sistema presentado, cuyo diagrama de módulos se puede observar en la Figura 1, consta de varias partes.

1. Un reconocedor de voz: que se encarga de traducir la voz en una secuencia de palabras. Este módulo permite reconocer habla en lenguaje natural (habla continua) e independiente del locutor. Para esta aplicación el vocabulario posible contiene 485 palabras, lo que permite conseguir tasas de aciertos por encima del 90%.

2.- Un módulo de traducción automática: traduce la secuencia de palabras en una secuencia de signos de la LSE. Se ha utilizado un sistema basado en reglas de traducción elaboradas por una persona experta. Mediante este método, la traducción entre palabras y signos se realiza aplicando las reglas: las reglas van combinando palabras y generando signos, que a su vez se pueden recombinar con otros signos o palabras para formar la secuencia completa final.

3. Un módulo de representación de los signos: basado en un agente animado en 3D. Se dispone de dos agentes animados, que podrán usarse de forma indistinta. Uno de ellos es 'VGuido', desarrollado en el proyecto eSIGN, y el segundo 'Sara' ha sido desarrollado en este trabajo. 'Sara' ha sido desarrollada atendiendo a las necesidades de perfeccionamiento de las partes del cuerpo determinantes a la hora de representar un signo de la lengua de signos (cabeza y brazos). Asimismo, se ha implementado una interfaz gráfica capaz de generar signos, posiciones y animaciones, que funciona como módulo de representación del sistema desarrollado, y también como sistema independiente de diseño de animaciones para cada signo, al servicio de un logopeda, con la posibilidad de incrementar el número de signos a realizar, creando bibliotecas específicas. Para ello se ha dotado al sistema gráfico de herramientas y funcionalidades de alto nivel que permiten desarrollar nuevas animaciones con cierta facilidad para el usuario.

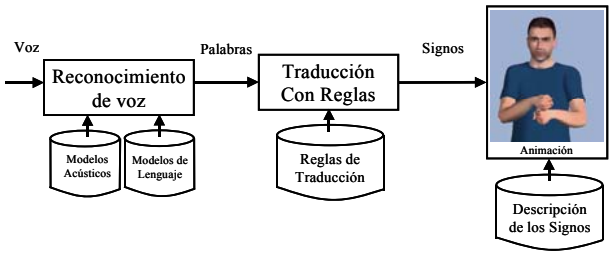


Figura 1. Diagrama de módulos del sistema desarrollado

3. DOMINIO DE APLICACIÓN

Los módulos de análisis semántico y de generación de signos de este sistema están diseñados para el dominio de aplicación de apoyo al servicio de solicitud/ renovación del DNI (Documento Nacional de Identidad) o pasaporte. Dicho proceso se gestiona desde el Ministerio del Interior, y por lo tanto, toda la información que se ha usado para el análisis de este proceso de solicitud y renovación del DNI se ha obtenido de la página oficial del Ministerio de Interior.

A partir de la información de dicha página se redactaron 201 frases (con 458 palabras diferentes) y sus respectivas traducciones (con 270 signos diferentes). Se ha mantenido por lo general la estructura de las frases originales, poniendo mucho interés en generar diferentes niveles de uso, o grados de flexibilidad a la hora de realizar la traducción del español a la LSE. Si bien es cierto que, por lo general, se ha tratado de simular los hábitos de las personas sordomudas que prefieren dividir una misma oración en una serie de sentencias mucho más cortas y directas. Esto ha supuesto una reducción sustancial del tamaño de algunas de las oraciones compuestas, tal y como se muestra en el siguiente ejemplo:

Frase original: "El documento tramitado lo recogerá su titular o persona que autorice mediante la presentación del correspondiente resguardo, en la misma Oficina donde lo solicitó."

Algunas de las frases generadas a partir de la original:

- "Para recoger el denei tienes que ser el titular o tener su permiso."
- "Para recoger el denei deberá presentar el correspondiente resguardo."
- "Para recoger el denei debes ir allí donde lo solicitaste."

El sistema pretende, por lo tanto, conseguir que un funcionario de la administración del Estado que no conozca en absoluto la LSE pueda explicar a una persona sorda qué necesita para poder solicitar o renovar su DNI, cuales son las utilidades de dicho documento, dónde puede viajar él, y todo tipo de información relativa al DNI o el pasaporte.

4. RECONOCEDOR DE VOZ

Para la elaboración de los módulos de reconocimiento de voz y de traducción se ha utilizado la plataforma para el desarrollo de aplicaciones con voz del Grupo de Tecnología del Habla (GTH) denominada SERVIVOX [21].

El módulo de reconocimiento de voz convierte las frases pronunciadas en lenguaje natural a texto. Para la implementación de dicho módulo se ha usado un reconocedor de habla continua, desarrollado en el GTH. Dicho reconocedor utiliza modelos de Markov continuos, modelo de lenguaje 2-gram, 16 modelos de ruido para hacer frente a los ruidos que puedan producirse (ruidos ambiente, ruidos del locutor como carraspeos, muletillas, etc.), y un diccionario de 458 palabras. La gramática necesaria para dicho reconocedor ha sido obtenida a partir de una lista con 201 frases del dominio de aplicación. Para crear dicha gramática se usó un programa de creación de gramáticas desarrollado en el GTH.

Para comprobar el efecto del Reconocimiento Automático del Habla (RAH) en el sistema se procedió a la lectura de las frases del dominio de aplicación para su reconocimiento automático. Así se obtuvieron por escrito tanto las transcripciones como las frases reconocidas.

5. TRADUCCIÓN DE TEXTO A SIGNOS

La estrategia de traducción considerada se basa en reglas de traducción desarrolladas por una persona experta. En este caso, las relaciones entre los signos y las palabras se definen manualmente. Las reglas van combinando palabras y generando signos, que a su vez se pueden recombinar con otros signos o palabras para dar lugar a la secuencia de signos final. El sistema de traducción por reglas implementado podría clasificarse como un híbrido entre la traducción directa (posee un tipo de reglas que traducen una palabra por su signo equivalente en LSE), y la traducción indirecta con transferencia de tipo sintáctico (se ha creado un modelo

sintáctico del español y otro de la LSE para hacer corresponder uno con otro).

El primer paso para la realización de este sistema de traducción ha sido un análisis pormenorizado de la LSE, y la correspondiente extracción e implementación de las reglas sintácticas. En este proceso fue fundamental la referencia [8]. En segundo lugar se lleva a cabo la traducción según las reglas sintácticas anteriormente elaboradas. Dicho proceso se ha descompuesto en las siguientes etapas:

1.- Categorización. En primer lugar se asigna una o varias categorías sintáctico-semánticas a cada una de las palabras que forman el vocabulario. La codificación utilizada para dichas categorías contiene una serie de prefijos y sufijos, tal y como se muestra en la Figura 2.

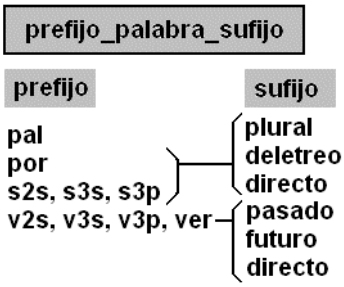


Figura 2. Sistema de categorías

Lo fundamental de este sistema de categorías es la capacidad que proporciona para discernir si la palabra es un verbo (v2s, v3s, v3p, ver) o un sustantivo (por, s2s, s3s, s3p), si puede ser un sujeto (s2s, s3s, s3p) o no (por), información del tiempo verbal (presente por defecto, pasado o futuro), del número (singular por defecto, plural), si es una palabra de traducción directa (directo) o si por el contrario debe sufrir un proceso de deletreo (deletreo) como es el caso de los nombres propios o las siglas. El resto de palabras (que no se clasifican en los grupos anteriores) se codifican con el prefijo pal y sin sufijo.

2.- Implementación de las reglas antes de eliminar basura. Antes de eliminar las palabras basura (palabras que no aportan información en el proceso de traducción) se aplica un conjunto importante de las reglas desarrolladas. Estas reglas se pueden clasificar en 4 tipos que se describen a continuación:

- Una palabra genera un signo las palabras que se traducen según esta regla están categorizadas como 'por_palabra_directo', y su traducción es PALABRA (misma palabra en mayúsculas).
- Varias palabras generan un signo para este tipo de reglas se han usado las primitivas 'reescribeX', donde X puede ser 2, 3, 4, 5, 6 y 7, cuyo cometido es la reagrupación de los X bloques o tokens (que pueden ser palabras, signos o signos intermedios) iniciales en un único bloque final.
- Una palabra da lugar a varios signos para estas reglas se han usado las primitivas 'desdoblaX' (donde X puede ser 2, 3, 4, 5 ó 6), desdobla2cond (que permite extraer información del siguiente bloque), traduce_deletreos (que obtiene a partir de una palabra los signos necesarios para su deletreo), suj_omitidos (que añade un bloque para el sujeto de la frase, imprescindible en LSE), y plur_pas_fut (que añade los signos necesarios para indicar que la palabra siguiente está en plural, o que el verbo tiene información de pasado o futuro). Este tipo de reglas pretende dividir un bloque inicial en tantos como sean necesarios según la regla de traducción aplicada.
- Varias palabras se traducen en varios signos para esta regla se implementaron las primitivas 'reescribeXaY' (cuando el número de bloques finales es distinto del inicial) y 'reescribeXaX' (en el caso de que se obtengan tantos bloques como había inicialmente). Este tipo de reglas surge de la necesidad de resolver las situaciones en las que una expresión en el lenguaje natural genera una expresión gestual (y no un único signo).

3.- Eliminación de las palabras basura. Eliminación de las palabras basura o con prefijo pal_. Se eliminan los bloques que no aportan información a la frase a traducir: o bien fueron categorizados como basura inicialmente o nunca han formado parte de una regla anterior y por tanto no son útiles para la traducción.

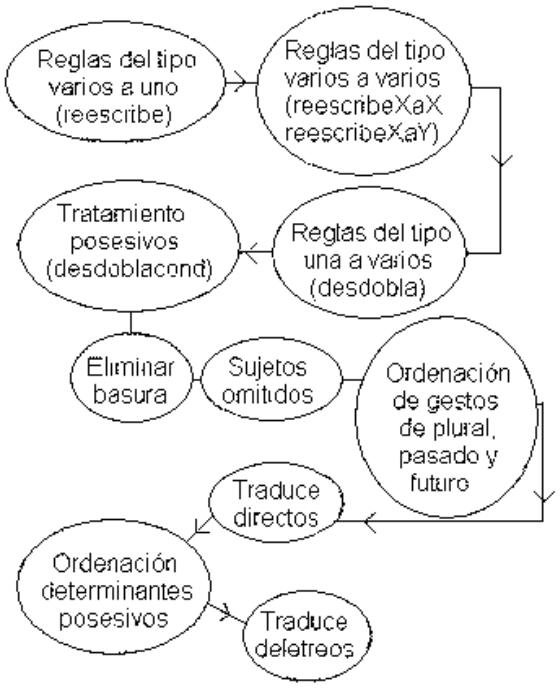


Figura 3. Esquema de las reglas y primitivas utilizadas en el proceso de traducción.

4.- Ordenación sintáctica de los signos/gestos y generación de la secuencia final. Esta ordenación se lleva a cabo gracias a las primitivas plur_pas_fut (que crea los signos indicativos de plural, pasado y futuro y coloca delante del sustantivo el signo indicativo de

plural), pon_primero (que trata de colocar al comienzo de la frase los signos indicativos de PASADO y FUTURO) y posesivos (que coloca los signos que signan el posesivo detrás del signo que les sigue).

El proceso completo de traducción queda reflejado en la Figura 3.

7. RESULTADOS

Para evaluar el funcionamiento de este primer prototipo, se ha calculado los porcentajes de signos correctos, signos insertados (en relación con la referencia de traducción), signos borrados (según la referencia), y signos sustituidos. Para obtener estos porcentajes, la secuencia de signos resultado de la traducción se compara con la referencia (o referencias y hay varias alternativas de traducción) mediante un algoritmo de programación dinámica (o distancia de Levenshtein) considerando costes iguales para todos los tipos de errores. En esta evaluación se consideraron las 201 frases descritas en el apartado 3. Con estos porcentajes es posible calcular el porcentaje de signos con error (PSE) de forma similar a como se calcula la tasa de error en un sistema de reconocimiento de habla. En esta evaluación también se ha calculado la medida BLEU (BiLingual Evaluation Understudy). Esta medida es menos estricta que la PSE y está siendo muy utilizada en traducción automática [12]. Los resultados finales obtenidos se presentan en la Tabla 1.

TIPO	BLEU	PSE (%)	INS (%)	BOR (%)	SUB (%)
TEXTO	0.83	15.4	4.2	9.8	2.4
VOZ	**0.64**	**25.2**	**5.9**	**16.8**	**2.5**

Tabla 1. ***Resultados finales obtenidos con el sistema basados en reglas***

Como se puede observar, el módulo de traducción presenta un alto porcentaje de borrados en comparación con el resto de errores. Este hecho es debido a la estrategia de las reglas: el reconocedor de voz comete errores que hace que varios patrones de palabras no aparezcan (para activar algunas de las reglas definidas) y dichas reglas no generen correctamente los signos resultado de dicha traducción.

8. REPRESENTACIÓN DE LOS SIGNOS

Para la representación de los signos se presentan dos avatares. Uno de ellos, es 'VGuido', el avatar desarrollado en el marco el proyecto europeo eSIGN. En este trabajo se ha desarrollado otro avatar, centrándonos en dotar de especial detalle las partes más importantes a la hora de representar los signos: los brazos y la cabeza. Para ello se ha realizado un sistema basado en jerarquías de huesos, implementado sobre OpenGL, resultando un prototipo denominado 'Sara'. Se comentará el caso de Sara en detalle. El sistema de huesos para los brazos consta de dos esqueletos (con 17 huesos cada uno), uno para cada brazo. La cabeza es un objeto en formato .3DS (3D StudioMax), que se importa como objeto aparte. A continuación se explican en detalle cada una de estas partes del avatar. El sistema completo se muestra en la Figura 4.

1.- El sistema de brazos: el sistema de brazos consta de dos esqueletos, uno para cada brazo. Cada esqueleto (cada brazo) tendrá un hueso padre, u origen del cual dependerán los demás. Asimismo, cada hueso tendrá un padre, que será el hueso inmediatamente superior, y uno o varios hijos. De esta forma se asegura un movimiento realista y proporcionado. Los huesos están cubiertos con una malla, a modo de piel. Dicha malla se asocia al sistema mediante un cálculo ponderado de pesos. Cada punto de la malla estará asociado a un hueso concreto (si está claro cuál es el hueso más cercano). Es decir, se tienen dos tipos de casos: puntos de la piel que deben moverse con un hueso concreto y puntos de la piel que, por estar en zonas críticas, deben moverse según el movimiento de dos huesos (zonas de unión entre huesos). Una vez definidos los esqueletos, se describe el sistema de movimiento.

Cada hueso está definido por las traslaciones tridimensionales (x,y,z) de su punto de origen (el punto de unión con el hueso padre), y las rotaciones tridimensionales. A la hora de representar un movimiento, se ha comprobado que este queda perfectamente definido por las rotaciones parciales de cada uno de los huesos únicamente. Cada vez que se mueve un hueso, se actualizan sus rotaciones, haciendo por extensión que se actualicen las rotaciones de los huesos hijos. Al final se adapta la malla (piel) a la nueva situación de los huesos.

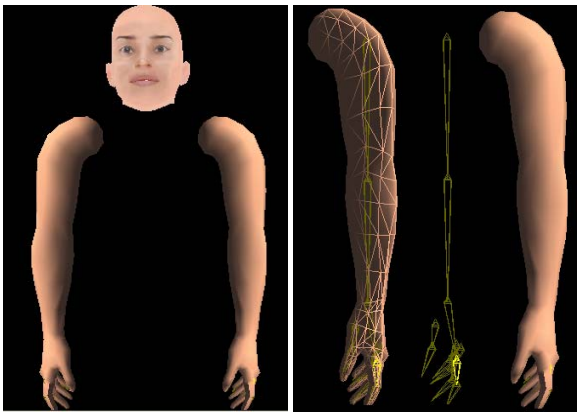


Figura 4. Sistema de representación de animaciones: 'Sara', y sistema jerárquico de huesos, con textura asociada.

2.-La cabeza: es un objeto importado en el sistema en formato *.3DS. No dispone de tanta adaptabilidad como los brazos, solamente es capaz de realizar movimientos, sin expresiones. No obstante, es suficiente para lo que se quiere implementar, ya que su principal función es acompañar o servir de referente a los movimientos de los brazos.

En el procedimiento para diseñar una animación se realizan las siguientes acciones, por este orden: creación de posiciones (parciales y/o totales), creación de animaciones.

a) *Creación de posiciones:* la herramienta desarrollada ofrece la posibilidad de crear y guardar posiciones, para cargarlas posteriormente. Se puede trabajar con posiciones del esqueleto completo, o posiciones únicamente de las manos. En ambos casos, el procedimiento es el mismo: se guardan en un archivo los datos referentes a las rotaciones de los huesos implicados (todo el esqueleto o solamente los huesos de las manos). De esta forma se pueden guardar posiciones concretas, posiciones comunes en la representación de la lengua de signos, para posteriormente combinarlas entre sí, formando la secuencia de signos deseada.
b) *Creación de animaciones:* para la creación de animaciones, se define un array de posiciones. Las animaciones se crearán como secuencias de posiciones. Para ello, se definen tres tipos de posiciones: 0, 1 y 2. Las posiciones por defecto serán de tipo 0, que significa que no han sido especificadas. Las posiciones tipo 1 son las que se establecen específicamente con una posición concreta. Las posiciones inicial y final serán siempre de tipo 1 por defecto. La creación de las posiciones intermedias se basará en la interpolación lineal, entre las posiciones definidas de tipo 1. Las posiciones obtenidas de esta interpolación se denominarán de tipo 2. La animación consistirá en la representación sucesiva de cada posición en pantalla. La creación de animaciones resulta sencilla habiendo dotado al sistema de utilidades como: guardar, cargar y 'resetear' posiciones, creación y borrado de interpolaciones (incluso para animaciones ya guardadas), y previsualización de resultados. Las animaciones podrán ser de 60 o 120 posiciones, pudiendo fácilmente conmutar de un modo a otro según la velocidad del signo.

Mediante la generación de animaciones, se realiza la creación de la base de datos que permitirá representar los signos provenientes del módulo de traducción descrito anteriormente. Asimismo, se presenta la posibilidad de crear animaciones nuevas, de forma que el sistema funcione de manera independiente, al servicio de un logopeda que necesite ampliar la biblioteca de signos animados.

Para la presentación en pantalla, el sistema dispone de un generador de videos, de forma que al mostrar una animación, o una secuencia de animaciones, se crea automáticamente un video que puede ser utilizado como representación en cualquier sistema, dotando así a la aplicación de mayor adaptabilidad.

Para la representación de grandes secuencias de signos, se ha definido la estructura llamada: superframe. Un superframe se formará concatenando animaciones con un número de posiciones definido (60 o 120). Estas secuencias se unirán mediante pequeños fragmentos de animación, de 10-20 posiciones tipo 0, de forma que cuando se tenga definida la cadena de animaciones se pueda interpolar la secuencia, creando movimientos de unión entre los distintos signos a representar.

Para todas estas funciones, se ofrece una interfaz gráfica completa, con todas las opciones clasificadas de forma sencilla e intuitiva para que el usuario lo utilice fácilmente (Figura 5).

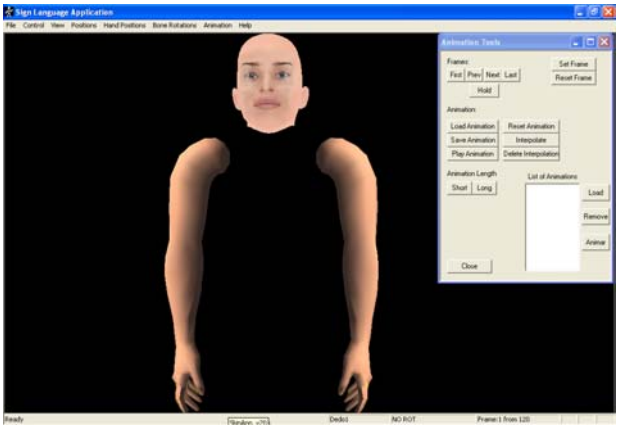


Figura 5. Interfaz gráfica de la aplicación para la representación de signos mediante el avatar 'Sara'.

En segundo avatar considerado es VGuido. VGuido es un agente animado en 3D que se ha incorporado en el sistema como un control ActiveX. Este agente animado ha sido desarrollado en el proyecto eSIGN y permite representar la secuencia de gestos concatenando diferentes animaciones asociadas a cada uno de los gestos (Figura 6).

Figura 6. Agente animado VGuido

Cada una de las animaciones asociadas a los signos a representar se definen utilizando un script en lenguaje SiGML (Signing Gesture Markup Language). SiGML es una aplicación del lenguaje XML. Mediante este lenguaje se definen aspectos relacionados con la posición de las manos, la cabeza, la velocidad de signado, el tamaño del gesto, etc. La descripción de

estos aspectos está basada en la notación HamNoSys [21]. Se dispone de un entorno visual que permite generar los ficheros de tipo SiGML para cada signo.

9. CONCLUSIONES

Este artículo ha presentado los primeros experimentos de un prototipo de traducción de voz a Lengua de Signos Española (LSE) en un dominio restringido. Este dominio consiste en las frases pronunciadas por un funcionario cuando atiende a las personas que desean obtener o renovar el DNI o el pasaporte. El sistema de traducción implementado está formado por 3 bloques: un reconocedor de voz para decodificar la voz en palabras. Posteriormente un módulo de traducción de lenguaje natural genera la secuencia de signos que correspondería con la salida del reconocedor. Finalmente, un módulo con un agente animado en 3D anima la secuencia de signos resultado de la traducción. Como resultados de la traducción se ha obtenido un 25.2% PSE (Porcentaje de Signos Erróneos) y un BLEU (BiLingual Evaluation Understudy) igual a 0.64.

10. AGRADECIMIENTOS

Los autores quieren agradecer al consorcio eSIGN (Essential Sign Language Information on Government Networks) el permiso para utilizar el eSIGN Editor y el avatar en 3D desarrollado por ellos (VGUIDO). Este trabajo ha sido financiado por los siguientes proyectos de investigación TINA (UPM-CAM. REF: R05/10922), ROBINT (DPI2004-07908-C02) y EDECAN (TIN2005-08660-C04).

11. BIBLIOGRAFÍA

[1] Stokoe, W., (1960). "Sign Language structure: an outline of the visual communication systems of the American deaf". Studies in Linguistics. Buffalo.

[2] Christopoulos, C. Bonvillian, J., (1985). "Sign Language". J. of Communication. Disorders, no 18 1-20.

[3] Pyers J.E., (2006) "Indicating the body: Expression of body part terminology in American Sign Language". Language Sciences. Available online 4 January 2006.

[4] Hansen, B., (1975). "Varieties in Danish Sign Language". Sign Language Studies, 8: 249-256.

[5] Kyle, J., (1981). "British Sign Language". Special Education, 8: 19-23. 1981.

[6] Penn, C., Lewis, R., and Greenstein, A., (1984). "Sign Language in South Africa". South African Disorder of Communication, 31: 6-11. 1984.

[7] Notoya, M., Suzuki, S., Furukawa, M., and Umeda, R., (1986). "Method and Acquisition of Sign Language in Profoundly Deaf Infants". Japan Journal of Logopedics and Phoniatrics, 27: 235-243. 1986.

[8] Rodríguez, M.A. (1991). "Lenguaje de signos" Phd Dissertation. CNSE and ONCE. Madrid. Spain.

[9] Och J., H. Ney. (2002). "Discriminative Training and Maximum Entropy Models for Statistical Machine Translation". Annual Meeting of the Ass. ACL, Philadelphia, PA, pp. 295-302. 2002.

[10] Sumita E., Y. Akiba, T. Doi et al. (2003). "A Corpus-Centered Approach to Spoken Language Translation". Conf. Of Ass. for Computational Linguistics (ACL) Hungary. pp171-174.

[11] Casacuberta F., E. Vidal. (2004). "Machine Translation with Inferred Stochastic Finite-State Transducers". Comp. Linguistics, V30, nº 2, 205-225.

[12] Papineni K., S. Roukos, T. Ward, W.J. Zhu. (2002) "BLEU: a method for automatic evaluation of machine translation". 40th Annual Meeting of the ACL, Philadelphia, PA, pp. 311-318. 2002.

[13] Och J., H. Ney. (2003). "A systematic comparison of various alignment models". Computational Linguistics, Vol. 29, No. 1 pp. 19-51, 2003.

[14] Koehn P., F.J. Och D. Marcu. (2003) "Statistical Phrase-based translation". Human Language Technology Conference 2003 (HLT-NAACL 2003), Edmonton, Canada, pp. 127-133, May 2003.

[15] http://www.sign-lang.uni-hamburg.de/eSIGN/

[16] S. Morrissey and A. Way. 2005. An example-based approach to translating sign language. In Workshop Example-Based Machine Translation (MT X–05), pages109–116, Phuket, Thailand, September.

[17] M. Huenerfauth. 2004. A multi-path architecture for machine translation of English text into American Sign language animation. HLT-NAACL, Boston, MA, USA.

[18] S.J. Cox, M. Lincoln, J Tryggvason, M Nakisa, M. Wells, Mand Tutt, and S Abbott. TESSA, a system to aid communication with deaf people. In ASSETS 2002, pages 205-212, Edinburgh, Scotland, 2002.

[19] J. Bungeroth and H. Ney: Statistical Sign Language Translation. In Workshop on Representation and Processing of Sign Languages, LREC 2004, 105-108.

[20] M. Zahedi, D. Keysers, and H. Ney: Appearance-Based Recognition of Words in American Sign Language. In IbPRIA 2005, 2nd Iberian Conference on Pattern Recognition and Image Analysis, Estoril, Portugal, June 2005.

[21] J.M. Montero: Manual del SERVIVOX. GTH-DIE. UPM, 1998.

[22] Prillwitz, S., R. Leven, H. Zienert, T. Hanke, J. Henning, et-al. (1989). Hamburg Notation System for Sign Languages – An introductory Guide. International Studies on Sign Language and the Communication of the Deaf, Volume 5. Institute of German Sign Language and Communication of the Deaf, University of Hamburg, 1989.

AUTOMATIC VS. MANUAL TOPIC SEGMENTATION AND INDEXATION IN BROADCAST NEWS

R. Amaral[(1,2,3)], *H. Meinedo*[(1,3)], *D. Caseiro*[(1,3)], *I. Trancoso*[(1,3)], *J. Neto*[(1,3)] *

(1) Instituto Superior Técnico
(2) Instituto Politécnico de Setúbal
(3) L^2F - Spoken Language Systems Lab, INESC-ID
{ramaral,meinedo,dcaseiro,imt,jpn}@l2f.inesc-id.pt
http://www.l2f.inesc-id.pt

ABSTRACT

This paper describes the latest progress in our work on Broadcast News for European Portuguese. The central modules of our media watch system that matches the topic of each news story against the user preferences registered in the system are: audio pre-processing, speech recognition and topic segmentation and indexation. The main focus of the paper is on the impact of the errors made by the earlier modules in the last ones. This impact is in our opinion an essential diagnostic tool for the improvement of the overall pipeline system.

1. INTRODUCTION

In a media watch system dealing with selective dissemination of Broadcast News, there are several components whose performance influences the next stages. In the system developed at our lab, the first of these components is the audio pre-processing (APP) module which performs speech/non-speech classification, speaker segmentation, speaker clustering, and gender and background conditions classification. The second component is the automatic speech recognition (ASR) module that converts the segments classified as speech into text. The XML file produced by these earlier modules includes not only the transcribed text, but also additional information such as the segment duration, the acoustic background classification (e.g. clean/music/noise), the speaker gender and the identification of the speaker cluster. The third component is the topic segmentation (TS) module that takes as input this XML file and splits the broadcast news program show into constituent stories. Last in this pipeline of modules is the topic indexation (TI) module that assigns one or multiple topics selected from a thematic thesaurus, thus adding more metadata to the original XML file. The goal of this paper is to study the influence of the performance of the earlier modules on the next ones.

The use of a thematic thesaurus for indexation was introduced by RTP, the Portuguese public broadcast company, and our partner in the past European project ALERT. This thesaurus follows rules which are generally adopted within EBU (European Broadcast Union) and has been used by RTP since 2002 in its daily manual indexation task. It has a hierarchical structure that covers all possible topics, with 22 thematic areas in the first level, and up to 9 lower levels. In our system, we implemented only 3 levels, which are enough to represent the user profile information that we need to match against the topics produced by the indexation module.

This paper is structured into four main sections, each one devoted to one of the four modules. Rather than lumping all the results together, we will present them individually for each section, in order to be able to better compare the oracle performance of each module with the one in which all previous components are automatic. Before describing each module and the corresponding results, we shall describe the corpus that served as the basis for this study. The last section summarizes the main conclusions and future work.

2. THE EUROPEAN PORTUGUESE BN CORPUS

The European Portuguese Broadcast News corpus, collected in close cooperation with RTP, involves different types of news shows, national and regional, from morning to late evening, including both normal broadcasts and specific ones dedicated to sports and financial news. The corpus is divided into 3 main subsets:

- SR (Speech Recognition) - The SR corpus contains around 61h of manually transcribed news, collected during a period of 3 months, with the primary goal of training acoustic models and adapting the language models of our ASR module. The corpus is subdivided into training (51h), development (6h), and evaluation sets (4h).

* H. Meinedo was sponsored by a FCT scholarship (SFRH/BD/6125/2001). This work was partially funded by FCT projects POSI/PLP/47175/2002 and POSC/PLP/58697/2004. INESC-ID Lisboa had support from the POSI program of the "Quadro Comunitário de Apoio III". The order of the first two authors was randomly selected.

- TD (Topic Detection) - The TD corpus contains around 300h of topic labeled news, collected during the following 9 months. All the data was manually segmented into stories or fillers (short segments spoken by the anchor announcing important news that will be reported later), and each story was manually indexed according to the thematic thesaurus. The corresponding orthographic transcriptions were automatically generated by our ASR module.

- JE (Joint Evaluation) - The JE corpus contains around 13h, corresponding to the last two weeks of the collection period. It was fully manually transcribed, both in terms of orthographic and topic labels. All the evaluation work described in this paper concerns this corpus. 17% was manually classified as being in the F0 focus condition (planned speech, no background noise, high bandwidth channel, native speech), whereas 61% is in the F4 condition (speech under degraded acoustical conditions).

3. AUDIO PRE-PROCESSING

The APP module includes five separate components: three for classification (speech/non-speech, gender and background), one for speaker clustering and one for acoustic change detection. All components are model-based, making extensive used of feed-forward fully connected Multi-Layer Perceptrons (MLP) trained with the back-propagation algorithm on the SR training corpus [1].

The speech/non-speech module is responsible for identifying audio portions that contain clean speech, and audio portions that instead contain noisy speech or any other sound or noise, such as music, traffic, etc. This serves two purposes: first, no time will be wasted trying to recognize audio portions that do not contain speech; second, it reduces the probability of speaker clustering mistakes.

Gender classification distinguishes between male and female speakers and is used to improve speaker clustering. By clustering separately each gender class we have a smaller distance matrix when evaluating cluster distances which effectively reduces the search space. It also avoids short segments having opposite gender tags being erroneously clustered together. Background status classification indicates if the background is clean, has noise or music. Although it could be used to switch between tuned acoustic models trained separately for each background condition, it is only being used for topic segmentation purposes. All three classifiers share the same architecture: an MLP with 9 input context frames, a hidden layer with 300 sigmoidal units, and an output unit which can be viewed as giving a probabilistic estimate of the input frame being speech or non-speech. When the acoustic change detector hypothesizes the start of a new segment, the first 300 frames of that segment are used to calculate the speech/non-speech, gender and background classification. Each classifier computes the decision with the highest average probability over all the frames.

The main goal of the acoustic change detector is to detect audio locations where speakers or background conditions have changed. It uses a hybrid two-stage algorithm. The first stage generates a large set of candidate change points which in the second stage are evaluated to eliminate the ones that do not correspond to true speaker change boundaries [1]. The first stage uses two complementary algorithms. It evaluates in the cepstral domain the similarity between two contiguous windows of fixed length using the symmetric Kullback-Liebler distance [2]. This is followed by an energy-based algorithm that detects when the median drops bellow the long term average. The second stage uses an MLP classifier, with a large 300-frame input context of acoustic features (12^{th} order PLP plus log energy) and a hidden layer with 150 sigmoidal units.

The goal of speaker clustering is to identify and group together all speech segments that were uttered by the same speaker. After the acoustic change detector signals the existence of a new boundary and the classification modules determine that the new segment contains speech, the first 300 frames of the segment are compared with all the clusters found so far, for the same gender. The segment is merged with the cluster with the lowest distance, provided it falls bellow a predefined threshold. The distance is also computed by one of the two gender-specific MLP classifiers.

3.1. Audio Pre-Processing Results

Table 1 summarizes the results for the components of the APP module computed over the JE corpus. Speech/non-speech, gender and background classification results are reported in terms of percentage of correctly classified frames for each class and accuracy, defined as the ratio between the number of correctly classified frames and the total number of frames. In order to evaluate the clustering, a bi-directional one-to-one mapping of reference speakers to clusters was computed (NIST rich text transcription evaluation script). The Q-measure is defined as the geometrical mean of the percentage of cluster frames belonging to the correct speaker and the percentage of speaker frames labeled with the correct cluster. Another performance measure is the DER (Diarization Error Rate) which is computed as the percentage of frames with an incorrect cluster-speaker correspondence.

Generally, these APP results are quite good, except in terms of background classification, which is a rather difficult task, and DER, for which results around 10% have been reported, obtained with state of the art speaker identification techniques like feature warping and model adaptations [3]. We are currently working on reducing the number of clusters belonging to the same speaker.

Our system was compared against the best algorithms evaluated in [4] applied to a common database, having achieved similar results in most categories.

Speech/ Non-Speech		Speech 98.9	Non-speech 54.9	Accuracy 95.4
Gender		Male 96.5	Female 97.5	Accuracy 97.0
Background	Clean 77.4	Music 79.0	Noise 71.4	Accuracy 73.5
Clustering		Q 76.9	Q map 84.7	DER 29.8

Table 1. Audio Pre-Processing results.

4. AUTOMATIC SPEECH RECOGNITION

The second module in our pipeline system is a hybrid automatic speech recognizer [5] that combines the temporal modeling capabilities of Hidden Markov Models (HMMs) with the pattern discriminative classification capabilities of MLPs. The acoustic modeling combines phone probabilities generated by several MLPs trained on distinct feature sets: PLP (Perceptual Linear Prediction), Log-RASTA (log-RelAtive SpecTrAl) and MSG (Modulation SpectroGram). Each MLP classifier incorporates local acoustic context via an input window of 13 frames. The resulting network has two non-linear hidden layers with 1500 units each and 40 softmax output units (38 phones plus silence and breath noises). The vocabulary includes around 57k words. The lexicon includes multiple pronunciations, totaling 65k entries. The corresponding out-of-vocabulary (OOV) rate is 1.4%. The language model which is a 4-gram backoff model was created by interpolating a 4-gram newspaper text language model built from over 604M words with a 3-gram model based on the transcriptions of the SR training set with 532k words. The language models were smoothed using Knesser-Ney discounting and entropy pruning. The perplexity obtained in a development set is 112.9.

Our decoder is based on the Weighted Finite-State Transducer (WFST) approach to large vocabulary speech recognition [6]. In this approach, the search space is a large WFST that maps HMMs (or in some cases, observations) to words. This WFST is built by composing various components represented as WFSTs. In our case, the search space integrates the HMM/MLP topology transducer, the lexicon transducer and the language model one. Traditionally, this composition and subsequent optimization is done in an offline compilation step. A unique characteristic of our decoder is its ability to compose and optimize the various components of the system in runtime. A specialized WFST composition algorithm was developed [7] that composes and optimizes the lexicon and language model components in a single step. Furthermore, the algorithm can support lazy implementations so that only the fragment of the search space required in runtime is computed. This algorithm is able to perform true composition and determinization of the search space while approximating other operations such as pushing and minimization. This dynamic approach has several advantages relative to the static approach. The first one is memory efficiency, the specialized algorithm requires less memory than the explicit determinization algorithm used in the offline compilation step, moreover, since only a small fraction of the search space is computed, it also requires less runtime memory. This memory efficiency allows us to use large 4-gram language models in a single pass of the decoder. Other approaches are forced to use a smaller language model in the first pass and rescore with a larger language model. The second advantage is flexibility, the dynamic approach allows for quick runtime reconfiguration of the decoder since the original components are available in runtime and can be quickly adapted or replaced.

4.1. Confidence Measures

Associating confidence scores to the recognized text is essential for evaluating the impact of potential recognition errors. Hence, confidence scoring was recently integrated in the ASR module. In a first step, the decoder is used to generate the best word and phone sequence, including information about the word and phone boundaries, as well as search space statistics. Then, for each recognized phone, a set of confidence features are extracted from the utterance and from the statistics collected during decoding. The phone confidence features are combined into word level confidence features. Finally, a maximum entropy classifier is used to classify words as correct or incorrect. The word level confidence feature set includes various recognition scores (recognition score, acoustic score and word posterior probability [8]), search space statistics (number of competing hypotheses and number of competing phones), and phone log likelihood ratios between the hypothesized phone and the best competing one. All features are scaled to the $[0, 1]$ interval. The maximum entropy classifier [9] combines these features according to:

$$P(correct|w_i) = \frac{1}{Z(w_i)} exp[\sum_{i=i}^{F} \lambda_i f_i(w_i)] \quad (1)$$

where w_i is the word, F is the number of features, $f_i(w_i)$ is a feature, $Z(w_i)$ is a normalization factor and λ_i are the model parameters. The detector was trained on the SR training corpus. When evaluated on the JE corpus, an equal-error-rate of 24% was obtained.

4.2. ASR Results with Manual and Automatic Pre-Processing

Table 2 presents the word error rate (WER) results on the JE corpus, for two different focus conditions (F0 and all conditions), and in two different experiments: according to the manual pre-processing (reference classifications and boundaries) and according to the automatic preprocessing defined by the APP module.

The performance is comparable in both experiments with only 0.7% absolute increase in WER. This increase

	% WER	
APP	F0	All
Manual	11.3	23.5
Automatic	11.6	24.2

Table 2. Speech recognition results.

can be explained by speech / non-speech classification errors, that is, word deletions caused by noisy speech segments tagged by APP as non-speech, and word insertions caused by noisy "silence" segments marked by APP as containing speech. The other source for errors is related to the differences between manual and automatic sentence-like unit (SU) boundaries. Since the APP tends to create larger than "real" SUs, the problem seems to be in the language model which is introducing erroneous words (mostly function words), connecting different SUs. A qualitative analysis indicates the following types of error:

- Errors due to severe vowel reduction. Vowel reduction, including quality change, devoicing and deletion, is specially important for European Portuguese, being one of the features that distinguishes it from Brazilian Portuguese and that makes it more difficult to learn for a foreign speaker. It may take the form of (1) intra-word vowel devoicing; (2) voicing assimilation; and (3) vowel and consonant deletion and coalescence. Both (2) and (3) may occur within and across word boundaries. Contractions are very common, with both partial or full syllable truncation and vowel coalescence. As a result of vowel deletion, rather complex consonant clusters can be formed across word boundaries. This type of error is strongly affected by high speech rate. The relatively high deletion rate may be partly attributed to severe vowel reduction and affects mostly (typically short) function words.

- Errors due to OOVs. This affects namely foreign names. It is known that one OOV term can lead to between 1.6 and 2 additional errors [10].

- Errors in inflected forms. This affects mostly verbal forms (Portuguese verbs typically have above 50 different forms, excluding clitics), and gender and number distinctions in names and adjectives. It is worth exploring the possibility of using some post-processing parsing step for detecting and hopefully correcting some of these agreement errors. Some of these errors are due to the fact that the correct inflected forms are not included in the lexicon.

- Errors around speech disfluencies. This is the type of error that is most specific of the spontaneous speech, a condition that is fairly frequent in the JE corpus. The frequency of repetitions, repairs, restarts and filled pauses is very high in these conditions, in agreement with values of one disfluency every 20 words cited in [11]. Unfortunately, the training corpus for Broadcast News included a very small representation of such examples.

- Errors due to inconsistent spelling of the manual transcriptions. The most common inconsistencies occur for foreign names and for entries spelled both as separate words and as a single word.

These WER results are worse than the ones that are quoted for other languages, such English (less than 16% with Real-Time performance [12]), a fact that can be partly attributed to the reduced amount of training data for European Portuguese.

5. TOPIC SEGMENTATION

The goal of TS module is to split the broadcast news show into the constituent stories. This may be done taking into account the characteristic structure of broadcast news shows [13]. They typically consist of a sequence of segments that can either be stories or fillers. The fact that all stories start with a segment spoken by the anchor, and are typically further developed by out-of-studio reports and/or interviews is the most important heuristic that can be exploited in this context. Hence, the simplest TS algorithm is the one that starts by defining potential story boundaries in every transition non-anchor / anchor. In the next step, the algorithm tries to eliminate stories that are too short, because of the difficulty of assigning a topic with so little transcribed material. In these cases, the short story segment is merged with the following one with the same speaker and background. Other heuristics are also adopted to avoid too long stories spoken only by the anchor, which may in fact include more than one story without further developments.

The identification of the anchor is done on the basis of the speaker clustering information, as the cluster with the largest number of turns. A minor refinement was recently introduced to account for the cases where there are two anchors (although not present in the JE corpus).

5.1. Topic Segmentation Results with Manual and Automatic Prior Processing

The evaluation of the topic segmentation was done using the standard measures Recall (% of detected boundaries), Precision (% of marks which are genuine boundaries) and F-measure (defined as $2RP/(R+P)$). Table 3 shows the TS results, using the Recall, Precision, and F-measure metrics, as well the metric adopted in the 2001 Topic Detection and Tracking benchmark NIST evaluation, with the same cost values of miss and false alarms [14]. These results, together with the field trials we have conducted [15], show that boundary deletion is a critical problem. In fact, our very simple TS algorithm has several pitfalls: it fails when all the story is spoken by the anchor, without further reports or interviews, leading to a merge with the

next story; ii) it fails when the filler is not detected by a speaker / background condition change, also leading to a merge with the next story (19% of the program events are fillers); iii) it fails when there is a special anchor for a part of the broadcast (i.e. sports anchor), although in this case one could argue that all the stories are about the same generic topic; iv) it fails when the anchor(s) is not correctly identified.

APP	ASR	Recall	Precision	F-meas.	Cost
Manual	Manual	79.0	60.3	67.2	0.78
Manual	Auto	76.6	60.3	66.3	0.75
Auto	Auto	70.2	56.4	61.6	0.66

Table 3. Topic Segmentation results.

6. TOPIC INDEXATION

Topic identification is a two-stage process, that starts with the detection of the most probable top-level story topics and then finds for those topics all the second and third level descriptors that are relevant for the indexation.

For each of the 22 top-level domains, topic and non-topic unigram language models were created using the stories of the TD corpus which were pre-processed in order to remove function words and lemmatize the remaining ones. Topic detection is based on the log likelihood ratio between the topic likelihood $p(W/T_i)$ and the non-topic likelihood $p(W/\overline{T_i})$. The detection of any topic in a story occurs every time the correspondent score is higher than a predefined threshold. The threshold is different for each topic in order to account for the differences in the modeling quality of the topics.

In the second step, we count the number of occurrences of the words corresponding to the domain tree leafs and normalize these values with the number of words in the story text. Once the tree leaf occurrences are counted, we go up the tree accumulating in each node all the normalized occurrences from the nodes below [16]. The decision of whether a node concept is relevant for the story is made only at the second and third upper node levels, by comparing the accumulated occurrences with a predefined threshold.

6.1. Topic Indexation Results with Manual and Automatic Prior Processing

In order to conduct the topic indexation experiments we started by choosing the best threshold for the word confidence measure as well as for the topic confidence measure. The tuning of these thresholds was done with the development corpus in the following manner: the word confidence threshold was ranged from 0 do 1, and topic models were created using the correspondent topic material available. Obviously higher threshold values decrease the amount of automatic transcriptions available to train each topic. Topic indexation was then performed in the development corpus in order to find the topic thresholds corresponding to the best topic accuracy (91.9%). The use of these confidence measures led to rejecting 42.0% of the original topic training material.

Once the word and topic confidence thresholds were defined, the evaluation of the indexation performance was done for all the stories of the JE corpus, ignoring filler segments. The correctness and accuracy scores obtained using only the top-level topic are shown in Table 4, assuming manually segmented stories. Topic accuracy is defined as the ratio between the number of correct detections and the total number of topics, and topic correctness as the ratio between the number of correct detections minus false detections (false alarms) and the total number of topics. The results for lower levels are very dependent on the amount of training material in each of these lower level topics (the second level includes over 1600 topic descriptors, and hence very few material for some topics).

When using topic models created with the non-rejected keywords, we observed a slight decrease in the number of misses and an increase in the number of false alarms. We also observed a slight decrease with manual transcriptions, which we attributed to the fact that the topic models were built using ASR transcriptions.

APP	ASR	Correctness	Accuracy
Manual	Manual	91.5	91.3
Manual	Auto w/o conf	93.8	91.5
Manual	Auto w/ conf	94.1	91.7
Auto	Auto w/ conf	93.9	91.4

Table 4. Topic indexation results.

These results represent a significant improvement over previous versions [17], mainly attributed to allowing multiple topics per story, just as in the manual classification. A close inspection of the table shows similar results for the topic indexation with auto or manual APP. The adoption of the word confidence measure made a small improvement in the indexation results, mainly due to the reduced amount of data to train the topic models. The results are shown in terms of topic classification and not story classification.

Whereas one could find comparable results for topic segmentation in the TDT2001 evaluation program [14], the topic indexation task has no parallel, because it is thesaurus-oriented.

7. CONCLUSIONS AND FUTURE WORK

This paper studied the impact of earlier errors in the last modules of our pipelined BN media watch system. This impact is in our opinion an essential diagnostic tool for its overall improvement.

Our APP module has a good performance, while maintaining a very low latency for stream-based operation. The impact of its errors on the ASR performance is small

(0.7% absolute) when compared with the manual references. The greatest impact of APP errors is in terms of topic segmentation, given the heuristically-based approach that is crucially dependent on anchor detection precision.

Our ASR module also has a good performance, although the results for European Portuguese are not yet at the level of the ones for languages like English, where much larger amounts of training data are available. We believe that unsupervised training approaches will be very helpful in this context. Our current work in terms of ASR is focused on dynamic vocabulary adaptation, and processing spontaneous speech, namely in terms of dealing with disfluencies and sentence boundary detection.

The ASR errors seem to have very little impact on the performance of the two next modules, which may be partly justified by the type of errors (e.g. errors in function words and in inflected forms are not relevant for indexation purposes).

Topic segmentation still has several pitfalls which we plan to reduce for instance by exploring video cues. In terms of topic indexation, our efforts in building better topic models using a discriminative training technique based on the conditional maximum likelihood criterion for the implemented Naive Bayes classifier [18] have not yet been successful. This may be due to the small amount of manually topic-annotated training data.

In parallel with this work, we are also currently working on unsupervised adaptation of topic detection models, unsupervised training of better acoustic models and improving speaker clustering by using the latest state of the art speaker identification techniques.

8. REFERENCES

[1] H. Meinedo and J. Neto, "A stream-based audio segmentation, classification and clustering preprocessing system for broadcast news using ann models," in *Proc. Interspeech '2005*, Lisbon, Portugal, 2005.

[2] M. Siegler, U. Jain, B. Raj, and R. Stern, "Automatic segmentation, classification and clustering of broadcast news," in *Proceedings DARPA Speech Recognition Workshop*, 1997.

[3] X. Zhu et al., "Combining speaker identification and BIC for speaker diarization," in *Proc. Interspeech '2005*, Lisbon, Portugal, Sept. 2005.

[4] J. Zibert et al., "The COST278 broadcast news segmentation and speaker clustering evaluation - overview, methodology, systems, results," in *Proc. Interspeech '2005*, Lisbon, Portugal, Sept. 2005.

[5] H. Meinedo, D. Caseiro, J. Neto, and I. Trancoso, "Audimus.media: a broadcast news speech recognition system for the european portuguese language," in *Proc. PROPOR '2003*, Faro, Portugal, June 2003.

[6] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," in *ASR 2000 Workshop*, Sept. 2000.

[7] D. Caseiro and I. Trancoso, "A specialized on-the-fly algorithm for lexicon and language model composition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1281–1291, July 2005.

[8] D. Williams, *Knowing what you don't know: roles for confidence measures in automatic speech recognition*, Ph.D. thesis, Univ. Sheffield, UK, 1999.

[9] Adam L. Berger, Stephen Della Pietra, and Vincent J. Della Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[10] J.-L. Gauvain, L. Lamel, and G. Adda, "Developments in continuous speech dictation using the ARPA WSJ task," in *Proc. ICASSP '1995*, Detroit, USA, May 1995.

[11] E. Shriberg, "Spontaneous speech: How people really talk, and why engineers should care," in *Proc. Interspeech '2005*, Lisbon, Portugal, Sept. 2005.

[12] S. Matsoukas et al., "The 2004 BBN 1XRT recognition systems for english broadcast news and conversational telephone speech," in *Proc. Interspeech '2005*, Lisbon, Portugal, Sept. 2005.

[13] R. Barzilay, M. Collins, J. Hirschberg, and S. Whittaker, "The rules behind roles: Identifying speaker role in radio broadcast," in *Proc. AAAI 2000*, Austin, USA, July 2000.

[14] NIST Speech Group, "The 2001 topic detection and tracking (TDT2001) task definition and evaluation plan," 2001.

[15] I. Trancoso, J. Neto, H. Meinedo, and R. Amaral, "Evaluation of an alert system for selective dissemination of broadcast news," in *Proc. Eurospeech '2003*, Geneva, Switzerland, Sept. 2003.

[16] A. Gelbukh, G. Sidorov, and A. Guzmán-Arenas, "Document indexing with a concept hierarchy," in *Proc. NDDL 2001 - 1st Int. Workshop on New Developments in Digital Libraries*, Setúbal, Portugal, July 2001.

[17] R. Amaral and I. Trancoso, "Improving the topic indexation and segmentation modules of a media watch system," in *Proc. ICSLP '2004*, Jeju, Korea, Oct. 2004.

[18] C. Chelba, M. Mahajan, and A. Acero, "Speech utterance classification," in *Proc. ICASSP '2003*, Hong Kong, Apr. 2003.

EVALUACIÓN DE UN CORPUS DE DIÁLOGOS MEDIANTE UN GESTOR DE DIÁLOGO ESTOCÁSTICO

David Griol, Lluís F. Hurtado, Emilio Sanchis, Encarna Segarra

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València. E-46022 València, Spain
{dgriol,lhurtado,esanchis,esegarra}@dsic.upv.es

RESUMEN

El principal objetivo de este artículo es evaluar las características fundamentales de un corpus de diálogos mediante su uso en una aproximación estocástica para realizar la gestión del diálogo. El gestor se basa en la generación de la nueva respuesta del sistema a través de un proceso de clasificación mediante redes neuronales. En dicho proceso, se tiene en cuenta la información suministrada por el usuario a lo largo del diálogo y la última respuesta proporcionada por el sistema. El modelo de gestión se aprende automáticamente a partir de un corpus de aprendizaje etiquetado en términos de actos de diálogo.

Esta aproximación se ha aplicado para la elaboración de un gestor de diálogo estocástico en el marco del proyecto DIHANA, cuya tarea es proporcionar información sobre trayectos en tren en el territorio español.

En el artículo se describe la metodología propuesta y se presentan las diferentes experimentaciones llevadas a cabo para evaluar la influencia de las características más relevantes del corpus DIHANA en el funcionamiento del gestor.

1. INTRODUCCIÓN

Un tema de especial interés, y de especial dificultad, en el campo de las Tecnologías del Habla es el desarrollo de sistemas de diálogo entre las computadoras y sus usuarios humanos. Un sistema de diálogo es un sistema automático capaz de emular a un ser humano en un diálogo con otra persona, con el objetivo de que el sistema cumpla con una cierta tarea (dar una cierta información, o proporcionar ciertos servicios).

Usualmente, diferentes módulos suelen interactuar para llevar a cabo este objetivo final: deben reconocer las palabras pronunciadas por el usuario, comprender su significado, gestionar el diálogo, realizar el tratamiento de errores, acceder a las bases de datos y generar la respuesta oral.

En cuanto a la gestión del diálogo, a medida que los procesos de diseño, implementación y evaluación de las estrategias de gestión del diálogo se hacen cada vez más complejos, el interés de la comunidad científica se centra en la utilización de métodos estocásticos basados en el aprendizaje de un modelo a partir de un corpus de datos correctamente etiquetado. Los modelos estadísticos permiten modelar la variabilidad de comportamientos del usuario, presentando el inconveniente de su dependencia con respecto a la calidad y cantidad de las muestras disponibles en los corpus adquiridos. En este contexto, pueden referenciarse diferentes metodologías estocásticas que modelan el comportamiento del gestor de diálogo [1] [2] [3].

Recientemente, hemos presentado una aproximación estocástica para el desarrollo de un gestor del diálogo [4]. Esta aproximación se basa principalmente en la estimación de un modelo estocástico a partir de las secuencias de actos de diálogo de usuario y de sistema obtenidos a partir de un corpus de entrenamiento. Dado un turno de usuario, el gestor de diálogo asigna la nueva respuesta del sistema como resultado de un proceso de clasificación. Para realizar esta clasificación hemos utilizado redes neuronales.

En este artículo, realizamos diferentes estudios para evaluar, mediante el uso del gestor de diálogo, diferentes aspectos representativos de un corpus de datos adquirido en el marco del proyecto DIHANA [5]. Este proyecto tiene como uno de sus principales propósitos el diseño y desarrollo de un sistema de diálogo adecuado para el acceso a la información mediante habla espontánea. La tarea del proyecto es la consulta en castellano a un sistema de información sobre trayectos de trenes de largo recorrido.

La sección 2 del artículo describe las principales características del corpus adquirido para el proyecto DIHANA, detallándose el proceso de adquisición y de etiquetado del corpus. La sección 3 presenta el gestor de diálogo estocástico utilizado para la evaluación del corpus DIHANA. En esta sección se describe la metodología de gestión propuesta, la representación utilizada para describir la historia del diálogo y el proceso de clasificación para obtener la respuesta del sistema. La sección 4 muestra los resultados de la evaluación del corpus teniendo en cuenta diferentes características representativas del mismo. Finalmente, en la sección 5 se exponen las conclusiones y trabajo futuro.

Este trabajo se ha desarrollado en el marco del proyecto EDECÁN subvencionado por la CICYT número TIN2005-08660-C04-02.

2. EL CORPUS DIHANA

La adquisición de un corpus específico de diálogos usuario-sistema plantea una gran dificultad; ya que, para que esta adquisición se realice de una manera natural se precisa un sistema de diálogo que funcione eficientemente, pero para desarrollar un sistema de diálogo eficiente es necesario una gran cantidad de datos (diálogos naturales) para el entrenamiento de sus modelos. Una de las posibles soluciones para este problema consiste en la utilización de la técnica de Mago de Oz, en la que una persona asume el papel del gestor de diálogo y ayuda al usuario a obtener respuestas a sus consultas siguiendo una estrategia definida.

En el proyecto DIHANA se adquirió un corpus de 900 diálogos utilizando la técnica del Mago de Oz siguiendo una estrategia prefijada. En esta estrategia, el gestor interacciona con el usuario en base a los niveles de confianza suministrados por el sistema, la información proporcionada por el usuario en el turno correspondiente y el estado de una estructura de datos que denominamos registro del diálogo (*dialog register* (DR)). El registro del diálogo se define como una estructura de datos que contiene la información sobre los valores de los conceptos y atributos suministrados por el usuario a través de la historia previa del diálogo.

Si todos los datos del DR disponen de una medida de confianza asociada mayor que el umbral fijado (estado seguro), el gestor elige una de las tres interacciones siguientes:

- Confirmación Implícita y Consulta a la base de datos si el DR dispone del valor de uno de los conceptos y, al menos, de los valores de sus atributos mínimos. (Ej. *Le consulto horarios de trenes con salida en Valencia destino Zaragoza para el viernes 22 de septiembre.*)
- Socitud de información al usuario si el DR no dispone de ningún concepto y/o de alguno de sus atributos mínimos (sin valor por defecto).
- Confirmación Mixta. (Ej. *Quiere horarios a Barcelona , ¿saliendo desde Granada?*). En las confirmaciones mixtas se incorporan referencias no únicamente a valores marcados como poco fiables (*Granada* en el ejemplo anterior), sino también a uno o más conceptos con suficiente fiabilidad (*solicitud de horarios* y *Barcelona*), favoreciéndose una mayor naturalidad en el diálogo. Se realiza sobre el 30 % de turnos seguros en lugar de una Confirmación Implícita-Consulta.

Si el estado es inseguro (aquel en el que uno o más datos del DR poseen una confianza menor que el umbral), el gestor selecciona una de las dos interacciones siguientes:

- Confirmación Explícita del primero de los ítems inciertos que aparezca en el DR. (Ej. *¿Quiere viajar en Talgo?*)
- Confirmación Mixta para darle naturalidad al diálogo. Se realiza sobre el 30 % de turnos de diálogo inseguros en lugar de una Confirmación Explícita.

La adquisición del corpus se llevó a cabo en tres sedes (Bilbao, Zaragoza y Valencia), disponiéndose de un Mago de Oz distinto en cada una de las sedes y utilizándose el mismo sistema y estrategia para la adquisición de diálogos. En la adquisición participaron 225 usuarios (153 hombres y 72 mujeres; 75 usuarios por sede), el número total de turnos de usuario que se adquirieron es 6280 (una media de siete turnos de usuario por diálogo). El vocabulario contiene 823 palabras.

2.1. Etiquetado del corpus

La representación de los turnos de usuario y sistema se realiza en términos de actos de diálogo. Para el caso de los turnos de usuario, los actos de diálogo se corresponden con la interpretación semántica de la intervención del usuario en base a frames (conceptos y atributos). Para la tarea se definieron ocho conceptos y diez atributos (Figura 1).

Conceptos	**Atributos**
Hora	*Origen*
Precio	*Destino*
Tipo-Tren	*Fecha-salida*
Tiempo-Recorrido	*Fecha-Llegada*
Servicios	*Hora-Salida*
Afirmación	*Hora-Llegada*
Negación	*Clase*
No-Entendido	*Tipo-tren*
	Número-Orden
	Servicios

Figura 1. Listado de conceptos y atributos definidos en la tarea DIHANA.

La Figura 2 muestra un ejemplo de la interpretación semántica de un turno de usuario realizada por el módulo de comprensión.

Turno de usuario:
Quisiera conocer los horarios y precios desde Madrid.
Interpretación semántica:
(Hora) Origen: Madrid (Precio) Origen: Madrid

Figura 2. Interpretación semántica de un turno de usuario.

En el caso de los turnos de sistema se definió un etiquetado a tres niveles. El primer nivel contiene las etiquetas referentes a un diálogo general independientemen-

te de la tarea. El segundo nivel representa los conceptos existentes en el turno, siendo dependiente de la tarea. El tercer nivel representa los valores de los atributos dados en el turno. La Figura 3 muestra las etiquetas definidas para cada uno de los niveles del etiquetado.

Primer nivel
Apertura, *Cierre*, *Indefinido*, *No-Entendido*, *Espera*, *Nueva-Consulta*, *Afirmación*, *Negación*, *Pregunta*, *Confirmación* y *Respuesta*
Segundo y tercer nivel
Hora-salida, *Hora-Llegada*, *Precio*, *Tipo-tren*, *Origen*, *Destino*, *Fecha*, *Número-orden*, *Número-Trenes*, *Servicios*, *Clase*, *Tipo-Viaje*, *Tiempo-Recorrido* y *Nil*

Figura 3. Listado de etiquetas definidas para el etiquetado de los turnos de sistema.

En la Figura 4 se muestra un ejemplo del etiquetado de un turno de sistema.

Turno de Sistema:
Le consulto horarios de Valencia a Zaragoza para el viernes 22 de septiembre.
Etiquetado:
(M:Confirmacion:Hora_salida:Destino,Dia,Origen)

Figura 4. Etiquetado de un turno del sistema.

3. EL GESTOR DE DIÁLOGO ESTOCÁSTICO

El gestor de diálogo desarrollado se basa en la modelización estocástica de las secuencias de actos de diálogo, del usuario y del sistema [4]. El gestor genera turnos de sistema basándose únicamente en la información suministrada por los turnos de usuario y la información contenida en el modelo.

Una descripción formal del modelo estocástico propuesto es la siguiente:

Sea A_i la salida del sistema de diálogo (turno de sistema) en el instante i, expresada en términos de actos de diálogo. Sea U_i la representación semántica del turno de usuario (la salida generada por el módulo de comprensión para la intervención del usuario) en el instante i, expresada en términos de frames. Un diálogo comienza con un turno de sistema que da la bienvenida al usuario y le ofrece sus servicios; llamamos a este turno A_1. Consideramos que un diálogo es una secuencia de pares (*turno de sistema, turno de usuario*):

$$(A_1, U_1), \cdots, (A_i, U_i), \cdots, (A_n, U_n)$$

donde A_1 es el turno de bienvenida del sistema, y U_n es el último turno de usuario. De ahora en adelante, denotaremos el par (A_i, U_i) como S_i, el estado de la secuencia del diálogo en el instante i.

En este contexto, consideramos que, en el instante i, el objetivo del gestor de diálogo es encontrar la mejor respuesta de sistema A_i. Esta selección es un proceso local para cada instante i y tiene en cuenta la secuencia de estados de diálogo que preceden a dicho instante. La selección se realiza mediante la siguiente maximización:

$$\hat{A}_i = \underset{A_i \in \mathcal{A}}{\operatorname{argmax}} P(A_i | S_1, \cdots, S_{i-1})$$

donde el conjunto $\mathcal{A}$ contiene todas las posibles respuestas del sistema. Como el número de posibles secuencias de estados es muy grande, establecemos una partición en el espacio de las secuencias de estados, es decir, en la historia del diálogo precediendo el instante i). Para ello, utilizamos el concepto de DR.

Para una secuencia de estados de un diálogo, existe su correspondiente secuencia de DR, donde el primer registro del diálogo (DR_1) contiene la información inicial por defecto del gestor del diálogo (*Origen* y *Clase*), y los siguientes registros DR_i se actualizan teniendo en cuenta la información suministrada durante la evolución del diálogo. Toda la información almacenada en el registro del diálogo en un instante i (DR_i) es un resumen de la información suministrada por la secuencia $S_1, \cdots, S_{i-1}$. Cabe destacar que diferentes secuencias de estados pueden conducir al mismo DR.

De este modo, la partición en el espacio de secuencias de estados se basa en considerar que dos secuencias de estados diferentes son equivalentes si conducen al mismo DR_i. Mediante esta partición obtenemos una gran reducción en el número de historias diferentes en los diálogos a expensas de la pérdida del orden cronológico en el que se suministró la información. Este factor no es un relevante para determinar la próxima respuesta del sistema A_i.

Tras aplicar las consideraciones anteriores y establecer la relación de equivalencia en las historias de los diálogos, la selección de la mejor A_i viene dada por:

$$\hat{A}_i = \underset{A_i \in \mathcal{A}}{\operatorname{argmax}} P(A_i | DR_{i-1}, S_{i-1})$$

Cada turno de usuario suministra al sistema información relativa a la tarea, es decir, el usuario solicita información sobre un concepto específico y/o suministra los valores de determinados atributos. No obstante, un turno de usuario puede además aportar otros tipos de información, como por ejemplo información independiente de la tarea. Éste es el caso de los turnos correspondientes a los actos de diálogo *Afirmación*, *Negación* y *No-Entendido*. Este tipo de información implica una toma de decisiones diferente a una mera actualización del registro DR_{i-1}. Por esta razón, para la selección de la mejor respuesta del sistema A_i, debemos tener en cuenta el DR generado desde el turno 1 al turno $i-2$, y considerar explícitamente el último estado S_{i-1}.

3.1. Representación del Registro del Diálogo

Para la tarea DIHANA, el DR se ha definido como una secuencia de 15 campos, cada uno de ellos asociado a un determinado concepto o atributo.

Para que el gestor de diálogo determine la siguiente respuesta, asumimos que no son significativos los valores exactos de los atributos. Estos valores son importantes para acceder a la base de datos y construir la respuesta del sistema en lenguaje natural. Sin embargo, la única información necesaria para determinar la siguiente acción del sistema es la presencia o no de conceptos y atributos. Por tanto, la información que almacena el DR es una codificación de cada uno de sus campos en términos de tres valores, $\{0, 1, 2\}$, de acuerdo con el siguiente criterio:

- **0**: El usuario no ha suministrado el concepto o valor del atributo correspondiente.
- **1**: El concepto o atributo está presente con una medida de confianza superior a un umbral prefijado. Las medidas de confianza se generan durante los procesos de reconocimiento y comprensión [6].
- **2**: El concepto o atributo está presente con una medida de confianza inferior al umbral.

3.2. Clasificación mediante perceptrones multicapa

Los perceptrones multicapa (MLPs) [7] son las redes neuronales más comúnmente utilizadas en tareas de clasificación. Utilizando un perceptrón multicapa, la capa de entrada se definió teniendo en cuenta la codificación utilizada para el par de entrada (DR_{i-1}, S_{i-1}). La capa de salida se definió de acuerdo con el número de posibles respuestas del sistema (51 en nuestro caso), seleccionándose como salida la respuesta de sistema A_i con mayor probabilidad asociada.

La representación definida para representar el par de entrada (DR_{i-1}, S_{i-1}) es la siguiente:

- Los dos primeros niveles del etiquetado de la última respuesta dada por el sistema (A_{i-1}): Esta información se modela mediante una variable, que posee tantos bits como posibles combinaciones de estos dos niveles (51).

$$\vec{x}_1 = (x_{1_1}, x_{1_2}, x_{1_3}, \cdots, x_{1_{51}}) \in \{0,1\}^{51}$$

- Registro del diálogo (DR): Tal y como se ha comentado previamente, el DR almacena un total de quince características (5 conceptos y 10 atributos). Cada una de estas características pueden tomar los valores $\{0, 1, 2\}$. De este modo, cada uno de los atributos del DR puede modelarse utilizando una variable con tres bits.

$$\vec{x}_i = (x_{i_1}, x_{i_2}, x_{i_3}) \in \{0,1\}^3 \, i = 2, ..., 10$$

  Dado que el sistema está orientado a completar los datos mínimos para hacer una única consulta a la base de datos, la estrategia definida para el Mago de Oz limita que en el DR únicamente puede existir un único concepto (consulta) activo simultáneamente. De esta forma, los diferentes valores que pueden tomar el conjunto de los conceptos se codifican mediante una única variable de 11 bits.

$$\vec{x}_{11} = (x_{i_1}, x_{i_2}, ..., x_{i_{11}}) \in \{0,1\}^{11}$$

- Información independiente de la tarea (actos de diálogo *Afirmación*, *Negación*, y *No-Entendido*): Estos tres actos de diálogo se han codificado de forma idéntica a los atributos almacenados en el DR. De esta forma, cada uno de estos tres actos de diálogo puede tomar los valores $\{0, 1, 2\}$ y modelarse utilizando una variable con tres bits.

$$\vec{x}_i = (x_{i_1}, x_{i_2}, x_{i_3}) \in \{0,1\}^3 \, i = 12, ..., 14$$

De este modo, la variable (DR_{i-1}, S_{i-1}) puede representarse mediante el vector de 14 características:

$$(DR_{i-1}, S_{i-1}) = (\vec{x}_1, \vec{x}_2, \vec{x}_3, \cdots, \vec{x}_{14})$$

4. EVALUACIÓN

La evaluación de las características del corpus se llevó a cabo mediante un proceso de validación cruzada. En cada una de las experimentaciones que se presentan, el corpus se dividió aleatoriamente en cinco subconjuntos. Cada evaluación, de este modo, consistió en cinco experimentaciones. En cada una de ellas se utilizó un subconjunto diferente de los cinco definidos como muestras de test, y el 80 % del corpus restante se utilizó como partición de entrenamiento.

Los clasificadores mediante MLP se programaron utilizando un software desarrollado por nuestro grupo de investigación. Los MLPs se entrenaron utilizando el algoritmo de Backpropagation con momento [7]. La topología y parámetros del algoritmo (como la tasa de aprendizaje y el momento) se estimaron mediante una búsqueda exhaustiva, utilizando el error cuadrático medio obtenido (MSE) como criterio de parada para el conjunto de validación (20 % de las muestras de entrenamiento). La mejor topología fue dos capas ocultas con 100 y 10 neuronas respectivamente.

Para realizar la evaluación se definieron cuatro medidas. La primera es el porcentaje de respuestas que coinciden con aquella generada por el Mago de Oz ($\%exacta$). Teniendo en cuenta que la estrategia del Mago de Oz posibilita un conjunto de respuestas válidas dado un determinado estado del diálogo, la segunda medida indica el porcentaje de respuestas que siguen dicha estrategia ($\%estrategia$). La tercera medida es el porcentaje de respuestas que son coherentes con el estado actual del diálogo

($\%correcta$), aunque no necesariamente sigan la estrategia fijada para el Mago de Oz. Finalmente, la cuarta medida es el porcentaje de respuestas que no son compatibles con el estado actual del diálogo ($\%error$), provocando el fallo del diálogo. Estas tres últimas medidas han sido obtenidas tras un proceso de revisión manual. Los resultados que son relevantes en el ámbito de la gestión del diálogo son $\%estrategia$, $\%correcta$ y $\%error$.

4.1. Evaluación del gestor de diálogo

Para realizar una evaluación del funcionamiento de la metodología estocástica, se realizaron unas particiones de entrenamiento y test que equilibran el número de turnos de usuario en cuanto a su localización, sexo, duración temporal de las intervenciones y número de palabras. Los conjuntos de entrenamiento contenían 4420 muestras y los de test, 1105 muestras.

Los resultados obtenidos para cada una de las medidas definidas aparecen en la Figura 5. Dichos resultados muestran el funcionamiento satisfactorio del gestor de diálogo desarrollado. La codificación propuesta para representar el estado del diálogo y el buen funcionamiento del clasificador MLP hacen posible que la respuesta generada por el sistema coincida con una de las respuestas definidas en la estrategia en un porcentaje del 97,73 %. Dicha respuesta, además, coincide exactamente con la seleccionada por el Mago de Oz en un 78,62 % de los casos. Finalmente, el número de respuestas que pueden generar el fallo del sistema es sólo un 0,18 %. Teniendo en cuenta que la media de turnos de sistema por diálogo es aproximadamente 7, a partir del porcentaje $\%error$ obtenido puede estimarse que la probabilidad de que un diálogo sea fallido es del 1,25 %.

$\%exacta$	78,62 %
$\%estrategia$	97,73 %
$\%correcta$	99,68 %
$\%error$	0,18 %

Figura 5. Evaluación del modelo estocástico.

4.2. Evaluación de la influencia de la talla del corpus

Para realizar una evaluación del funcionamiento del modelo de diálogo frente a la talla del corpus de entrenamiento, se utilizaron las mismas particiones que en el apartado anterior, descartándose muestras de forma aleatoria en los conjuntos de entrenamiento para reducir el tamaño de esta partición. Se llevaron a cabo tres experimentaciones, utilizándose un 75 % de las muestras de entrenamiento (3315 muestras), un 50 % de dicho conjunto (2210 muestras) y un 25 % (1105 muestras). Los conjuntos de test son idénticos a los del apartado anterior. La Figura 6 muestra los resultados de las experimentaciones llevadas a cabo.

	100 %	75 %	50 %	25 %
$\%exacta$	78,62 %	73,77 %	71,17 %	66,21 %
$\%estrategia$	97,73 %	94,88 %	92,56 %	90,61 %
$\%correcta$	99,68 %	98,62 %	97,33 %	96,10 %
$\%error$	0,18 %	0,52 %	0,69 %	0,95 %

Figura 6. Evaluación del modelo estocástico frente a diversas tallas del corpus de entrenamiento.

La evaluación realizada muestra el correcto funcionamiento del gestor incluso si se utiliza únicamente un 25 % del corpus de entrenamiento para construir el modelo estocástico. De esta forma, los resultados obtenidos para la medida $\%correcta$ son muy similares en las tres experimentaciones. No obstante, si utilizamos únicamente un 25 % del corpus de entrenamiento, el porcentaje de respuestas del sistema que pueden causar el fallo del diálogo es del 0,95 %, lo que nos indica que aproximadamente una de cada cien respuestas dadas por el sistema ocasionará el fallo del diálogo (6,46 % de probabilidad de fallo del diálogo).

El alto porcentaje de acierto del gestor de diálogo (incluso para un tamaño del corpus de entrenamiento reducido) puede explicarse porque existen muchos estados del diálogo que, o bien son muy frecuentes en el corpus, o bien son similares a otros estados y, por tanto, fácilmente clasificables por el MLP.

4.3. Evaluación de la influencia del sexo

Para evaluar el comportamiento del gestor teniendo en cuenta el sexo de los usuarios, se realizaron particiones del corpus con igual número de muestras de mujeres y de hombres. Los resultados de esta experimentación aparecen en la Figura 7, indicándose qué partición se utilizó como entrenamiento y como test (Entrenamiento / Test).

De la observación de los resultados, puede destacarse que prácticamente no existen diferencias remarcables en los resultados obtenidos si se realiza el aprendizaje del modelo utilizando únicamente muestras de hombres o de mujeres (primeras dos columnas de resultados en la Figura 7). Mayores diferencias se observan en las experimentaciones llevadas a cabo evaluando el modelo teniendo en cuenta el sexo de los usuarios en las particiones de test (tercera y cuarta columna de dicha figura). La diferencias obtenidas en esta última experimentación parecen indicar una mayor similitud en las muestras de hombres.

4.4. Evaluación de la procedencia de los diálogos

La evaluación del gestor de diálogo en este contexto se llevó a cabo partiendo de las mismas particiones que en el primer apartado de la evaluación, entrenando el modelo de diálogo únicamente con las muestras procedentes de la sede a evaluar y utilizando las mismas particiones de test que en dicho apartado (muestras procedentes de las tres

	Mujeres / Ambos	Hombres / Ambos	Ambos / Mujeres	Ambos / Hombres
$\%exacta$	70,55 %	69,97 %	71,72 %	76,09 %
$\%estrategia$	92,56 %	95,92 %	94,67 %	97,79 %
$\%correcta$	96,93 %	97,96 %	97,75 %	99,71 %
$\%error$	0,73 %	0,58 %	0,49 %	0,14 %

Figura 7. Evaluación del modelo estocástico teniendo en cuenta el sexo de los usuarios.

sedes). Los resultados de la experimentación aparecen reflejados en la Figura 8.

	Sede1	Sede2	Sede3
$\%exacta$	71,94 %	77,24 %	70,39 %
$\%estrategia$	95,55 %	95,51 %	90,03 %
$\%correcta$	99,16 %	97,11 %	92,15 %
$\%error$	0,42 %	1,44 %	3,92 %

Figura 8. Evaluación del modelo estocástico teniendo en cuenta la procedencia de los diálogos.

De la observación de los resultados de las diferentes experimentaciones, se aprecia un mejor funcionamiento de la metodología realizando el aprendizaje del modelo con los diálogos adquiridos en la Sede1. Aprendiendo el modelo únicamente con los diálogos de la Sede2, se obtiene un porcentaje de respuestas que siguen la estrategia ($\%estrategia$) equivalente al de la Sede1. Sin embargo, el porcentaje de respuestas erróneas ($\%error$) es tres veces mayor. En cuanto al corpus de diálogos adquirido en la Sede3, a pesar de obtener un porcentaje de respuestas correctas del 92,15 %, el número de respuestas de sistema que pueden causar el fallo del diálogo es muy elevado en comparación con el obtenido en las dos sedes restantes (probabilidad del 24,42 % de que el diálogo sea fallido). Por tanto, teniendo en cuenta la medida $\%error$, existe una diferencia significativa entre los diálogos adquiridos en cada una de las sedes.

5. CONCLUSIONES

En este artículo, hemos presentado una aproximación para el desarrollo de un gestor de diálogo estocástico, aprendida a partir de un conjunto de muestras de entrenamiento y basada en un proceso de clasificación para obtener la siguiente respuesta del sistema. La metodología propuesta es aplicable a otras tareas que compartan la filosofía de solicitar información al usuario para completar un determinado objetivo (tareas $slot - filling$).

Se ha realizado una evaluación para conocer tanto el funcionamiento de la metodología utilizada como las características del corpus adquirido para nuestra tarea. Esta experimentación nos ha permitido conocer el buen funcionamiento del sistema, así como evaluar la influencia de las características más relevantes del corpus en el funcionamiento del modelo estocástico. Los diferentes trabajos llevados a cabo para realizar la elección más adecuada de la codificación de la historia del diálogo, la metodología de clasificación propuesta y la calidad y coherencia del corpus, son las principales causas del buen funcionamiento del sistema. Como trabajo futuro, cabe destacar la utilización de un simulador de usuarios desarrollado en nuestro grupo de investigación con los objetivos principales de ampliar la evaluación presentada y mejorar el modelo de gestión estocástico.

6. BIBLIOGRAFÍA

[1] J. Williams, P. Poupart, y S. Young, "Partially Observable Markov Decision Processes with Continuous Observations for Dialogue Management," in *Recent Trends in Discourse and Dialogue. Eds L. Dybkjaer and W. Minker, Springer*, 2006.

[2] E. Levin, R. Pieraccini, y W. Eckert, "A stochastic model of human-machine interaction for learning dialog strategies," in *IEEE Transactions on Speech and Audio Processing*, 2000, pp. 8(1):11–23.

[3] F. Torres, L.F. Hurtado, F. García, E. Sanchis, y E. Segarra, "Error handling in a stochastic dialog system through confidence measures," in *Speech Communication*, 2005, pp. (45):211–229.

[4] L.F Hurtado, D. Griol, E. Sanchis, y E. Segarra, "A Stochastic Approach for Dialog Management based on Neural Networks," in *Proc. Interspeech*, 2006.

[5] J.M. Benedí, A. Varona, y E. Lleida, "DIHANA: Sistema de diálogo para el acceso a la información en habla espontánea en diferentes entornos," in *Actas de las III Jornadas en Tecnología del Habla*, Valencia (España), 2004, pp. 141–146.

[6] F. García, L. Hurtado, E. Sanchis, y E. Segarra, "The incorporation of Confidence Measures to Language Understanding," in *International Conference on Text Speech and Dialogue (TSD 2003). Lecture Notes in Artificial Intelligence series 2807*, Ceské Budejovice (Czech Republic), 2003, pp. 165–172.

[7] D. E. Rumelhart, G. E. Hinton, y R. J. Williams, *PDP: Computational models of cognition and perception, I*, chapter Learning internal representations by error propagation, pp. 319–362, MIT Press, 1986.

DS-UCAT: SISTEMA DE DIÁLOGO MULTIMODAL Y MULTILINGÜE PARA UN ENTORNO EDUCATIVO

Ramón López-Cózar[1], Zoraida Callejas[1], Germán Montoro[2], Pablo Haya[2]

[1]Dpto. Lenguajes y Sistemas Informáticos, ETS Ingenierías Informática y Telecomunicaciones, Universidad de Granada, {rlopezc, zoraida}@ugr.es

[2]Dpto. de Ingeniería Informática, Universidad Autónoma de Madrid {German.Montoro, Pablo.Haya}@uam.es

RESUMEN

En este artículo presentamos un sistema de diálogo multimodal y multilingüe que estamos desarrollando para proporcionar asistencia a estudiantes y profesores en algunas de sus actividades habituales en un entorno educativo, p. e. en una Facultad de una Universidad. Tenemos previsto que además de interactuar con el usuario, el sistema pueda interactuar con el entorno en que éste se encuentra en un momento dado, el cual puede cambiar a lo largo de una interacción conforme el usuario se mueve dentro del centro educativo. El artículo describe la arquitectura del sistema, muestra cómo se realiza la interacción con la versión actual del mismo, y comenta cómo tenemos previsto utilizar técnicas de inteligencia ambiental para mejorar su funcionamiento.

1. INTRODUCCIÓN

Recientemente se han desarrollado diversas tecnologías para facilitar el uso de los ordenadores a personas no familiarizadas con la Informática. Por ejemplo, los sistemas de diálogo oral (*Spoken Dialogue Systems*) son programas informáticos diseñados para emular el comportamiento y la capacidad de comunicación de un ser humano ante una tarea dada [1]. Su finalidad es proporcionar información a los usuarios (usualmente a través del teléfono) usando habla como medio de interacción. Algunos sistemas de este tipo que pueden encontrarse en la bibliografía son los siguientes: Pegasus [2], Voyager [3], Dialogos [4], Saplen [5], Jupiter [6], Mercury [7], Dihana [8], Ritel [9] and UAH [10]. Los sistemas de diálogo multimodal (*Multimodal Dialogue Systems*) son el resultado de una sofisticación de los sistemas de diálogo oral, pues permiten que el usuario pueda utilizar diversos canales de comunicación durante la interacción (p. e. habla, texto, sonidos, etc.). Para hacer posible la interacción con usuarios que hablan distintos idiomas, los sistemas de diálogo multilingüe (*Multilingual Dialogue Systems*) permiten que los usuarios puedan comunicarse con el ordenador usando un idioma de entre varios disponibles (p. e. Castellano o Inglés), siendo la funcionalidad del sistema la misma independientemente del idioma utilizado [11].

Por otra parte, los denominados entornos *"activos"* (también llamados entornos de *"inteligencia ambiental"*) tienen como finalidad proporcionar un modo de interacción natural entre el entorno y sus *"habitantes"* [12]. Su objetivo es que el entorno ayude a las personas en su vida diaria, ofreciendo modos no intrusivos de comunicación. De este modo, se pretende que los hogares, aulas, despachos o laboratorios, por ejemplo, sean capaces de asistir a los usuarios en la realización de sus tareas cotidianas [13], [14].

Tras esta breve introducción, el resto del artículo está organizado como sigue. La sección 2 presenta el sistema de diálogo DS-UCAT, comentando su organización mediante documentos X+V, sus interfaces oral y visual, la conexión entre ambas interfaces, y las bases de datos usadas actualmente por el sistema, así como otras que están pendientes de ser creadas. La sección 3 describe cómo se realiza la interacción con la versión actual del sistema, y muestra, a modo de ejemplo, cómo se puede realizar la consulta multimodal de fuentes bibliográficas. La sección 4 comenta cómo tenemos previsto implementar la interacción entre el sistema de diálogo y el entorno educativo. Finalmente, la sección 5 presenta las conclusiones y comenta líneas de trabajo futuro.

2. EL SISTEMA DS-UCAT

Una línea de investigación del proyecto UCAT (*Ubiquitous Collaborative Training*)[1] está dedicada a la implementación de un sistema de diálogo multimodal y multilingüe denominado DS-UCAT (*Dialogue System for Ubiquitous Collaborative Training*). La finalidad de este sistema es proporcionar asistencia a profesores y

[1] Este trabajo ha sido financiado por el Ministerio de Ciencia y Tecnología, mediante el proyecto TIN2004-03140 Ubiquitous Collaborative Training.

estudiantes en algunas de sus actividades cotidianas dentro un centro educativo (p. e. en una Facultad de una Universidad). En el diseño actual consideramos que el sistema podrá prestar servicio en tres tipos de ubicaciones dentro del entorno educativo: biblioteca, despachos de profesores y aulas. En nuestra implementación actual el sistema puede interactuar con el usuario usando sonidos, voz, gráficos y texto. La entrada multimodal permite al usuario proporcionar información al sistema a través de habla, teclado y ratón. Por ejemplo, el usuario puede consultar si existe disponibilidad de libros sobre una determinada materia en la biblioteca pronunciando el nombre de la materia (p. e. Matemáticas), seleccionándola con el ratón en una lista de materias, o escribiendo el nombre de la materia en un campo de un formulario. Dado que la salida del sistema también es multimodal, para esta consulta el sistema genera un mensaje oral indicando que se muestra en pantalla un listado con libros encontrados en la base de datos relacionados con la materia indicada.

La Figura 1 muestra la arquitectura modular de la versión actual del sistema, compuesta por un servidor de documentos XHTML+Voice (también llamados documentos X+V) al que acceden los dispositivos móviles de los usuarios (Tablet PCs, ordenadores portátiles o PDAs) mediante conexiones inalámbricas.

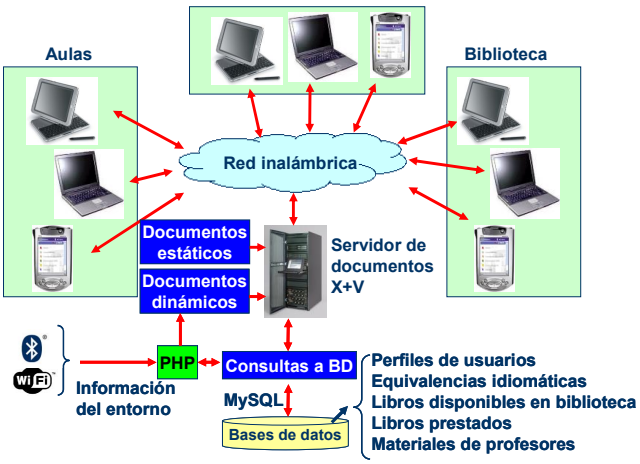


Figura 1. *Arquitectura del sistema DS-UCAT.*

2.1. Documentos XHTML+Voice (X+V)

El sistema está implementado mediante un conjunto de documentos X+V. Algunos de ellos se encuentran almacenados en el servidor de documentos X+V mostrado en la Figura 1, mientras que otros se generan dinámicamente usando programas PHP que tienen en cuenta las características y preferencias de los usuarios, p. e. sexo (masculino/femenino) y lenguaje de interacción preferido (p. e. Inglés), así como información extraída de diversas bases de datos. Los documentos X+V constan de formularios cuyos campos se pueden rellenar mediante habla, texto o selecciones de opciones efectuadas con el ratón.

Para visualizar los documentos, los usuarios deben disponer en sus dispositivos móviles de un navegador que soporte la interacción oral y soporte la especificación X+V. En nuestra implementación usamos el navegador Opera[2], el cual permite introducir comandos mediante voz (p. e. para recargar una página, detener el acceso a Internet, volver hacia atrás, etc.). Además, usando las gramáticas adecuadas para el reconocimiento del habla (RAH), el navegador puede reconocer las frases pronunciadas relacionadas con nuestro dominio de aplicación (entorno educativo). Para implementar la interacción multimodal hemos desarrollado dos interfaces (una para la interacción oral y la otra para la interacción visual) que describimos a continuación.

2.1.1. Interfaz oral

La interfaz oral permite realizar el RAH y la síntesis del habla. Ambos procesos se llevan a cabo mediante el reconocedor y sintetizador de habla incluidos en el navegador Opera. En nuestra implementación actual el RAH se realiza mediante la técnica tap-&-talk, es decir, el usuario debe hacer clic en un determinado icono del navegador y mantenerlo pulsado mientras está hablando (no obstante, esta es una características que puede ser cambiada fácilmente en la sección de "preferencias" del navegador). Para realizar el RAH y la comprensión del habla utilizamos gramáticas JSGF (Java Speech Grammar Format). Algunas de estas gramáticas las usamos a nivel de formulario (para que se puedan rellenar varios campos del mismo en una única interacción oral) mientras que otras las usamos a nivel de campo (para que sólo se pueda rellenar un campo del formulario en una interacción). Algunas gramáticas son estáticas, mientras que otras se generan dinámicamente usando programas PHP que realizan consultas MySQL a las bases de datos e incluyen los datos obtenidos en el vocabulario de dichas gramáticas (p. e. títulos de libros). Por ejemplo, utilizando una de estas gramáticas para reconocer consultas de libros efectuadas en el idioma Inglés, si el usuario pronuncia la frase "I need books about math please", el sistema rellena el campo "materia" con la palabra "math".

Las gramáticas usadas para el reconocimiento de consultas de libros deben ser actualizadas conforme se producen cambios en el catálogo de libros disponibles en la biblioteca, de ahí que se generen de forma dinámica a partir del contenido de la base de datos Libros Disponible en biblioteca (ver Figura 1). Para actualizar estas gramáticas hemos implementado un programa PHP que realiza dos tareas. En primer lugar, realiza consultas MySQL a las base de datos para obtener la información relativa a los libros disponibles,

[2] http://www.opera.com

como autores, títulos y materias. En segundo lugar, crea gramáticas para reconocer tanto frases completas como datos de forma aislada (p. e. títulos, autores o materias) usando la información obtenida en la etapa anterior.

En la salida del sistema, la síntesis de habla se realiza incluyendo frases en formato de texto en sentencias VoiceXML[3] del tipo <prompt> … </prompt>. Estas frases se transforman en voz mediante el sintetizador de voz integrado en el navegador. Algunas de estas frases tienen un formato fijo, mientras que otras se crean en tiempo de ejecución teniendo en cuenta el tipo de usuario (profesor o estudiante), su sexo (masculino o femenino) y la información extraída de algunas bases de datos.

2.1.2. Interfaz visual

En la entrada al sistema, la interacción visual se usa para obtener datos del usuario mediante campos y botones para la selección de opciones mediante el ratón, típicamente usados en XHTML (ver Figuras 2 y 3). En la salida del sistema, la interacción visual se utiliza para proporcionar al usuario los datos obtenidos de las bases de datos (p. e. lista de libros disponibles relacionados con una determinada materia), así como información acerca del usuario actual (ver Figura 2).

2.1.3. Conexión entre ambas interfaces

Para crear la conexión entre ambas interfaces usamos manejadores de eventos que incluimos en la sección body de los documentos X+V. De esta forma, cuando el navegador carga el documento para realizar consultas bibliográficas, por ejemplo, se produce el evento on load y, como consecuencia, se ejecuta un formulario VoiceXML al que hemos llamado initial_vform para atender dicho evento. Para que una frase del usuario pueda rellenar varios campos de un formulario en una misma interacción, usamos una etiqueta del tipo: <vxml:initial name="initial_vform" … </initial> (típicamente empleada en VoiceXML) y empleamos una gramática a nivel de formulario. Así, mediante el documento para la consulta bibliográfica, el sistema puede generar el mensaje oral "Por favor, realice una consulta bibliográfica", y el usuario puede responder a este mensaje pronunciando un número variable de ítems de información, como por ejemplo: autores, autores y año de publicación, autores, año de publicación y materia, etc. También usamos en nuestra implementación eventos del tipo: ev:event="onclick", cuya finalidad es generar un evento conforme el usuario hace clic en un campo de un formulario. El manejador para este tipo de evento es un código escrito en VoiceXML para obtener el valor proporcionado por el usuario para dicho campo.

[3] http://www.w3.org/TR/voicexml2.0

2.2. Bases de datos

Para proporcionar información a los usuarios e interactuar con ellos adecuadamente, el sistema realiza consultas MySQL a varias bases de datos. La base de datos Perfiles de usuarios (ver Figura 1) contiene información acerca de los usuarios del sistema, incluyendo su nombre, sexo, dirección y número de teléfono. Esta base de datos también almacena tres tipos de preferencias: i) lenguaje para la interacción (Castellano o Inglés, de momento), ii) interacción oral (habilitada/deshabilitada) y iii) tipo de voz del sistema (masculina/femenina).

La base de datos Equivalencias idiomáticas almacena expresiones en diversos idiomas correspondientes a tipos concretos de frases que el sistema puede usar a lo largo de la interacción. La selección de un tipo de expresiones u otro se realiza en base al idioma seleccionado por el usuario en su perfil. Un ejemplo de tales expresiones es el mensaje de bienvenida que el sistema proporciona al usuario confirme éste accede al sistema: "Welcome to the DS-UCAT system" para la interacción en Inglés, o "Bienvenido al Sistema DS-UCAT" para la interacción en Castellano.

Adicionalmente, en nuestra implementación actual el sistema usa dos bases de datos de forma provisional, que en una implementación posterior deberán ser substituidas por bases de datos reales. Por una parte, la base de datos Libros disponibles en biblioteca almacena información relativa a fuentes bibliográficas supuestamente disponibles en la biblioteca del centro educativo. Por otra, la base de datos Libros prestados almacena datos de libros supuestamente prestados por la biblioteca a usuarios del sistema (profesores o estudiantes). Usando ambas bases de datos el sistema puede responder a preguntas como la mostrada en la Figura 3 para una interacción en Inglés. Empleando el formulario de la figura, cuando el usuario haciendo clic en un campo, el sistema genera un mensaje oral para solicitar al usuario que introduzca un dato para ese campo (p. e. "Book title?"). El usuario puede proporcionar este dato de forma oral o mediante texto. Tras consultar la base de datos Libros disponibles en biblioteca, el sistema puede generar oralmente un mensaje del tipo "The following books are available ….", o bien, el mensaje "No books were found" si no encontró ningún registro en la base de datos que satisfaga los criterios de búsqueda.

La base de datos Materiales de profesores es una base de datos pendiente de ser creada cuya finalidad será almacenar documentos proporcionados a los estudiantes por los profesores. Nuestro objetivo es que los documentos X+V para los entornos de trabajo Clase y Despacho de Profesor (también pendientes de ser creados) muestren una vista de esta base de datos, de forma similar a como lo hace el documento X+V mostrado en la Figura 3 usado para consultar la base de datos de Libros disponibles en biblioteca.

3. USO DEL SISTEMA DS-UCAT

Para poder interactuar con la versión actual del sistema el usuario debe comenzar por identificarse mediante un nombre de usuario y una clave. A partir de esta identificación el sistema determina el tipo de usuario de que se trata, consultando la base de datos Perfiles de usuarios. En función de las características del usuario, el sistema determina el idioma que debe usar durante la interacción, así como el uso (o no) de interacción oral, y el tipo de voz que debe emplear (masculina o femenina). Por ejemplo, la Figura 2 presenta la ventana mostrada inicialmente por el sistema para un usuario que seleccionó Inglés como idioma de interacción. Además, dado que este usuario indicó en su perfil que deseaba utilizar la modalidad oral, a la vez que el sistema muestra la pantalla de bienvenida, genera el mensaje oral "Hello Ramón, welcome to the DS-UCAT system".

Como se puede observar en la Figura, tras haberse identificado en el sistema, el usuario debe seleccionar el entorno de trabajo que desea utilizar (biblioteca, aula o despacho de profesor). Esta selección inicial le permite trabajar en un entorno que no es aquél en que se encuentra en un momento dado, lo que le permite, por ejemplo, realizar consultas bibliográficas desde un aula.

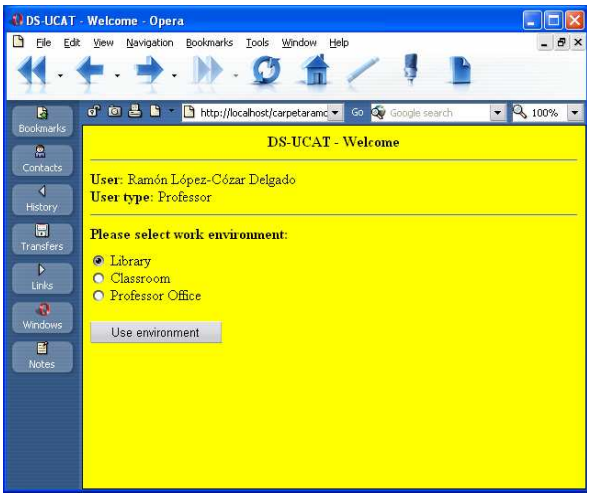


Figura 2. *Mensaje de bienvenida para interacción en Inglés.*

La Figura 3 muestra el documento X+V que aparece en pantalla conforme el usuario comienza a usar el entorno de trabajo Biblioteca. Dado que el usuario habilitó el uso de interacción oral en su perfil, el sistema general el mensaje oral "Please enter a book query". Para realizar la entrada mediante voz, el usuario debe mantener pulsada una determinada tecla o hacer clic en un determinado botón del navegador conforme pronuncia cada frase (técnica tap-&-talk). Además, puede efectuar la consulta proporcionado los datos de la consulta mediante el teclado y el ratón.

Hemos implementado en el sistema las dos estrategias de interacción típicamente empleadas en sistemas de diálogo: mixta y dirigida por el sistema. El uso de una u otra depende del tipo de documento X+V con el que interactúa el usuario en un momento dado. Por ejemplo, si la interacción se realiza con el documento diseñado para realizar consultas bibliográficas, el sistema utiliza interacción mixta para permitir el relleno de varios campos del formulario en una misma interacción (p. e. mediante la pronunciación de la frase "I need information about books written by Prieto in 2006"). Si la frase no puede ser reconocida correctamente, el sistema solicita datos al usuario con objeto de rellenar los campos del formulario uno a uno (iniciativa dirigida por el sistema).

El sistema tiene en cuenta tres tipos de eventos que suelen ocurrir en una interacción oral con un sistema de diálogo: i) que el usuario solicite ayuda, ii) que el usuario no pronuncie ninguna palabra cuando el sistema lo espera, y iii) que la entrada obtenida del usuario no pueda ser reconocida por ninguna gramática activa. Para procesar estos eventos, usamos contralores de eventos de tipo help, noinput y nomatch, respectivamente, de forma similar a como se usan en VoiceXML.

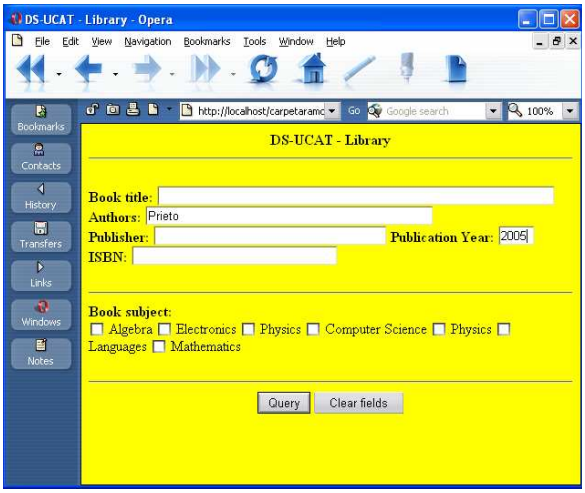


Figura 3. *Consulta bibliográfica para interacción en Inglés.*

4. INTERACCIÓN ENTRE EL SISTEMA DS-UCAT Y EL ENTORNO EDUCATIVO

Otra línea de trabajo dentro del proyecto UCAT (*Ubiquitous Collaborative Training*) está relacionada con la implementación de un sistema de identificación y localización automática de usuarios en el entorno educativo. Su finalidad es personalizar diversos servicios ofrecidos dependiendo del usuario y de la situación de éste dentro del entorno. Para implementar este sistema estamos utilizando una combinación de diversas tecnologías cuya integración permita obtener un conocimiento más preciso del contexto del usuario. Concretamente, por una parte hemos desplegado un

sistema de identificación y localización mediante radio frecuencia que permite conocer los lugares a los que acceden las personas. La infraestructura del sistema consta de un conjunto de antenas RFID (radio frecuencia) situadas en los marcos de entrada de las distintas estancias. Para ser localizado dentro del entorno, cada usuario dispone de una tarjeta RFID pasiva que le identifica unívocamente. Cuando una persona accede a una estancia, la antena correspondiente lee la información de la tarjeta RFID y comunica a un servidor central la nueva localización del usuario. Este mecanismo ha sido desarrollado y utilizado dentro de un entorno aplicado al hogar (ver Figura 4), lo que nos ha permitido comprobar su funcionalidad en condiciones reales para controlar oralmente lámparas, equipos de música y otros dispositivos [15]. Por otra parte, dado que puede no ser posible situar antenas en todas las estancias, también hemos desarrollado un sistema que permite detectar la actividad de las personas en su equipo de trabajo. De este modo se puede determinar la localización de una persona si se encuentra trabajando con su ordenador de la oficina.

Tenemos previsto hacer uso de estos mecanismos de localización e identificación en el sistema DS-UCAT con objeto de mejorar su funcionamiento. De esta forma, se evitará la necesidad de realizar una identificación inicial mediante nombre de usuario y contraseña para acceder al sistema, pues éste podrá determinar qué usuario está interactuando a partir de la información obtenida del servidor central de localización.

Figura 4. *Entorno de interacción para el hogar desarrollado en el proyecto Odisea.*

Además, se podrá adaptar automáticamente diversos aspectos del funcionamiento del sistema. Por ejemplo, el sistema de RAH quizás se pueda adaptar al usuario y al entorno en que éste se encuentra, empleando modelos acústicos y de lenguaje específicos para cada tipo de ubicación dentro del entorno educativo. También se podría adaptar la forma en que el sistema interacciona con el usuario, usando para ello diferentes modelos de diálogo, específicos para el lugar concreto dentro del centro educativo en que se encuentre el usuario en cada momento.

5. CONCLUSIONES Y TRABAJO FUTURO

Este trabajo se ha centrado, fundamentalmente, en nuestra tarea dentro del proyecto UCAT relacionada con la implementación de un sistema de diálogo multimodal y multiligüe que proporcione asistencia a profesores y estudiantes en algunas de sus actividades cotidianas dentro de un centro educativo (p. e. en una Facultad de una Universidad). El artículo ha abordado la implementación actual del mismo, basada en documentos X+V. Algunos de estos documentos están almacenados en un servidor central, mientras que otros se generan dinámicamente en tiempo de ejecución mediante programas PHP. Este servidor interactúa con los dispositivos móviles de los usuarios mediante conexiones inalámbricas. A modo de ejemplo, hemos mostrado cómo se puede realizar la consulta multimodal de fuentes bibliográficas.

Las principales líneas de trabajo futuro están relacionadas con las siguientes tareas:

- Creación de los documentos X+V y las bases de datos necesarias para permitir la interacción con el sistema en los entornos de trabajo Aula y Despacho de Profesor.

- Desarrollo de la funcionalidad del sistema que permita interactuar con los dispositivos del entorno. Para ello tenemos previsto utilizar el gestor de diálogos Odisea (usado en el entorno del hogar citado anteriormente [15]), en el cual estamos trabajando actualmente para adaptarlo a los nuevos entornos de trabajo dentro del centro educativo [16]. De esta forma lograremos, por ejemplo, que los profesores puedan interactuar de forma multimodal y multilingüe con el proyector de transparencias, las luces del aula, o el equipo de música de su despacho.

6. BIBLIOGRAFÍA

[1] McTear, M. F. *Spoken Dialogue Technology: Toward the Conversational User Interface*, Springer, 2004.

[2] Zue, V., Seneff, S., Polifroni, J., Phillips, M., Pao, C., Goodine, D., Goddeau, D., Glass, J. "PEGASUS: A spoken dialogue interface for on-line air travel planning". *Speech Communication*, 15, pp. 331-340, 1994.

[3] Glass, J., Flammia, G., Goodine, D., Phillips, M., Polifroni, J., Sakai, S., Seneff, S., Zue, V. "Multilingual spoken-language understanding in the MIT Voyager system". *Speech Communication*, 17 (1-2), pp. 1-18, 1995.

[4] Danielli, M. “On the use of expectations for detecting and repairing human-machine miscommunication”. Working notes of the AAAI Workshop on Detecting, Repairing, and Preventing Human-Machine Miscommunication, pp. 87-93, 1996.

[5] López-Cózar, R., García, P., Díaz, J., Rubio, A. J. “A voice activated dialogue system for fast-food restaurant applications”. Proc. Eurospeech, pp. 1783-1786, 1997.

[6] Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T., Hetherington, L. “Jupiter: A telephone-based conversational interface for weather information”. *IEEE Trans. on Speech and Audio Proc.*, 8(1), pp. 85-96, 2000.

[7] Seneff S., Polifroni J. “Dialogue management in the Mercury flight reservation system”. Proc. ANLP-NAACL Satellite Workshop, pp. 1-6, p. 2000.

[8] Torres, F., Sanchis, E., Segarra, E. “Learning of stochastic dialog models through a dialogue simulation technique”. Proc. Interspeech, pp. 817-820, 2005.

[9] Galibert, O., Illouz, G., Rosset, S. “Ritel : An open-domain, human-computer dialog system”. Proc. Interspeech, pp. 909-912, 2005.

[10] Callejas, Z., López-Cózar, R. “Implementing modular dialogue systems – A case of study”. Proc. ISCA Research Workshop on Applied Spoken Language Interaction in Distributed Environments, 2005.

[11] López-Cózar, R., Araki, M. *Spoken, Multilingual and Multimodal Dialogue Systems: Development and Assessment*. John Wiley & Sons Publishers, 2005.

[12] Coen, M. “Building brains for intelligent environments”. Proc. National Conference on Artificial Intelligence, 1998.

[13] Adler, A., Davis, R. “Speech and Sketching for Multimodal Design”. Proc. 9th International Conference on Intelligent User Interfaces, pp.214-216, 2004.

[14] Milward, D., Beveridge, M. A. “Ontologies and the structure of dialogue”. Proc. 8th Workshop on the Semantics and Pragmatics of Dialogue, 2004.

[15] Haya, P. A., Montoro, G., Alamán, X. “A prototype of a context-based architecture for intelligent home environments”, Proc. International Conference on Cooperative Information Systems (CoopIS 2004). pp. 477-491, 2004.

[16] Montoro, G., Haya, P. A., Alamán, X., López-Cózar, R., Callejas, Z. “A proposal for an XML definition of a dynamic spoken interface for ambient intelligence”, International Conference on Intelligent Computing (ICIC 06), pp. 711-716, 2006.

ENTRENAMIENTO DE MODELOS DE LENGUAJE CONEXIONISTAS CON GRANDES VOCABULARIOS

Francisco Zamora[*]

Universitat Jaume I
Dep. de Llenguatges i Sistemes Informàtics
fzamora@guest.uji.es

Salvador España, Jorge Gorbe, María José Castro

Universidad Politécnica de Valencia
Dep. de Sistemas Informáticos y Computación
{sespana, jgorbe, mcastro}@dsic.upv.es

RESUMEN

Las redes neuronales artificiales (RNAs) demuestran su utilidad en ámbitos muy diversos, especialmente en reconocimiento de formas. En particular, los modelos conexionistas resultan especialmente interesantes para el modelado de lenguaje como alternativa a los modelos de tipo estadístico (por ejemplo, n-gramas estadísticos).

No obstante, los modelos de lenguaje conexionistas basados en RNAs presentan algunos problemas de entrenamiento conforme crece el vocabulario del lenguaje a modelar, debido al aumento en el número de parámetros de las redes. El presente trabajo aborda el problema de entrenamiento de RNAs grandes a través de una herramienta que hemos desarrollado, *April*, que implementa eficientemente el entrenamiento de las RNAs.

Keywords: redes neuronales, modelado de lenguaje, n-gramas, grandes vocabularios, entrenamiento eficiente

1. INTRODUCCIÓN

El objetivo de un modelo de lenguaje en un sistema de reconocimiento estadístico de formas consiste básicamente en estimar la probabilidad de una secuencia de palabras $W = w_1, w_2, \ldots, w_{|W|}$. Los modelos de lenguaje de tipo estadístico calculan la probabilidad a priori $P(W)$ realizando la siguiente descomposición:

$$P(W) = \prod_{i=1}^{|W|} P(w_i|w_1^{i-1}) \quad (1)$$

Los modelos de n-gramas [8] simplifican la probabilidad de aparición de una palabra, dadas todas las anteriores, a una historia que sólo considera las $n-1$ palabras anteriores:

$$P(W) = \prod_{i=1}^{|W|} P(w_i|w_1^{i-1}) \approx \prod_{i=1}^{|W|} P(w_i|w_{i-n+1}^{i-1}) \quad (2)$$

Esta probabilidad se puede estimar a partir de un corpus, por lo que se puede obtener de forma automática sin necesidad de introducir conocimiento de tipo deductivo de la lengua para la cual se quiere obtener el modelo de lenguaje. Una de las dificultades de estos modelos consiste en asignar probabilidades a combinaciones de palabras que no hayan aparecido en el corpus. Este problema es inevitable, por lo que es necesario usar métodos de suavizado [5].

Dada la gran utilidad de estos modelos, diversos autores se han planteado su modelización mediante RNAs [9, 13, 2, 4, 7, 10]. Este nuevo paradigma presenta ciertas ventajas e inconvenientes. Entre sus ventajas destacamos el hecho de que un modelo de n-gramas calculado mediante RNAs no necesita aplicar ningún tipo de suavizado explícitamente, puesto que las RNAs interpolan la función que aproximan ante entradas desconocidas.

2. PROBLEMAS ENTRENANDO RNAs GRANDES

El mayor problema de los n-gramas conexionistas se encuentra en la gran cantidad de parámetros a estimar, debido al tamaño de las redes que pueden llegar a aparecer. Si queremos calcular un modelo de n-gramas, la red necesitará tener $(n-1)m$ unidades de entrada, siendo m el número de unidades necesarias para codificar una palabra del corpus. Esto hace que la codificación del vocabulario tenga una importancia decisiva en el coste del cálculo de los modelos. En diversos trabajos previos se han utilizado diversas técnicas para codificar el vocabulario de la tarea [12].

La codificación más sencilla de todas es la *local*, que consiste en usar como representación de cada palabra un vector con tantas componentes como palabras tiene el vocabulario. Este vector tendrá todas las componentes a 0 menos una de ellas que estará a 1 y que indica qué palabra del vocabulario representa dicho vector.

Por otro lado existe la codificación *distribuida*, que consiste en utilizar vectores con un tamaño inferior a la talla del vocabulario, de manera que se pueden decidir formas muy diversas de codificar las palabras. Una forma automática de obtener esta codificación es mediante RNAs como la de la figura 1, donde se utilizan redes que tienen tantas entradas y salidas como palabras del vocabu-

[*]Este trabajo ha sido parcialmente financiado por el proyecto de la Generalitat Valenciana GV06/302.

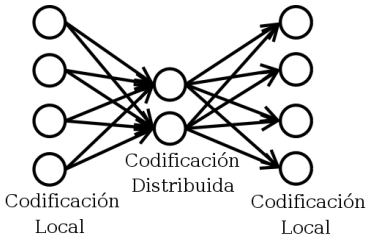


Figura 1. *Topología de red para calcular automáticamente la codificación distribuida de un vocabulario, en este caso con 4 palabras de vocabulario y 2 unidades de codificación.*

lario (Ω), y una capa oculta relativamente pequeña, típicamente $\log_2 \Omega$. Es aconsejable disponer un par de capas ocultas más, una entre la entrada y la capa oculta original, y otra entre esta última y la salida[3]. La red se entrena situando en la entrada y en la salida la codificación local de cada una de las palabras, de manera que el patrón de entrada y de salida deseada es el mismo. Una vez está entrenada la red, la activación de la capa oculta con cada una de las palabras del vocabulario sirve como codificación distribuida.

Otro de los factores decisivos en el entrenamiento de redes con muchos parámetros es el alto coste computacional del algoritmo de entrenamiento. Para poder reducir el tiempo de los entrenamientos hemos desarrollado la herramienta *April*, con una implementación muy eficiente del algoritmo de retropropagación del error con *momentum* [14] y con un gran número de utilidades que permiten manipular conjuntos de datos, haciendo mucho más fácil la obtención de los pares de patrones entrada/salida de las RNAs, además de conseguir un uso reducido de memoria. Más adelante comentaremos el coste temporal de los experimentos, y cómo esta herramienta nos ha ayudado a reducirlo.

3. MODELO DE N-GRAMAS CONEXIONISTAS

Para poder calcular n-gramas mediante RNAs se construyen redes con una topología muy específica. Primero hay que decidir la forma de codificación del vocabulario, y después el número de unidades que tendrá la capa oculta. Construiremos una red con $(n-1)m$ entradas, donde n es el valor del n-grama y m el número de unidades necesarias para codificar una palabra, de manera que la entrada de la red para la posición j de la secuencia W será $h_j = w_{j-n+1}, \ldots, w_{j-2}, w_{j-1}$, codificando cada w_i con m unidades. La salida de la red tendrá tantas unidades como palabras tenga el vocabulario (Ω), y la capa de salida calculará la probabilidad a posteriori de *todas* las palabras del vocabulario:

$$P(w_j = i|h_j) \qquad \forall i = 1, \ldots, \Omega \qquad (3)$$

3.1. Topología de las redes

Hemos señalado algunos detalles de la topología general de las redes en lo que respecta a las entradas y salidas. Vamos a examinar más características, y algunos problemas de la codificación de las palabras.

La codificación distribuida puede ser obtenida de formas muy diversas, presentando algunos métodos más problemas de entrenamiento que otros. Así, por ejemplo, la técnica de la figura 1 converge muy lentamente para ciertos tamaños de entrada. Pero es posible introducir una capa extra en la red completa, como se ve en la figura 2, con la que se calculan los n-gramas, de manera que esta capa extra realice la codificación. La entrada de la red será la codificación local de las $n-1$ anteriores palabras de la frase, y la salida de la red la probabilidad a posteriori de *todas* las palabras del vocabulario (expresión 3). Por tanto, ahora tendremos una red que tendrá $(n-1)\Omega$ unidades de entrada. Se añadirá una nueva capa de unidades que no tendrán conexiones todas a todas. Esta nueva capa estará dividida en conjuntos de N_p unidades, de manera que cada conjunto se conectará con Ω unidades de entrada, y tendremos $n-1$ conjuntos, de manera que el conjunto i estará asociado con las unidades de entrada que representen a la palabra i*-ésima*. A esta capa la denominaremos capa de *proyección*, y es la capa que realizará la codificación distribuida del vocabulario. Para reducir el número de parámetros y que la codificación de una palabra sea siempre la misma, independientemente de la posición de aparición en la entrada de la red, se ligan las conexiones entre los conjuntos de la capa de proyección [10]. Sin esta ligadura de conexiones, la convergencia de los entrenamientos no está asegurada, ya que aparece una cantidad de parámetros excesiva [12]. La ligadura de pesos en las redes es una nueva característica que hemos tenido que añadir a la herramienta *April* para poder llevar a cabo estos experimentos [3] [1].

4. EXPERIMENTACIÓN

Los experimentos se han realizado utilizando una partición del corpus Dihana [1], utilizando la herramienta *April*, desarrollada en nuestro grupo de investigación, para entrenar los modelos neuronales. Los parámetros utilizados para realizar los entrenamientos se muestran en la tabla 1, los detalles de la partición del corpus en conjuntos de entrenamiento, validación y test aparecen en la tabla 2.

Se ha realizado un entrenamiento de las RNAs durante 500 épocas, guardando la red con menor MSE en el conjunto de validación. La figura 3 muestra un ejemplo de la evolución del MSE durante las primeras 40 épocas en el entrenamiento de la red para 4-gramas. El tiempo empleado por ciclo en este entrenamiento ha sido de aproximadamente 18,9 segundos. Este tiempo incluye el coste de

[1] Para ligar los pesos, primero se ajustan los pesos mediante el algoritmo de retroprogación del error, y después, cada conjunto de pesos ligados entre sí se sustituye por la media del conjunto.

Parámetro	Valor
Factor de aprendizaje (η):	0,005 calculado en pruebas preliminares
Momentum (μ):	0,002 calculado en pruebas preliminares
Weight decay (λ):	0,00003 calculado en pruebas preliminares
Función de activación de la capa oculta (f_o):	$tanh$
Función de activación de la capa de salida (f_s):	$softmax$
Función de error (MSE):	*error cuadrático medio*
Tamaño de una palabra en la entrada (Ω):	430 unidades
Tamaño de capa de proyección (N_p):	10 unidades
Tamaño de la capa oculta (N_o):	512 unidades
Valor n de los n-gramas:	$2, 3, 4$

Tabla 1. *Parámetros de los entrenamientos.*

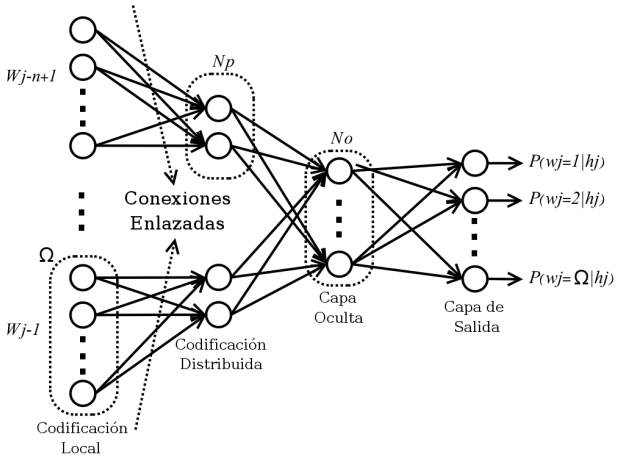


Figura 2. *Topología de red para calcular n-gramas conexionistas.*

	Palabras	Frases	Patrones
Entrenamiento:	3695	460	4152
Validación:	774	100	874
Test:	3185	427	3610

Tabla 2. *Datos relativos a la partición del corpus Dihana utilizada en los experimentos, con un vocabulario de 430 palabras.*

un ciclo completo de las muestras de entrenamiento y de validación, y está muy por debajo de los tiempos que podríamos obtener con herramientas como *SNNS* [15], que en experimentos similares [14] ha llegado a ser hasta 5 veces más lenta. Además, la herramienta *April* ha facilitado también el coste de la preparación de los entrenamientos gracias a las distintas utilidades que posé para manipular este tipo de datos y crear patrones de entrada/salida que hacen un uso reducido de memoria.

Una vez entrenada la red, hemos calculado la perplejidad con el conjunto de test. Los modelos estadísticos de n-gramas han sido obtenidos mediante el *SRI Language Modeling Toolkit* [11] para poder comparar las perplejidades, utilizando la opción que admite proporcionar todo el vocabulario del corpus. Hemos utilizado los conjuntos de entrenamiento y validación para calcular los parámetros mediante el *SRI*.

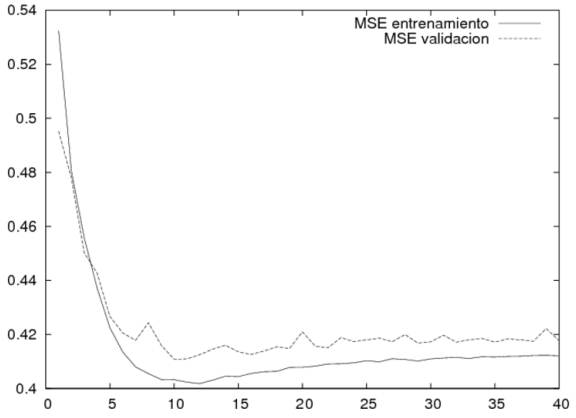


Figura 3. *Detalle de la gráfica de la evolución del MSE durante el entrenamiento del modelo de 4-gramas. La mejor red fue encontrada en la época 10.*

5. CONCLUSIONES

Podemos ver en la tabla 3 que la perplejidad del modelo conexionista es menor en la mayoría de los casos. A pesar de eso, debemos ser precavidos puesto que no

Tipo de modelo	2-gramas	3-gramas	4-gramas
n-gramas conexionistas	26,81	18,96	15,22
n-gramas estadísticos (SRI)	22,35	20,19	20,62

Tabla 3. *Resultados de perplejidad para la tarea Dihana con n-gramas conexionistas y estadísticos, calculados con el SRI. Tarea Dihana.*

existe una relación estricta entre las perplejidades y los resultados generados por un sistema de reconocimiento de habla [6]. Por ello nos planteamos como trabajo futuro la utilización de este modelo en un sistema de reconocimiento del habla para realizar mediciones de las tasas de error reales.

Para lograr reducir el coste temporal de los experiments hemos necesitado emplear un gran esfuerzo en implementar algoritmos y estruturas de datos eficientes. Después de todo este esfuerzo ha surgido la herramienta *April* con la que hemos realizado este trabajo.

6. BIBLIOGRAFÍA

[1] José-Miguel Benedí et al. Design and acquisition of a telephone spontaneous speech dialogue corpus in Spanish: DIHANA. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 1636–1639. European Language Resources Association, 2006.

[2] Yoshua Bengio, Rejean Ducharme, and Pascal Vincent. A neural probabilistic language model. In *NIPS*, pages 932–938, 2000.

[3] C. M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1996.

[4] María José Castro and Federico Prat. New Directions in Connectionist Language Modeling. In *Computational Methods in Neural Modeling*, volume 2686 of *Lecture Notes in Computer Science*, pages 598–605. Springer-Verlag, 2003.

[5] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 310–318, 1996.

[6] P. Clarkson and A. Robinson. The applicability of adaptive language modelling for the broadcast news task. In Proceedings International Conference on Spoken Language Processing, 1998.

[7] A. Emami and F. Jelinek. Exact Training of a Neural Syntactic Language Model. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 245–248, 2004.

[8] F. Jelinek. Statistical methods for speech recognition. *MIT PRess, Cambridge, MA, USA*, 1997.

[9] M. Nakamura and K. Shikano. A study of English word category prediction based on neural networks. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 731–734, 1989.

[10] Holger Schwenk and Jean-Luc Gauvain. Training neural network language models on very large corpora. *HLT/EMNLP*, pages 201–208, 2005.

[11] A. Stolcke. Srilm – an extensible language modeling toolkit. In Proceedings International Conference on Spoken Language Processing., 2002.

[12] Salvador Tortajada Velert and María José Castro Bleda. Diferentes aproximaciones a la codificación del vocabulario en modelado de lenguaje conexionista. *En Actas de las IV Jornadas de Tecnologías del Habla*, 2006.

[13] W. Xu and A. Rudnicky. Can Artificial Neural Networkds learn Language Models? *Proceedings of the 6th International Conference in Spoken Language Processing*, 2000.

[14] Francisco Zamora Martínez. Implementación eficiente del algoritmo de retropropagación del error con *momentum* para redes hacia delante generales. *Proyecto Final de Carrera.* Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, 2005.

[15] Andreas Zell, Niels Mache, Ralf Huebner, Michael Schmalzl, Tilman Sommer, and Thomas Korb. SNNS: Stuttgart Neural Network Simulator. User manual Version 4.1. Technical report, Stuttgart, 1995.

INCORPORATING SEMANTIC KNOWLEDGE TO THE LANGUAGE MODEL IN A SPEECH UNDERSTANDING SYSTEM

Sergio Grau, Encarna Segarra, Emilio Sanchis, Fernando García, Lluís F. Hurtado

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València, E-46022 València, Spain
{sgrau,esegarra,esanchis,fgarcia,lhurtado}@dsic.upv.es

ABSTRACT

The majority of the understanding systems follows an architecture based on two modules, a speech recognition module and an understanding module. Usually, only syntactic restrictions are incorporated to the speech recognition module through the language model and the semantic restrictions are incorporated in the understanding module. In this work, we present an approach to language understanding where the semantic knowledge involved in the understanding process is incorporated through an adequate definition of the language model of the automatic speech recognition module. Then, both the recognition and understanding processes incorporate semantic knowledge. An evaluation of the behavior of the proposed understanding system in the framework of a dialog system is also presented. The results show that the use of semantic information in the language model of the speech recognizer provides for the best performance.

1. INTRODUCTION

In many systems of human-machine interaction, the understanding process is one of the most important parts. This is the case of spoken dialog systems in which the information that must be extracted from the user utterances is not the exact sequence of words, but the meaning of the utterance as well as the specific values that appear in it.

Although approaches to language understanding have traditionally used hand-built semantic rules to detect keywords that are used to fill slots in a frame, other approaches that are based on the use of stochastic models have been developed. The BNN-HUM [1], the AT&T-CHRONUS [2], and the LIMSI-ARISE [3] [4] are some examples of the use of Hidden Markov Models and N-gram models to stochastically model the understanding process from training data. There are also other statistical approaches based on classification, transduction, and grammatical inference techniques: [5], [6], [7], [8] and [9]. These stochastic approaches tackle the understanding process as a problem of transduction of the input sentence into a semantic representation. They can model the variability of the language from real data and take into account the possible sources of error.

An interesting point of study in the speech recognition/understanding system is how to apply the syntactic/semantic restrictions during the decoding process. The classical approach considers a language model of words (typically bigrams or trigrams) during the recognition process, and in a second phase, the sentence (or n-best sentences or a graph of words) obtained is analyzed by the understanding module in order to obtain the semantic representation of the input utterance. However, there is another possibility: to use semantic restrictions during the recognition process in order to guide the search not only by the syntactic restrictions but also by the semantic ones.

In this work, we present an approach to language understanding where the semantic knowledge involved in the understanding process is incorporated through an adequate definition of the language model of the automatic speech recognition module. Then, both the recognition and understanding processes incorporate semantic knowledge. We also present an integrated speech recognition/understanding system where the system makes both the recognition and the understanding processes in just a single step. This approach has been applied to the recognition/understanding module of the DIHANA dialog system, which answers queries by telephone about railway timetables and prices in Spanish [10]. Some experimental results are presented.

In section 2, we describe the DIHANA task and the semantic representation designed for it. In section 3, we present our approach to the understanding process. In section 4, we evaluate the performance of the system; and, in section 5, we present some conclusions.

2. THE DIHANA TASK AND THE SEMANTIC REPRESENTATION

One of the objectives of this project was the acquisition of a corpus of dialogs. The DIHANA task consists of a telephone-based information service for trains in Spanish. A set of 900 dialogs was acquired by using the Wizard of Oz technique. Two hundred and twenty-five naive speakers collaborated in the acquisition of dialogs corresponding to different scenarios. Each one produced 4 dialogs. Three scenarios were defined: timetables for a one-way trip or a two-way trip, prices, and services. The number of user turns was 6,280 and the vocabulary was 823.

As in many other dialog systems [3], the semantic representation chosen for the task is based on the concept of frame. Therefore, the understanding module generates one or more frames with their corresponding attributes as output. In this task, we identified eight concepts. Some of them are: DEPART-TIME, ARRIVAL-TIME, PRICE, etc. Each concept has a set of attributes associated to it (ORIGIN, DESTINATION, DEPART-TIME, ARRIVAL-TIME, TRAIN-TYPE, etc). This set represents the restrictions that the user can place on each concept in an utterance.

3. THE UNDERSTANDING SYSTEM

We propose an understanding system that works in two phases. The first phase consists of a transduction of the input sentence in terms of an intermediate semantic language. In the second phase, a set of rules transduces this intermediate representation in terms of frames. As the intermediate language is close to

the frame representation, this phase only requires a small set of rules to construct the frame. This second phase consists of the following: the deletion of irrelevant segments of the input sentence, the reordering of the relevant concepts and attributes that appeared in the user sentence following an order which has been defined a priori, the instantiation of certain task-dependent values, etc.

In order to represent the meaning of the sentences in terms of the intermediate semantic language, a set of 31 semantic units was defined. Some of them are: *query*, *affirmation*, <departure_hour> <price>, *origin_city*, *destination_city*, *departure_hour*, *departure_date*, *depart_marker*, *arrival_marker*. (We used < and > to distinguish concepts from attributes).

Each semantic unit represents the meaning of words (or sequences of words) in the sentences. For example, the semantic unit $query$ can be associated to "can you tell me", "please tell me", "what is", etc. This way, an input sentence (sequence of words) has a semantic sentence (sequence of semantic units) associated to it, and there is an inherent segmentation.

In this work, we propose two approaches to carry out the first phase of this understanding process. In the first one, the decoupled approach, an automatic speech recognition module produces a sequence of words, and then an understanding module translates it to a sequence of semantic units. In the second one, the integrated approach, the understanding is made through the incorporation of the syntactic and semantic knowledge into the automatic speech recognition module. This module generates, not only the recognized sentence, but also the corresponding sequence of semantic units.

From the sequence of semantic units, the second phase of the understanding process is applied, and the corresponding frame is obtained through a set of rules.

3.1. The decoupled approach

In this approach [6], two kinds of models must be learnt from a training set of semantically tagged and segmented sentences: a semantic model that represents the concatenations of semantic units, and a model for each semantic unit that represents the language of sequences of words associated to that semantic unit.

In order to learn these stochastic models, a set of sequences of semantic units associated to the input sentences, as well as the corresponding association of segments of words to the semantic units must be available. That is, let W be the vocabulary of the task, and let V be the alphabet of semantic units; the training set is a set of pairs (u,v) where:

$$u = u_1, u_2 \dots u_n, u_i = w_{i_1} w_{i_2} \dots w_{i_{|u_i|}},$$
$$w_{i_j} \in W,\ i = 1, \dots, n,\ j = 1, \dots, |u_i|$$
$$v = v_1 v_2 \dots v_n,\ v_i \in V,\ i = 1, \dots, n$$

Each sentence from W has an associated pair *(u,v)*, where *v* is the sequence of semantic units and *u* the sequence of segments in which the original sentence has been divided. An example with a sentence, the associated sequence of semantic units, and the corresponding segmentation is shown in Figure 1.

u_1: I would like u_2: the train timetables u_3: from Valencia u_4: to Barcelona	v_1: query v_2: <departure_hour> v_3: origin_city v_4: destination_city
Input pair (u,v) = ($u_1 u_2 u_3 u_4$,$v_1 v_2 v_3 v_4$)	
Output v = query <departure_hour> origin_city destination_city	

Figure 1. An example of a pair *(u,v)*.

When a training corpus is available, the learning of the sequential translator is carried out through the learning of two models: a model for the semantic language and a set of models (with one model for each semantic unit v_i).

For the understanding process, all the models must be combined in order to take advantage of all the syntactic and semantic restrictions. To do that, in the stochastic automaton for the semantic language, each state (which is associated to each semantic unit) is substituted by the corresponding stochastic automaton, (which represents the sequences of words associated to that semantic unit). The understanding process is performed using the Viterbi algorithm, which supplies the best path in the integrated model. This path not only gives the sequence of semantic units, but it also gives the segmentation associated to it.

3.2. The integrated approach

In this approach, the understanding model represents the semantic knowledge involved in the understanding process through an adequate definition of the language model of the automatic speech recognition module. Then, the recognition and understanding processes are performed at the same time by a single module. In this approach, not only syntactic restrictions, but also semantic restrictions are applied through the language model during the speech recognition process. This language model is learnt using the Morphic Generator Grammatical Inference (MGGI) methodology [11].

The MGGI methodology is a grammatical inference technique that allows us to obtain a certain variety of regular languages. The application of this methodology implies the definition of a renaming function; that is, each symbol of each input sample is renamed following a given function g. Then, a classical grammatical inference algorithm can be chosen to infer an automaton with the renamed training samples. Finally, the renamed symbols are converted back to the original ones in the obtained automaton. An important characteristic of this methodology is that different definitions of the function g will produce different models. Therefore, we can choose an adequate renaming function depending on the characteristics we want to represent in the model.

In this work, we defined the renaming function g in such a way that it specialized each word in a segment u_i by adding to it information about its semantic unit v_i. Figure 2 shows the application of this renaming function g to the example in Figure 1. We used # to concatenate each word with the name of its semantic unit.

u_1: I would like u_2: the train timetables u_3: from Valencia u_4: to Barcelona	v_1: query v_2: <departure_hour> v_3: origin_city v_4: destination_city
Input pair (u,v) = ($u_1 u_2 u_3 u_4$,$v_1 v_2 v_3 v_4$)	
Output g((u,v)) = I#query would#query like#query the#<departure_hour> train#<departure_hour> timetables#<departure_hour> from#origin_city Valencia#origin_city to#destination_city Barcelona#destination_city	

Figure 2. An example of the use of the renaming function.

The training corpus labeled as described above can be used as the language model in the speech recognition process. In this way, both syntactic restrictions, and semantic constraints are considered in the search space of the recognition process. As a result, the recognition process not only gives the sequence of words, but also the semantic label associated to each word.

Therefore, we can obtain the segmentation associated to the sentence by including in the same segment all the consecutive words labeled with the same semantic label, as shown in Figure 3.

Recognizer Output = I#query would#query like#query the#<departure_hour> train#<departure_hour> timetables#<departure_hour> from#origin_city Valencia#origin_city to#destination_city Barcelona#destination_city	
Output = (u,v)	
u_1: I would like	v_1: query
u_2: the train timetables	v_2: <departure_hour>
u_3: from Valencia	v_3: origin_city
u_4: to Barcelona	v_4: destination_city

Figure 3. Obtaining the pair *(u,v)* from the output of the speech recognition process.

4. EXPERIMENTAL RESULTS

We used the CMU Sphinx-II recognizer [12] to decode the user utterances. We trained semi-continuous acoustic models from 3,600 telephone-quality utterances acquired in the DIHANA project. The models were trained using a set of 25 phones plus silence for the Spanish.

In this section, we describe the results of the evaluation of our understanding systems. Three different understanding systems were tested:

- (Trigrams+UM) A decoupled understanding system with a speech recognizer that used a trigram of words as language model, and the understanding module (UM) based on the finite-state models presented in subsection 3.1.
- (MGGI_LM) An integrated understanding system that incorporates both the syntactic and semantic knowledge into the language model (see subsection 3.2).
- (MGGI_LM+UM) A decoupled understanding system with the MGGI_LM module used as speech recognition module and the understanding module (UM) presented in subsection 3.1.

The experiments were performed using the user turns of the 900 dialogs obtained through the Wizard of Oz technique. The characteristics of the transcribed corpus are shown in Table 1. The orthographic transcriptions of the user turns were semi-automatically segmented and labelled in terms of semantic units. The characteristics of this labelled and segmented corpus are shown in Table 2.

Table 1. Characteristics of the transcribed corpus

Number of turns	6,227
Number of words	47,196
Vocabulary size	823
Average number of words in turn	7.58
Longest turn	55

A cross-validation procedure was used to evaluate the performance of our understanding models. To this end, the experimental set was randomly split into five subsets of 1,246 turns. Our experiment consisted of five trials, each of which had a different combination of one subset (taken from the five subsets) as the test set. The remaining 4,981 turns were used as the training set.

Table 2. Characteristics of the labelled and segmented corpus

Number of semantic segments	18,570
Average number of words per segment	2.54
Highest number of words in a segment	25
Average number of segments per turn	2.98
Highest number of segments in a turn	16
Number of semantic segments without semantic relevance	1,631
Number of semantic units	31
Number of different sequences of semantic segments	1,592

We defined four measures to evaluate the accuracy of the models:

- the percentage of correct sequences of semantic units (%cssu).
- the percentage of correct semantic units (%csu).
- the percentage of correct frame names (%cfn); i.e., the percentage of resulting frame names that are exactly the same as the corresponding reference frame names.
- the percentage of correct frame slot names (frame name and its attribute names) (%cfsn).

The measure $\%csu$ allows us to evaluate the first phase of our understanding system. This measure is the concept accuracy and is calculated in the same way as the word accuracy used in speech recognition. The measures $\%cfn$ and $\%cfsn$ evaluate the overall understanding system. As shown in Section 2, the semantic representation of a sentence is made by one or more frames. The measure $\%cfn$ considers the output to be correct only when the obtained frame (frame name and its attribute names) is the same as the reference one. The measure $\%cfsn$ is the frame slot accuracy, that is, the number of correctly understood units (frame name and its attribute names) divided by the number of units in the reference.

A first experiment was performed using the correct transcriptions of the sentences in order to evaluate the understanding models when working under the best conditions of the input. The understanding model used in these experiment was UM, presented in subsection 3.1. We used a bigram for the semantic model and also for the semantic unit model for each semantic unit; the smoothing technique for bigrams was back-off with Good-Turing discounting. The results, which are reported as Text+UM, are shown in Table 3.

Experiments taking into account the recognition and understanding processes were also performed. Table 3 shows the results for the three understanding systems, Trigrams+UM, MGGI_LM+UM, and MGGI_LM, in terms of the understanding measures and in terms of the word accuracy (WA) of the recognition process.

Table 3. Understanding results.

LM	WA	%cssu	%csu	%cfn	%cfsn
Text+UM	-	79.6	90.4	90.2	93.9
Trigrams+UM	77.3	63.0	79.0	73.2	83.9
MGGI_LM+UM	77.5	64.2	79.5	74.1	84.5
MGGI_LM	77.5	62.3	78.0	71.2	82.5

The word accuracy results of the recognizer that used syntactic-semantic language models (MGGI_LM) slightly outperformed the corresponding results of the recognizer that used word trigrams. That means that the use of the semantic restrictions in the recognition process can be useful for speech recognition in dialogue systems tasks. On the other hand, the integrated model did not outperform the decoupled approaches in

terms of speech understanding results. However, using the recognized sequence of words from the MGGI_LM as input for the understanding module improved the overall results of the speech understanding system.

5. CONCLUSIONS AND FUTURE WORK

We have presented an approach to incorporate semantic knowledge to the language model of a speech recognition module within the framework of a dialogue system. We have studied the use of this semantic information in order to improve not only the word accuracy of the automatic speech recognition module but also the overall speech understanding results. For this reason, three different understanding systems have been tested. The best results were obtained using the sequence of words recognized by the MGGI_LM speech recognizer as input for the understanding module. Therefore, when syntactic and semantic information is used through the language model in the recognition process, the results outperform the understanding results when word trigrams are used.

6. ACKNOWLEDGEMENTS

Work partially supported by the Spanish CICYT under contract TIN2005-08660-C04-02

7. REFERENCES

[1] R. Schwartz, S. Miller, D. Stallard, y J. Makhoul, "Language understanding using hidden understanding models," in *Proc. ICSLP'96*, Philadelphia (USA), Oct. 1996, pp. 997–1000.

[2] E. Levin y P. Pieraccini, "Concept-based spontaneous speech understanding system.," in *Proc. of Eurospeech'95*, 1995, pp. 555–558.

[3] W. Minker, A. Waibel, y J. Mariani, *Stocastically-Based Semantic Analysis*, Kluwer Academic Publishers, Boston, 1999.

[4] H. Bonneau-Maynard y F. Lefèvre, "Investigating stochastic speech understanding," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'01)*, 2001.

[5] K. Macherey, F.J Och, y H. Hermann Ney, "Natural Language Understanding Using Statistical Machine Translation," in *Proc. of EUROSPEECH*, September 2001, pp. 2205–2208.

[6] E. Segarra, E. Sanchis, F. García, y L.F. Hurtado, "Extracting semantic information through automatic learning techniques," in *International Journal of Pattern Recognition and Artificial Intelligence*, Salt Lake City (USA), 2002, pp. 16(3):301–307.

[7] Yulan He y S. Young, "A data-driven spoken language understanding system," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'03)*, 2003, pp. 583–588.

[8] S. Gupta, N.K. Bangalore, "Segmenting spoken language utterances into clauses for semantic clasiffication," in *Proc. of ASRU*, 2003, pp. 525–530.

[9] Y. Esteve, C Raymond, y R. Bechet, F. De Mori, "Conceptual Decoding for Spoken Dialog systems," in *Proc. of Eurospeech*, 2003, vol. 1, pp. 617–620.

[10] J.M. Benedí, A. Varona, y E. Lleida, "DIHANA: Sistema de diálogo para el acceso a la información en habla espontánea en diferentes entornos," in *Actas de las III Jornadas en Tecnología del Habla*, Valencia (España), 2004, pp. 141–146.

[11] E. Segarra y L. Hurtado, "Construction of Language Models using Morfic Generator Grammatical Inference MGGI Methodology," in *Proc. of Eurospeech*, 1997, pp. 2695–2698.

[12] Xuedong Huang, Fileno Alleva, Hsiao-Wuen Hon, Mei-Yuh Hwang, y Ronald Rosenfeld, "The SPHINX-II speech recognition system: an overview," *Computer Speech and Language*, vol. 7, no. 2, pp. 137–148, 1993.

EXPERIMENTS IN SPEECH DRIVEN INFORMATION RETRIEVAL FOR SPANISH LANGUAGE

César González-Ferreras, Valentín Cardeñoso-Payo

Departamento de Informática
Universidad de Valladolid
{cesargf,valen}@infor.uva.es

ABSTRACT

The paper reports on the evaluation of a system that allows users to search information using spoken queries. The front end is a large vocabulary continuous speech recognizer which translates the query from speech to text and puts it through an information retrieval engine to retrieve the set of relevant documents for that query. The system is designed for Spanish language. The performance of the system was evaluated using the test suites of CLEF, which is an evaluation forum similar to TREC. 10 different speakers were recorded reading the queries. Results of different experiments are reported. Best results were obtained with a language model of 60,000 words and medium length queries: loss of precision of 23.79% (compared to using perfect transcription of the queries) with a WER of 18.4%. Overall, these results are encouraging and provide a solid basis for the feasibility of building speech driven information retrieval systems.

1. INTRODUCTION

Nowadays web contents are an important source of information. We use the web to access different kinds of information, both for leisure and business. There is also a growing interest in providing speech access to web contents. Using speech is natural for most of the people, and thus it can provide a more usable interaction. Mobile devices allow web access anytime and everywhere, and would also benefit from speech interaction, increasing their usability, which is limited because of their small displays.

Different approaches have been proposed to access web contents using speech. One of the most natural and effective is using speech as the input to an information retrieval engine. In fact, search is one of the most used ways to access the world wide web. Search can also help to deal with the limitations of the speech channel, which does not allow to send much information over it.

This paper reports on the evaluation of a system that provides spoken access to an information retrieval engine. The front end is a large vocabulary continuous speech recognizer (LVCSR) which translates the query from speech to text and puts it through an information retrieval (IR) engine to retrieve the set of relevant documents for that query. The system is designed for Spanish language.

The performance of the system was evaluated using the test suites of CLEF, which is an evaluation forum similar to TREC. Results of different experiments are reported. The impact of vocabulary size was tested first, comparing results of trigram language models (LM) with 20,000 and 60,000 words vocabulary sizes. Then, the influence of query length was measured, using medium length and short queries. Best results were obtained with LM of 60,000 words and medium length queries: loss of precision of 23.79% (compared to using perfect transcription of the queries) with a word error rate (WER) of 18.4%.

The structure of the paper is as follows: section 2 presents some related work; section 3 explains the system in detail; section 4 describes the experiments and shows the results; in section 5 we discuss about factors that affect system performance; section 6 presents conclusions and future work.

2. RELATED WORK

There has been little work on speech driven information retrieval. Some experiments have been carried out in English, Chinese and Japanese. All the experiments used a similar methodology: a standard IR test collection (designed to evaluate IR systems using text queries under standard and comparable conditions) is used; some speakers reading the queries are recorded; finally, system performance is evaluated and compared with results obtained using text queries.

First speech driven information retrieval experiments have been described in [1]. Dragon's LVCSR system with 20,000 and 30,000 words vocabulary and INQUERY probabilistic IR system were used. Two test collections were used: TIPSTER and Boston Globe. Experiments were reported using long, medium length and short queries. Results showed that increasing WER reduces precision and that long queries are more robust to recognition errors

This work has been partially supported by *Ministerio de Ciencia y Tencología* under project number TIC2003-08382-C05-03 and by *Consejería de Educación de la Junta de Castilla y León* under project number VA053A05.

than short queries. Based on these experiments, an interactive vocal information retrieval system was designed [2]. The system had the following components: a dialogue manager, a probabilistic IR system, a document summarization system and a document delivery system.

Experiments in Chinese have been presented in [3]. Two different speech recognizers were used: Microsoft SAPI 5.0 LVCSR (50,000 Chinese multi-character words) and HTK-based syllable speech recognition system. The IR system used the vector space model with tf-idf and pseudo relevance feedback. TREC5 and TREC6 databases were used as test collections. Retrieval performance on mobile devices with high-quality microphones (for example PDA) was satisfactory, although the performance over cellular phone was significantly worse.

Experiments in Japanese have been reported in [4]. Julius LVCSR with trigram LM of 20,000 and 60,000 words was used. The IR system was based on the probabilistic model. NTCIR-3 test collection was used. Results showed that using target document collection for language modeling and using bigger vocabulary size improved system performance. More experiments using the same test collection have been described in [5]. Julius and Spojus LVCSRs, using phone-based and syllable-based models were used. The IR system used the vectorial model and tf-idf. Different techniques for combining multiple LVCSRs using SVM learning were evaluated. Results showed an improvement both in speech recognition and retrieval accuracies. They also reported better results with a LM of larger vocabulary size.

3. SYSTEM OVERVIEW

The objective of the system is to retrieve all the documents relevant to a given spoken query. The architecture of the system is shown in figure 1. First, the user makes a query using speech. Then, the speech recognizer transcribes the spoken query into text. Finally, the information retrieval engine finds the list of documents relevant to that query. In the following sections we describe in detail both speech recognition and information retrieval systems.

3.1. Speech Recognition

For speech recognition we used SONIC, the University of Colorado large vocabulary continuous speech recognizer [6]. SONIC is based on continuous density hidden Markov model (CDHMM) technology and implements a two-pass search strategy using token-passing based recognition.

The triphone acoustic models were HMMs with associated gamma probability density functions to model state durations. Mel Frequency Cepstral Coefficients (MFCC) were used, with a standard 39-dimensional feature vector (12 MFCCs and normalized log energy along with the first and second order derivatives). We trained the acoustic models using Albayzin corpus [7] (13600 sentences read by 304 speakers).

Two trigram LMs with different vocabulary size were built: 20,000 and 60,000 words. The target document collection was used to train the language model, because this can result in an adaptation of the LVCSR to the given task and provides better system performance [4]. EFE94 document collection is composed of one year of newswire news (511 Mb). CMU-Cambridge statistical language modeling toolkit was used with Witten Bell discounting [8].

3.2. Information Retrieval

A modified version of an information retrieval engine developed for Spanish language was used [9]. It is based on the vector space model and term frequency-inverse document frequency (tf-idf) weighting scheme [10]. A stemming algorithm was used to reduce the dimensionality of the space and a stop word list to remove function words.

Given a query q, the similarity of that query with each document d_i in the document collection is calculated as follows:

$$sim(d_i, q) = \sum_{t_r \epsilon q} w_{r,i} \times w_{r,q} \quad (1)$$

$$w_{r,i} = (1 + log(tf_{r,i})) \times log\left(\frac{N}{df_r}\right) \quad (2)$$

$$w_{r,q} = log\left(\frac{N}{df_r}\right) \quad (3)$$

Where $tf_{r,i}$ represents the number of times a term t_r appears in the document d_i; df_r is the number of documents in the collection in which the term t_r appears; N is the number of documents in the collection.

4. EXPERIMENTS

In order to evaluate the performance of the system we made some experiments based on Cross-Language Evaluation Forum (CLEF) test-suites [11]. CLEF organizes evaluation campaigns in a similar way to TREC. Its aim is to develop an infrastructure for the testing and evaluation of information retrieval systems operating on European languages, in both monolingual and cross-language contexts. We have expanded CLEF 2001 Spanish monolingual IR test-suite to include spoken queries. In the following sections the experimental set-up is described and the results are presented.

4.1. Experimental Set-up

CLEF 2001 test suite includes a document collection, a set of topics and relevance judgements. The document collection has 215,738 documents of the year 1994 from EFE newswire agency (511 Mb). There are 49 topics (topic numbers 41-90) and each of them has three parts:

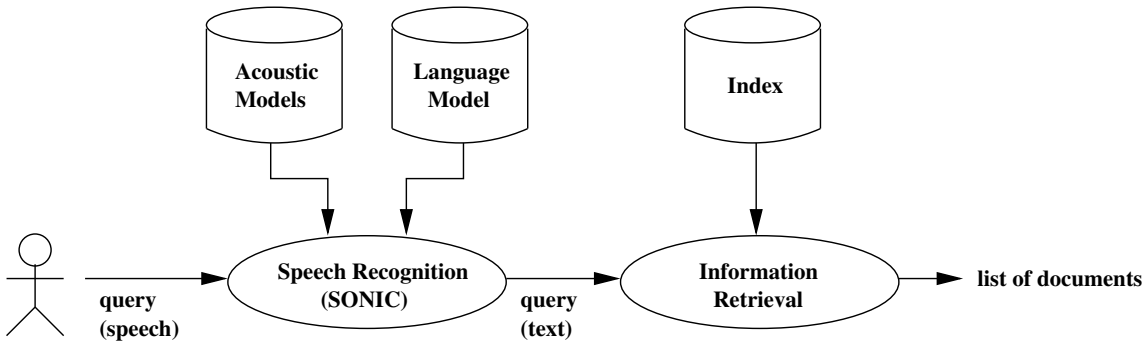


Figure 1. *Architecture of the system.*

```
<top>
<num>67</num>
<title>Ship Collisions</title>
<desc>Find information on the number
of people injured or killed in
collisions between ships.</desc>
<narr>Relevant documents report
information on the number of victims
(dead or injured) of collisions
between ships or maritime vessels of
all types.  Documents that speak of
victims without providing figures are
not relevant.</narr>
</top>
```

Figure 2. *Sample query (translated from Spanish).*

a brief title statement, a one-sentence description and a more complex narrative specifying the relevance assessment criteria (see figure 2). Relevance judgements determine the set of relevant documents for each topic, and were created using pooling techniques.

Given the topics, we tested our system using two different queries: (1) medium length queries (using description field, with mean length of 16 words, ranging from 5 to 33); (2) short queries (using title field, with mean length of 4 words, ranging from 1 to 8). We recorded queries of 10 different speakers (5 male and 5 female) using a headset microphone under office conditions, at 16 bit resolution and 16 kHz sampling frequency.

4.2. Results

First, spoken queries were processed by the speech recognizer and word error rate was calculated. Then, the best hypothesis was used by the information retrieval engine. The same methodology of CLEF was used to evaluate the results [11]. For each query, the 1000 most relevant documents (sorted by relevance) were retrieved, and using the relevance judgements mean average precision (MAP) was calculated.

Results are shown in tables 1 and 2. In the first table a LM of 20,000 words was used, with an OOV rate of 7.07%. In the second table a LM of 60,000 words was used, with an OOV rate of 2.02%. In each table we compare results using medium length queries (description) and short queries (title). For each speaker, we report the word error rate (WER), the mean average precision (MAP), the loss of MAP compared with using the original text queries (Ploss), and the correlation between WER and loss of MAP (corr).

5. DISCUSSION

We analyzed the impact of LM vocabulary size. Better results were obtained using a LM of 60,000 words, because there were fewer speech recognition errors. Speech driven information retrieval is a task with an open vocabulary, and better results will be obtained with larger vocabulary LM, because of its better vocabulary coverage.

We studied the impact of query length. Medium length queries had better results of precision than short queries, both in text and speech. Moreover, medium length spoken queries had fewer loss of precision than short spoken queries, and thus they seem to be more robust to speech recognition errors.

A correlation between WER and loss of precision was detected. The correlation was stronger in short queries than in medium length queries. The reason is that most of the words in short queries are keywords, so any recognition error has big impact in retrieval results. However, in medium length queries the recognition errors might happen in function words and do not degrade precision.

We analyzed the results of each individual query. Most of the queries had small loss of precision while some of the queries had high loss of precision. It means that in general queries did well, but there were some that did badly. For the experiment with a LM of 60,000 words and medium length queries, 70.61% of the queries had a loss of precision of less than 10% and 9.39% of the queries had a loss of precision between 10% and 30%. However, 11.63% of the queries had a loss of precision more than 90%.

	description				title			
	WER	**MAP**	**Ploss**	**corr**	**WER**	**MAP**	**Ploss**	**corr**
Text		0.4466				0.4373		
Speaker1	24.9%	0.2887	35.36%	0.42	31.4%	0.2540	41.92%	0.76
Speaker2	17.4%	0.3134	29.83%	0.28	23.2%	0.2980	31.85%	0.73
Speaker3	22.7%	0.2980	33.27%	0.22	18.8%	0.3094	29.25%	0.66
Speaker4	23.5%	0.2921	34.59%	0.41	26.1%	0.2795	36.09%	0.59
Speaker5	28.2%	0.3110	30.36%	0.47	23.2%	0.3017	31.01%	0.57
Speaker6	33.6%	0.2404	46.17%	0.50	27.5%	0.2755	37.00%	0.63
Speaker7	15.8%	0.3282	26.51%	0.44	21.3%	0.2928	33.04%	0.63
Speaker8	23.1%	0.3005	32.71%	0.51	26.1%	0.2926	33.09%	0.71
Speaker9	24.4%	0.2926	34.48%	0.36	21.7%	0.2713	37.96%	0.74
Speaker10	27.8%	0.3271	26.76%	0.43	26.1%	0.2911	33.43%	0.75
Mean	24.1%	0.2992	33.00%	0.41	24.5%	0.2866	34.46%	0.66

Table 1. *Results using a LM of 20,000 words for medium length queries (description) and short queries (title). (WER: word error rate; MAP: mean average precision; Ploss: loss of MAP compared with text queries; corr: correlation between WER and Ploss).*

	description				title			
	WER	**MAP**	**Ploss**	**corr**	**WER**	**MAP**	**Ploss**	**corr**
Text		0.4466				0.4373		
Speaker1	18.5%	0.3282	26.51%	0.62	25.1%	0.2938	32.82%	0.79
Speaker2	10.7%	0.3618	18.99%	0.60	12.6%	0.3586	18.00%	0.82
Speaker3	16.5%	0.3514	21.32%	0.29	11.1%	0.3513	19.67%	0.74
Speaker4	17.1%	0.3269	26.80%	0.47	14.5%	0.3362	23.12%	0.85
Speaker5	22.5%	0.3401	23.85%	0.47	15.9%	0.3305	24.42%	0.82
Speaker6	27.7%	0.2950	33.95%	0.56	20.3%	0.3239	25.93%	0.67
Speaker7	8.9%	0.3587	19.68%	0.32	14.5%	0.3176	27.37%	0.62
Speaker8	18.5%	0.3359	24.79%	0.63	15.5%	0.3320	24.08%	0.83
Speaker9	20.9%	0.3236	27.54%	0.41	13.5%	0.3297	24.61%	0.90
Speaker10	22.7%	0.3818	14.51%	0.50	18.4%	0.3305	24.42%	0.75
Mean	18.4%	0.3403	23.79%	0.48	16.1%	0.3304	24.44%	0.71

Table 2. *Results using a LM of 60,000 words for medium length queries (description) and short queries (title). (WER: word error rate; MAP: mean average precision; Ploss: loss of MAP compared with text queries; corr: correlation between WER and Ploss).*

6. CONCLUSIONS

In this paper we have described a system that allows users to retrieve information using spoken queries. We made different experiments in order to evaluate the system performance using a standard IR test suite. The impact of vocabulary size and query length were analyzed. Best results were obtained with a LM of 60,000 words and medium length queries: loss of precision of 23.79% (compared to using perfect transcription of the queries) with a WER of 18.4%. We also reported correlation between WER and loss of precision. Overall, these results are encouraging and provide a solid basis for the feasibility of building speech driven information retrieval systems.

As future work, we will focus on queries with high precision loss. We plan to study the factors that explain their performance decrease and to develop a method to avoid them. Another area of interest is related to the construction of spoken dialog systems, where user interaction could provide valuable feedback to improve the retrieval process.

7. ACKNOWLEDGEMENTS

The authors want to thank Joaquín Adiego and Pablo de la Fuente for their help in the information retrieval part of the system.

8. REFERENCES

[1] J. Barnett, S. Anderson, J. Broglio, M. Singh, R. Hudson, and S. W. Kuo, "Experiments in Spoken Queries for Document Retrieval," in *European Conference on Speech Communication and Technology (Eurospeech)*, 1997.

[2] F. Crestani, "Vocal Access to a Newspaper Archive: Assessing the Limitations of Current Voice Information Access Technology," *Journal of Intelligent Information Systems*, vol. 20, no. 2, pp. 161–180, 2003.

[3] E. Chang, F. Seide, H. Meng, Z. Chen, Y. Shi, and Y. Li, "A System for Spoken Query Information Retrieval on Mobile Devices," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 8, pp. 531–541, November 2002.

[4] A. Fujii and K. Itou, "Building a Test Collection for Speech-Driven Web Retrieval," in *European Conference on Speech Communication and Technology (Eurospeech)*, 2003.

[5] M. Matsushita, H. Nishizaki, S. Nakagawa, and T. Utsuro, "Keyword Recognition and Extraction by Multiple-LVCSRs with 60,000 Words in Speech-driven WEB Retrieval Task," in *International Conference on Spoken Language Processing (ICSLP)*, 2004.

[6] B. Pellom and K. Hacioglu, "Recent Improvements in the CU SONIC ASR System for Noisy Speech: The SPINE Task," in *International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2003.

[7] A. Moreno, D. Poch, A. Bonafonte, E. Lleida, J. Llisterri, J.B. Mariño, and C. Nadeu, "ALBAYZIN Speech Database: Design of the Phonetic Corpus," in *European Conference on Speech Communication and Technology (Eurospeech)*, 1993.

[8] P. Clarkson and R. Rosenfeld, "Statistical Language Modeling Using The CMU-Cambridge Toolkit," in *European Conference on Speech Communication and Technology (Eurospeech)*, 1997.

[9] J. Adiego, P. Fuente, J. Vegas, and M. Á. Villarroel, "System for Compressing and Retrieving Structured Documents," *UPGRADE*, vol. 3, no. 3, pp. 62–69, June 2002.

[10] G. Salton, A. Wong, and C. S. Yang, "A Vector Space Model for Automatic Indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, November 1975.

[11] M. Braschler and C. Peters, "CLEF Methodology and Metrics," in *Workshop of the Cross-Language Evaluation Forum (CLEF)*, 2001.

SISTEMA DE SÍNTESIS ARMÓNICO/ESTOCÁSTICO EN MODO PITCH-ASÍNCRONO APLICADO A CONVERSIÓN DE VOZ

Daniel Erro y Asunción Moreno

Universidad Politécnica de Cataluña, Barcelona

RESUMEN

En el presente artículo se describe un sistema de conversión de voz desarrollado a partir de un sistema de síntesis basado en el modelo armónico/estocástico. A diferencia de otros trabajos similares, no es necesario el funcionamiento pitch-síncrono de la herramienta para que las concatenaciones, modificaciones prosódicas y manipulaciones espectrales se realicen de manera rápida y simple. Los experimentos y comparaciones realizados muestran que el nivel de conversión es altamente satisfactorio a pesar de no usar corpus de entrenamiento paralelos, aunque se pone de manifiesto el compromiso entre el parecido y la calidad de la voz.

1. INTRODUCCIÓN

El propósito de los sistemas de conversión de voz es modificar la voz de un hablante *fuente* de manera que sea percibida como si se tratara de otro hablante *objetivo*, reemplazando las características físicas de la voz sin alterar el mensaje. Las técnicas de conversión de voz encuentran importantes aplicaciones en el campo de la síntesis de habla. En un sintetizador, el habla se genera habitualmente mediante la concatenación de unidades seleccionadas dentro de una base de datos, la cual ha sido construida con anterioridad grabando fragmentos de voz de un locutor cualificado. La inclusión de un sistema de conversión de voz a la salida del sintetizador permite personalizar el sistema sin necesidad de grabar bases de datos de voz de cada uno de los potenciales usuarios. Para ello simplemente se ha de entrenar una función de conversión entre la voz estándar del sintetizador, que será la *fuente*, de la que se dispone de una base de datos completa, y la voz del hablante *objetivo*, del cual se poseen en general pocos datos.

Hay una relación muy cercana entre el sistema de síntesis y el de conversión de voz, hasta el punto de que algunos autores consideran ambos bloques como uno solo. Al convertir una voz en otra no solo se consideran los rasgos espectrales, sino también aspectos prosódicos, y por lo tanto es importante la elección de un sistema de síntesis que permita la manipulación de todas estas características de manera flexible. La calidad de la señal convertida depende en gran medida de la que el sistema de síntesis pueda proporcionar, y se ve limitada por los ruidos de concatenación o discontinuidades espectrales. Por todo ello, es recomendable evaluar el sintetizador y el conversor de voz como un único bloque.

Los sistemas de síntesis basados en TD-PSOLA no son apropiados para la conversión de voz, ya que no asumen un modelo de señal, sino que trabajan directamente con las muestras de audio. Además, son poco resistentes a los ruidos de concatenación de unidades. Existen sistemas basados en LP-PSOLA que ofrecen un buen rendimiento [1, 2], pero se ha observado que a la hora de reconstruir la señal modificada se introducen ruidos derivados de la incoherencia de fase, especialmente cuando se opera sobre el residuo de la codificación LPC. Los sistemas basados en la descomposición armónica o sinusoidal de la señal se han empleado con éxito en tareas de conversión [3, 4], debido a su gran flexibilidad, pero se trata en general de sistemas pitch-síncronos, cuya calidad está condicionada a una correcta detección del inicio y final de cada período de pitch. Todo esto hace pensar en la utilización de un sistema de tipo sinusoidal que permita un mayor control sobre la fase.

En muchos de los trabajos publicados acerca de conversión de voz, se hace uso de un corpus de entrenamiento *paralelo*, es decir, que contiene vectores de parámetros acústicos de los locutores *fuente* y *objetivo* correspondientes al mismo sonido [3, 5]. Para elaborarlo, se acostumbra a grabar las mismas frases pronunciadas por ambos hablantes, alineando convenientemente los fragmentos de sonido equivalentes. Está claro, sin embargo, que no siempre se puede disponer de grabaciones con idénticas frases de cada par de locutores, de modo que recientemente se han propuesto algunas técnicas que permiten resolver este problema [1, 2, 4].

En este trabajo se ha perfeccionado un sistema de síntesis sobre el modelo armónico/estocástico [6] que trabaja con tramas de longitud fija, y se ha implementado un módulo de conversión de voz. Se ha comprobado que su rendimiento es notable aun cuando no se dispone de corpus paralelo de entrenamiento. Esto se debe, en parte, a que aprovecha la enorme flexibilidad del modelo armónico/estocástico para minimizar los errores derivados más propiamente de la reconstrucción de la señal a partir de sus parámetros ya modificados. En la sección 2 se describe el sistema de

síntesis empleado, insistiendo principalmente en los puntos que lo diferencian de otros sistemas parecidos. Se hace especial hincapié en lo referente al tratamiento de las fases de los armónicos. En la sección 3 se explica en detalle el método de conversión de voz. En el marco del proyecto europeo TC-STAR, el sistema completo ha sido evaluado como se detalla en la sección 4, dando lugar a las conclusiones que se exponen en la sección 5.

2. DESCRIPCIÓN DEL SISTEMA DE SÍNTESIS

El modelo armónico/estocástico asume que la señal de voz puede ser representada como la suma de dos componentes: el primero de ellos es una suma de sinusoides armónicamente relacionadas que modelan la parte sonora, y el segundo es un componente de aspecto ruidoso caracterizado por una determinada densidad espectral de potencia, que modela tanto las partes sordas de la señal como cualquier fenómeno de naturaleza no periódica, como puede ser el rozamiento del aire al hablar.

$$s[n] = \sum_{j=1}^{J(n)} A_j[n]\cos\left(\theta_j[n]\right) + e[n] \qquad (1)$$

Ambos componentes cambian a lo largo del tiempo, pero pueden considerarse estables dentro de pequeños intervalos, de modo que la señal es analizada por tramas.

2.1. Análisis y re-síntesis

Se han empleado tramas equiespaciadas centradas en las muestras $n = kN$, con $k = 1, 2, 3\ldots N$ es el número de muestras correspondiente a unos 8 ó 10ms de señal. En adelante, se llamará punto k a la muestra kN.

En cada trama, partiendo de una primera estimación del pitch, se decide si es sorda o sonora. Aunque se puede emplear cualquier método de detección de pitch, en este caso se ha recurrido a la señal electroglotográfica grabada simultáneamente con la voz. Si la k-ésima trama es sonora, se detectan las amplitudes y fases de las sinusoides en el punto k por debajo de 5KHz [7], corrigiendo el valor del pitch si es necesario. La elección de una frecuencia de corte fija se debe a la necesidad de parametrizar las envolventes a la hora de convertir la voz. La parte armónica es interpolada a lo largo de las diferentes tramas [8] y sustraída de la señal original para aislar la parte estocástica, que es analizada en cada trama usando la técnica LPC.

Para resintetizar la señal a partir de sus parámetros, en cada punto k se construye una trama de longitud $2N$ centrada en dicho punto sumando los armónicos detectados con amplitud, frecuencia y fase constantes, y añadiendo la contribución estocástica obtenida al filtrar ruido blanco y gaussiano a través del filtro LPC correspondiente.

$$s^{(k)}[n] = \sum_{j=1}^{J^{(k)}} A_j^{(k)} \cos\left(2\pi j \frac{f_0^{(k)}}{f_s} n + \varphi_j^{(k)}\right) + \sigma[n] * h_{LPC}^{(k)}[n] \qquad (2)$$

con n=–N,…,N–1. Las tramas se solapan entre sí usando ventanas triangulares.

$$s[kN+m] = \left(\tfrac{N-m}{N}\right)\cdot s^{(k)}[m] + \left(\tfrac{m}{N}\right)\cdot s^{(k+1)}[m-N] \qquad (3)$$

para m entre 0 y N–1. La señal resintetizada es prácticamente indistinguible de la original.

2.2. Modificaciones prosódicas

Dado que se están usando tramas de tamaño fijo, se han tenido que desarrollar procedimientos que permiten modificar los parámetros de cada trama de la señal de forma sencilla, sin perder la coherencia de fase entre diferentes tramas. Para ello se considera que las fases medidas son el resultado de sumar un término lineal en frecuencia y la respuesta en fase del tracto vocal, que varía con el tiempo.

2.2.1. Modificación en duración

Para modificar la duración de la señal, basta con cambiar el valor de N en la ecuación (3) para que las variaciones en amplitud y frecuencia se adapten a la nueva escala temporal. No obstante, las fases no se pueden mantener invariantes en cada punto k porque ha variado la distancia entre puntos de análisis, de modo que se rompe la coherencia. Es por eso que hay que corregir el término lineal de fase de modo que solamente la respuesta en fase del tracto vocal se mantenga en cada punto k sin perder la coherencia. Suponiendo que no cambia el tracto vocal y que el pitch varía linealmente entre los puntos k–1 y k, el incremento de fase del primer armónico entre dichos puntos se puede aproximar por medio de la función Ψ:

$$\Psi\left(f_0^{(k-1)}, f_0^{(k)}, N\right) = \left(f_0^{(k-1)} + f_0^{(k)}\right)\cdot \pi N \tfrac{1}{f_s} \qquad (4)$$

siendo f_s la frecuencia de muestreo. Se propone la siguiente corrección para las fases:

$$\Delta\varphi_1^{(k)} = \Psi\left(f_0^{(k-1)}, f_0^{(k)}, N'\right) - \Psi\left(f_0^{(k-1)}, f_0^{(k)}, N\right) \qquad (5a)$$

$$\varphi_j'^{(k)} = \varphi_j^{(k)} + j\sum_{q=1}^{k} \Delta\varphi_1^{(q)} \quad j = 1\ldots J^{(k)} \quad \forall k \qquad (5b)$$

siendo N' la nueva longitud de trama y Ψ la definida en (4). La corrección no afecta al tracto vocal sino solamente al término lineal de fase, como se pretendía. No es necesario modificar los coeficientes LPC de la parte estocástica.

2.2.2. Modificación en pitch

Para modificar el pitch de la señal, se multiplican las frecuencias por el factor deseado y se recalcula el número de armónicos hasta 5KHz. Las amplitudes de los nuevos armónicos se obtienen mediante una simple interpolación lineal entre las log-amplitudes medidas en el análisis [9], manteniendo la energía original con un factor multiplicativo. Si se elimina el término lineal de fase, la respuesta en fase del tracto vocal en los nuevos

armónicos se puede obtener interpolando linealmente la parte real e imaginaria de las amplitudes complejas [9]. El término lineal se suprime del siguiente modo [10]:

$$\varphi_{VT\,j}^{(k)} = \varphi_j^{(k)} - j\alpha^{(k)} \quad (6a)$$

$$\alpha^{(k)} = \arg\min \sum_{j=0}^{J^{(k)}-1} \left| \sqrt{A_j^{(k)}} e^{i\left(\varphi_j^{(k)} - j\alpha^{(k)}\right)} - \sqrt{A_{j+1}^{(k)}} e^{i\left(\varphi_{j+1}^{(k)} - (j+1)\alpha^{(k)}\right)} \right|^2 \quad (6b)$$

tomando $A_0^{(k)}=A_1^{(k)}$ y $\varphi_0^{(k)}=0$. Una vez calculada la fase del tracto vocal en los nuevos armónicos, se repone el término lineal usando el mismo valor de α. También en este caso habría que actualizar el término lineal de fase, pues la longitud del período cambia mientras que N se mantiene constante. La corrección se realiza por medio de la fórmula (5b) con el incremento siguiente:

$$\Delta\varphi_1^{(k)} = \Psi\left(f_0'^{(k-1)}, f_0'^{(k)}, N\right) - \Psi\left(f_0^{(k-1)}, f_0^{(k)}, N\right) \quad (7)$$

Los coeficientes de la parte estocástica no se modifican. Tanto el factor de modificación de pitch como el de duración pueden ser variables con el tiempo, y se pueden efectuar los dos tipos de modificaciones de forma simultánea.

2.3. Concatenación de unidades

Se dispone de una base de datos analizada como se ha descrito. Se desea concatenar una serie de unidades siguiendo unas especificaciones concretas de duración, energía y contorno de pitch. A cada unidad se le aplican las transformaciones prosódicas pertinentes mediante la modificación de sus amplitudes, frecuencias, fases y ganancia de los filtros LPC. De nuevo el esquema pitch-asíncrono plantea el problema de la coherencia de fase entre las unidades. Para evitarlo, se debe imponer que la componente lineal de la fase sea continua en el punto de concatenación. Siguiendo este planteamiento, para cada dos unidades adyacentes A y B se calculan en los puntos limítrofes k_A y k_B las correspondientes $\alpha^{(k_A)}$ y $\alpha^{(k_B)}$ conforme a la ecuación (6b), y se ajustan las fases de la segunda unidad imponiendo que el incremento del término lineal de fase entre k_A y k_B sea el dado por la función Ψ definida en (4):

$$\varphi_j'^{(k)} = \varphi_j^{(k)} + j\cdot\left(-\alpha^{(k_B)} + \alpha^{(k_A)} + \Psi\left(f_0^{(k_A)}, f_0^{(k_B)}, N\right)\right) \quad (8)$$
$$j = 1\ldots J^{(k)} \quad k = k_B, k_B+1, \ldots$$

Por ultimo, se suavizan las amplitudes de los armónicos en torno al punto de concatenación de modo que no haya discontinuidades espectrales que den lugar a ruido.

3. MÉTODO DE CONVERSIÓN DE VOZ

Para que la transformación entre dos hablantes sea óptima, habría que tener en cuenta aspectos muy diversos, desde la curva de entonación de cada uno hasta los vocablos o expresiones que los caracterizan. Este trabajo se centra solamente en la conversión espectral, aunque se realiza también una transformación básica de pitch. A continuación se desarrollan cada una de las etapas que componen el bloque de conversión.

3.1. Normalización del pitch

La frecuencia fundamental se caracteriza por una distribución de tipo log-normal. A partir de los datos disponibles de cada hablante se extrae una estimación de la media y la varianza de su $logf_0$. Se realiza la siguiente normalización:

$$\log f_0^{(\text{converted})} = \mu^{(\text{target})} + \frac{\sigma^{(\text{target})}}{\sigma^{(\text{source})}}\left(\log f_0^{(\text{source})} - \mu^{(\text{source})}\right) \quad (9)$$

3.2. Conversión del tracto vocal

El tracto vocal puede ser modelado por medio de distintos tipos de parámetros: amplitudes normalizadas a una determinada frecuencia fundamental, coeficientes cepstrales discretos, coeficientes LSF, etc. Los coeficientes LSF son los más habituales en conversión de voz, debido a que modelan bien la estructura de formantes y a que el error de estimación en uno de estos coeficientes afecta a una pequeña parte del espectro. En el presente trabajo se tuvieron en cuenta otras ventajas: estos coeficientes codifican también una envolvente de fase mínima, y además en la parte estocástica se emplea el mismo tipo de codificación, lo que permite relacionar ambas componentes de la señal como se verá más adelante.

Para traducir las amplitudes a coeficientes LSF se calcula el filtro todo-polos óptimo mediante la técnica del modelado todo-polos discreto [11], que trata de minimizar la medida de distorsión de Itakura-Saito entre la magnitud del filtro y las amplitudes de los armónicos. El orden del filtro utilizado es 14 para una frecuencia de muestreo de 16KHz, pues órdenes superiores aportan información poco relevante a la vez que complican la conversión. Para la obtención de la función de transformación se ajusta un modelo de mezclas gaussianas de orden m=8 a un corpus de entrenamiento compuesto por vectores de la forma $v=[x^T\ y^T]^T$, donde los vectores columna x e y contienen p=14 parámetros LSF del hablante fuente y objetivo, respectivamente, correspondientes a los mismos fonemas [5]. Se usa el algoritmo EM para buscar los α_i, μ_i y Σ_i tales que sea máxima la verosimilitud del conjunto de vectores v del corpus de entrenamiento, de acuerdo a

$$p(v) = \sum_{i=1}^{m} \alpha_i N(v, \mu_i, \Sigma_i) \quad (10a)$$

$$N(v, \mu, \Sigma) = \frac{1}{(2\pi)^{p/2}} |\Sigma|^{-1/2} \exp\left\{-\tfrac{1}{2}(v-\mu)^T \Sigma^{-1}(v-\mu)\right\} \quad (10b)$$

La función de transformación se define como

$$F(x) = \sum_{i=1}^{m} p_i(x)\cdot\left[\mu_i^y + \Sigma_i^{yx}\left(\Sigma_i^{xx}\right)^{-1}\left(x-\mu_i^x\right)\right] \quad (11a)$$

$$p_i(x) = \frac{\alpha_i N\left(x, \mu_i^x, \Sigma_i^{xx}\right)}{\sum_{j=1}^{m} \alpha_j N\left(x, \mu_j^x, \Sigma_j^{xx}\right)} \quad (11b)$$

$$\mu_i = \begin{bmatrix} \mu_i^x \\ \mu_i^y \end{bmatrix} \quad \Sigma_i = \begin{bmatrix} \Sigma_i^{xx} & \Sigma_i^{xy} \\ \Sigma_i^{yx} & \Sigma_i^{yy} \end{bmatrix} \qquad (11c,d)$$

Para la obtención de la función óptima de transformación es necesario disponer de corpus de entrenamiento paralelos. En este trabajo, sin embargo, se han construido corpus paralelizados, generando con ayuda de un sintetizador las mismas frases que se disponen del locutor objetivo [1].

3.3. Estimación de la envolvente de fase

Para que la voz sintetizada sea de alta calidad, las variaciones temporales en la magnitud del tracto vocal deben ir acompañadas de sus correspondientes variaciones en fase, manteniendo a la vez la coherencia entre tramas. Para satisfacer estos dos objetivos, se calcula en primer lugar un término lineal de fase de forma recursiva:

$$\varphi_j^{(k)} = \varphi_j^{(k-1)} + j\Psi\left(f_0^{(k-1)}, f_0^{(k)}, N\right) \qquad (12)$$

Al comienzo de cada región sonora, las fases son inicializadas a cero. Este primer paso no provoca discontinuidades de fase importantes, pero al no haber variaciones en la fase del tracto vocal entre las diferentes tramas, puede ocasionarse un desagradable ruido metálico en la señal sintetizada. Por este motivo se añade un segundo término:

$$\varphi_j'^{(k)} = \varphi_j^{(k)} + \arg\left\{1/A\left(f_j^{(k)}\right)\right\} \qquad (13)$$

siendo $1/A(f)$ el filtro todo-polos convertido a partir de las amplitudes de los armónicos. Este término no representa realmente la envolvente de fase del tracto vocal, pero proporciona variaciones de fase coherentes con las amplitudes, y la calidad que se alcanza es muy aceptable.

3.4. Predicción de la parte estocástica

Es sabido que la relevancia de la parte estocástica en conversión de voz es escasa en relación con la parte armónica de la señal [4]. Sin embargo, se debe tener en cuenta que la separación entre ambas componentes no es perfecta, siendo muy difícil separarlas por completo en tramas sonoras. Además, por encima de la frecuencia de corte de 5KHz se considera que no hay armónicos, cosa que no es realista a pesar de la alta calidad lograda. Por estos y otros motivos, se observa una gran correlación entre la forma del tracto vocal y la de la envolvente LPC que caracteriza la parte estocástica. Aprovechando que sus parametrizaciones son similares y que se dispone de un modelo de mezclas gaussianas que divide en clases solapadas el espacio acústico del tracto vocal, se pueden entrenar vectores η_i y matrices Γ_i tales que se optimice a lo largo del corpus de entrenamiento la relación siguiente:

$$y_{est} = \sum_{i=1}^{m} p_i(y) \cdot \left[\eta_i + \Gamma_i\left(\Sigma_i^{yy}\right)^{-1}\left(y - \mu_i^y\right)\right] \qquad (14)$$

siendo y_{est} el vector LSF estocástico. A la hora de convertir, para obtener la parte estocástica se empleará la fórmula (14) con el vector $F(x)$ calculado previamente en (11a) en lugar de y.

En las zonas sordas no se ha realizado transformación alguna, pues su importancia perceptual es muy relativa y los intentos de conversión efectuados no mejoran el resultado de manera visible, comprometiendo en algunos casos la calidad.

4. EXPERIMENTOS

Para evaluar el sistema de síntesis-conversión de voz, se han usado grabaciones de alta calidad de más de 150 frases, de duración en torno a 5 segundos, pronunciadas por 4 locutores profesionales: 2 hombres y 2 mujeres. El 80% del material grabado se ha utilizado para la elaboración del corpus de entrenamiento, y el resto para la evaluación propiamente. Se han usado otros dos métodos de conversión como referencia, cuyas características son las siguientes [1]:

- M1: se trata de un sintetizador basado en TD-PSOLA creado a partir de las grabaciones del locutor objetivo, de modo que no convierte voces, pero sirve como referencia.
- M2: se trata de un sistema basado en LP-PSOLA. Utiliza como datos de entrada no sólo las muestras de audio sino también información privilegiada de tipo fonético: vocal/ consonante, sordo/sonoro, etc. Usa árboles de decisión para clasificar los sonidos y aplicar distintas funciones de conversión a cada clase. Además, aumenta la resolución de la conversión mediante un proceso de selección+suavizado de tramas residuales de la codificación LPC. El sistema se ha entrenado con frases paralelas entre sí.

A diferencia del método M2, para el entrenamiento de la función de conversión del sistema propuesto se han creado corpus no paralelos con ayuda de un sintetizador, de modo que se evalúa la capacidad del sistema para actuar en situaciones en las que no se dispone de grabaciones de las mismas frases para todos los locutores.

Se han realizado pruebas de conversión en todas las direcciones posibles entre los locutores disponibles, a partir de las frases de test. Los resultados finales se han obtenido promediando los resultados de las conversiones individuales. Para obtener estos resultados, varios oyentes escucharon parejas de fragmentos de audio correspondientes a la voz convertida y la voz objetivo, puntuando el parecido entre 1 (diferentes) y 5 (iguales). Se puntuó también la calidad de la voz convertida entre 1 (mala) y 5 (excelente). Los resultados son los expuestos en la tabla 1:

	Parecido	Calidad
M1	3.35	3.20
M2	3.47	2.25
Propuesto	3.15	2.38

Tabla 1. *Resultados de la evaluación*

Se pueden realizar diversas observaciones. En primer lugar, cabría esperar una mayor puntuación en el parecido obtenido con M1, puesto que usa directamente la voz objetivo. Se puede concluir que el decremento en la calidad de la voz sintetizada debida a los pocos datos usados en síntesis, afecta de manera directa a la percepción del parecido por parte de los oyentes. Aclarado este punto, se puede comprobar que el sistema propuesto se encuentra prácticamente al nivel de M1 y M2, aun cuando estos utilizan información privilegiada, así como corpus paralelos de entrenamiento en el caso de M2. Los resultados obtenidos en cuanto a parecido de la voz son coherentes con el estado del arte.

La calidad obtenida es más baja de lo esperado. Los oyentes parecen preferir la voz de un sintetizador construido con pocos minutos de audio que la voz manipulada y convertida. No obstante, se obtienen mejores resultados en el sistema propuesto que en M2, lo cual parece ser debido sobre todo a haber evitado problemas derivados de la coherencia de fase. Aunque se esperaba una mayor mejora, hay que señalar que el proceso de alineado del corpus no paralelo puede estar influyendo negativamente en la calidad, mientras M2 está libre de estos fallos. En cualquier caso, se puede concluir que la manipulación de la señal ocasiona una pérdida en la naturalidad de la voz que hace que los oyentes la consideren de peor calidad.

Los resultados invitan a plantearse la búsqueda de métodos de manipulación que degraden en menor medida la calidad de la señal, pues el oyente prefiere una alta calidad que un perfecto parecido. Por lo tanto, en futuros trabajos se trabajará en aspectos mejorables como por ejemplo la utilización de envolventes de fase más realistas que la aproximación de fase mínima, así como de funciones de transformación de tracto vocal que provoquen un menor suavizado espectral y ensanchamiento de los formantes.

5. CONCLUSIONES

En este artículo se ha desarrollado un sistema de conversión de voz sobre un sintetizador basado en el modelado armónico/estocástico. Se ha logrado un grado satisfactorio de parecido entre voces convertidas y objetivos, al nivel de otros sistemas que usan información privilegiada. Sin embargo, aunque se superan problemas derivados de la coherencia de fase como se pretendía, no se logra evitar la degradación de la señal debida a la manipulación espectral. Es necesario seguir perfeccionando el método de conversión para que se comprometa menos la calidad de la señal.

6. AGRADECIMIENTOS

Este trabajo se ha llevado parcialmente a cabo con fondos procedentes del proyecto TC-STAR, *Technology and Corpora for Speech-to-Speech Translation* (FP6-506738). Los autores agradecen también a Javier Pérez y Ferran Diego su ayuda en la elaboración de los corpus de entrenamiento.

7. BIBLIOGRAFÍA

[1] H.Duxans, D.Erro, J.Perez, F.Diego, A.Bonafonte, A.Moreno, "Voice Conversion of Non-Aligned Data using Unit Selection", TC-STAR Workshop on Speech-to-Speech Translation, 2006.

[2] D.Sündermann, H.Höge, A.Bonafonte, H.Ney, J.Hirschberg, "TC-Star: Cross Language Voice Conversion Revisited", TC-STAR Workshop on Speech-to-Speech Translation, 2006.

[3] Y.Stylianou, O.Cappé, E.Moulines, "Continuous Probabilistic Transform for Voice Conversion", IEEE Trans. on Speech and Audio Proc., 1998.

[4] H.Ye, S.Young, "Quality-enhanced Voice Morphing using Maximum Likelihood Transformations", IEEE Trans. On Audio, Speech and Lang. Proc., 2006.

[5] A.Kain, M..Macon, "Spectral Voice Conversion for Text-to-Speech Synthesis", ICASSP'98, 1998.

[6] D.Erro, A.Moreno, "A Pitch-Asynchronous Simple Method for Speech Synthesis by Diphone Concatenation using the Deterministic plus Stochastic Model", Proc. SPECOM, 2005.

[7] Ph.Depalle, T.Hélie, "Extraction of Spectral Peak Parameters using a STFT Modeling and no Sidelobe Windows", Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 1997.

[8] R.J.McAulay, T.F.Quatieri, "Speech Analysis/ Synthesis based on a Sinusoidal Representation", IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. 34 (4), pp. 744-754, 1986.

[9] E.R.Banga, C.García-Mateo, X.Fernández-Salgado, "Concatenative Text-to-Speech Synthesis based on Sinusoidal Modeling", Improvements in Speech Synthesis, John Wiley and Sons, 2001.

[10] D.Chazan, R.Hoory, Z.Kons, D.Silberstein, A.Sorin, "Reducing the Footprint of the IBM Trainable Speech Synthesis System", Proc. ICSLP, 2002.

[11] A.El-Jaroudi, J.Makhoul, "Discrete All-Pole Modeling", IEEE Trans. on Signal Proc., 1991.

SÍNTESIS DE HABLA EMOCIONAL POR SELECCIÓN DE UNIDADES

Ignasi Esquerra

Centre de Recerca TALP
Departamento de Teoría de la Señal y Comunicaciones
Universitat Politècnica de Catalunya

RESUMEN

Una de las áreas de investigación en síntesis de habla más en boga actualmente es la síntesis de emociones y estilos de habla diferenciados del estilo por defecto habitual de los sistemas TTS. Principalmente existen dos maneras de atacar este problema: modelando adecuadamente la prosodia (entonación, duración e intensidad), o utilizando sistemas por concatenación con unidades grabadas con otros estilos. En el trabajo que se presenta a continuación se investiga la importancia relativa de la prosodia y de las unidades para sintetizar emociones y estilos con un sistema TTS por selección de unidades. Los experimentos realizados muestran diferencias entre las emociones y estilos estudiados, siendo unos más dependientes de la prosodia y menos de las características de la voz de las unidades.

1. LA SÍNTESIS DE HABLA EXPRESIVA

Gracias a la técnica de síntesis por selección de unidades a partir de grandes bases de datos se ha conseguido mejorar de forma substancial la calidad de los sistemas de conversión texto-habla. Sin embargo, en la mayoría de casos las voces sintéticas suenan todavía en cierto modo aburridas, con un mismo estilo de habla estándar, en especial cuando se sintetizan textos relativamente largos. Para mejorar la naturalidad de la voz sintética se debería considerar la variabilidad existente en la comunicación oral entre humanos. Así por ejemplo, hablamos más rápido o más lentamente, más fuerte o más flojo, según las condiciones del entorno, de nuestros interlocutores o incluso de nuestro estado de ánimo. A veces estos cambios en el estilo del habla no se reflejan solamente modificando la velocidad o intensidad del habla, sino también variando la entonación, el grado de articulación de los sonidos o las características tímbricas de la voz.

La síntesis de habla emocional cabría englobarla dentro del marco más general de la síntesis de voz con estilos de habla no neutros. Actualmente hay quien utiliza el término síntesis de habla ”expresiva” para significar que el habla humana es muy rica en matices, y que las emociones solamente son una parte más de la gran variabilidad del habla humana [1]. Históricamente el estudio de las emociones se ha realizado desde ámbitos muy diversos, como la psicología o la fonética acústica, y no ha sido hasta hace poco que también se ha abordado desde el punto de vista de las tecnologías del habla.

Desde el inicio del desarrollo de sintetizadores de voz ha habido diferentes intentos de dotarlos de la capacidad para generar voces con emoción [2,3]. Muchos de los primeros intentos de sintetizar habla emocionada se realizaron sobre sistemas basados en reglas, y en particular con sistemas de síntesis por formantes [4,5]. Éstos tienen la ventaja de poder actuar directamente sobre muchos de los parámetros que definen la prosodia y los sonidos en el modelo de síntesis, si bien la inteligibilidad y naturalidad es comparativamente peor que la de los sistemas más modernos basados en concatenación de unidades.

Actualmente la mayoría de sistemas texto-habla utilizan la concatenación de segmentos de voz extraídos de grandes bases de datos, seleccionando el segmento más apropiado en cada ocasión. Este hecho provoca que la voz generada mantiene las características propias del locutor y en cierta medida del estilo en que fueron grabadas. Para poder generar otros tipos de habla, es necesario disponer de corpus más extensos que recojan las variantes de habla deseadas [6].

En el caso del español hay pocos trabajos documentados de análisis o síntesis de emociones, entre los cuales cabe destacar los de A.Rodríguez [7] y J.M.Montero [8]. A destacar también el inicio de trabajos para el euskera de E.Navas [9], y para el caso del catalán de I.Iriondo [10].

En el presente trabajo se presenta un experimento de síntesis de habla con emociones realizadas con un sistema por selección y concatenación de unidades. Se ha utilizado una base de datos grabada con diferentes emociones y estilos para generar distintas bases de unidades, con las cuales se han sintetizado unas frases con la prosodia por defecto del sistema o con la prosodia analizada de las frases reales. Esto nos proporciona información sobre como afecta la prosodia o las características acústicas de las voces en la expresión de las emociones sintéticas.

En las secciones que siguen se introduce en primer lugar el sistema TTS y la base de datos utilizada. A continuación se presenta el método de generación de las bases de unidades, para seguidamente describir los experimentos de síntesis y sus resultados. Se finaliza con una discusión de las conclusiones del trabajo.

2. EL SISTEMA DE CONVERSIÓN TEXTO-HABLA

Una de las áreas principales de trabajo en el Centro de Tecnologías y Aplicaciones del Lenguaje y del Habla (TALP) es la conversión texto-habla, es decir la generación de voz sintética a partir de texto. Estos sistemas, que se conocen habitualmente con las siglas TTS (del inglés, *text-to-speech*), analizan y procesan los textos de entrada, los convierten en su correspondiente representación en sonidos, determinan cómo se deben pronunciar, y finalmente generan acústicamente dichos sonidos. El sistema desarrollado en la UPC, como la mayoría de sintetizadores actuales, genera la voz mediante la técnica de selección de unidades [11].

El primer módulo del sistema TTS consiste en el procesado lingüístico. Su misión es capturar el texto que se quiere leer y prepararlo para los módulos siguientes. En particular, se debe identificar las frases y oraciones, y convertir los dígitos, abreviaturas y otros signos en sus correspondientes transcripciones ortográficas. A continuación, mediante el uso de diccionarios o de reglas de trascripción el texto se convierte a fonemas. El siguiente módulo es el de generación de la prosodia, que proporciona los valores de frecuencia fundamental, duración e intensidad de los sonidos.

El módulo de generación de señal consiste en el algoritmo de selección de las unidades, y en las técnicas de procesado de señal para unirlas y modificarlas convenientemente. La síntesis por selección de unidades permite escoger de entre todos los segmentos de la base de datos la unidad que mejor se adecue a lo que determinan los módulos anteriores, tanto los valores de prosodia, factores fonéticos o fonológicos, o las características acústicas de los segmentos previamente seleccionados.

Todos estos módulos trabajan sobre una estructura de datos multicapa, donde las informaciones se guardan en diferentes niveles (por ejemplo, palabras, fonemas, unidades de síntesis) y cuyos valores se relacionan entre sí y con los demás niveles mediante vínculos del tipo contenido-continente.

3. LA BASE DE DATOS EMOCIONAL

Las señales de voz utilizadas en este trabajo son una parte de la base de datos INTERFACE en español [12]. Dos locutores, uno masculino y otro femenino, grabaron un corpus de frases, párrafos y palabras simulando las seis emociones estándar de MPEG4 además del estilo neutro. También se grabó parte del corpus en otros cuatro estilos de habla (tabla 1). Para las emociones principales se grabaron dos sesiones del corpus completo en días diferentes lo que representa unos 12 minutos de voz por locutor, sesión y emoción. Para los estilos de habla se dispone de unos 10 minutos correspondientes a una única sesión de grabación.

Las señales fueron grabadas a 32kHz, aunque después se redujo la frecuencia de muestreo a 16kHz para generar las bases de unidades para el conversor texto-habla. De forma simultanea se adquirió la señal de un laringógrafo, que se utilizó para la extracción de los instantes de periodicidad glotal necesarios para determinar la frecuencia fundamental.

Enfado (A)	Alto (H)
Asco (D)	Bajo (L)
Miedo (F)	Lento (W)
Alegría (J)	Rápido (Z)
Sorpresa (S)	
Tristeza (T)	
Neutro (N)	

Tabla 1. *Emociones y estilos de habla de la base.*

Con la ayuda del sistema de reconocimiento del habla de la UPC basado en modelos de Markov se procedió a segmentar de forma automática los ficheros de voz [13]. Debido a la existencia de importantes diferencias en la duración de los fonemas y la presencia de muchas pausas en las locuciones, se realizó una segmentación en dos fases: una primera para detectar la posición de las pausas reales y obtener una segmentación aproximada, y una segunda fase para ajustar mejor las fronteras entre fonemas. Se usaron modelos entrenados de forma conjunta para todos los estilos. No se realizó una verificación manual de los resultados de la segmentación, salvo en algunos casos muy concretos detectados al hacer posteriormente la síntesis.

Aunque la base de datos en su conjunto es relativamente grande, al repetirse las frases del corpus para cada emoción y estilo hace que el contenido en cuanto a número de unidades distintas sea pequeño. En la tabla 2, se compara el contenido de la base de datos emocional (IESSDB) con otra utilizada por el sistema TTS de mayor tamaño (ESMASE).

	IESSDB	ESMASE
palabras	1819	4411
fonemas	8681	22014
unidades	17362	44028
difonemas	494	539

Tabla 2. *Contenido de las bases de datos emocional y de síntesis.*

Se puede observar que la base IESSDB no está diseñada y suficientemente dimensionada como para cubrir todos los difonemas necesarios para la síntesis, aunque cabe decir que las unidades que faltan corresponden a combinaciones muy poco habituales entre fonemas y que, en caso de requerirse tal unidad, el sistema TTS las sustituye por unidades parecidas desde el punto de vista fonológico.

4. GENERACIÓN DE LAS BASES DE UNIDADES

Una de las ventajas, y a la vez inconveniente, de la síntesis basada en concatenación de unidades es que la voz sintética conserva las características del locutor o locutora que realizó la grabación de la base de datos oral. Si bien existen técnicas de procesado de señal (como por ejemplo TD-PSOLA) que permiten transformar las características temporales de las unidades que se concatenan, en general éstas no son suficientes para las grandes modificaciones prosódicas que se requieren para convertir una voz neutra en una voz emocionada. Por ejemplo, la frecuencia fundamental (F0) media de emociones como la alegría o la tristeza pueden estar bastante alejadas del valor medio normal del estilo neutro. En el caso de querer modificar además las características tímbricas de las unidades, sería necesario recurrir a otras técnicas de procesado de señal.

En un trabajo anterior [14] se vio cómo la frecuencia fundamental y las duraciones son factores importantes para reconocer las emociones. En particular se mostró cómo era posible transformar una frase prosódicamente neutra en una con emoción simplemente trasplantando su prosodia. Por otra parte, las frases con emoción a las cuales se les imponía un patrón de prosodia neutro continuaban siendo reconocibles por la calidad de la voz, aunque en menor medida.

4.1. Bases de unidades independientes por emoción

Mientras que en los sistemas basados en síntesis por formantes era relativamente fácil crear voces diferenciadas aplicando reglas y tablas de valores, en los sistemas basados en selección de unidades se necesita grabar bases diferentes para cada uno de los estilos de habla deseados [6].

Aprovechando que el sistema permite seleccionar las voces del sintetizador mediante comandos XML insertados en el texto de entrada, se pueden generar varias voces sintéticas del mismo locutor con diferentes estilos y conmutar entre ellas de la misma manera como se cambiaría de locutor (tabla 3).

<*SPEAKER="marta"*> Mientras encendía un cigarrillo <*SPEAKER="marta_surprise"*> me sorprendí a mí misma <*/SPEAKER*> pensando en cómo puede llegar cierta gente <*SPEAKER="marta_disgust"*> a ciertos puestos de responsabilidad <*/SPEAKER*> poseyendo una mentalidad a nivel de subcultura. <*/SPEAKER*>

Tabla 3. *Texto con selección de locutor/emoción.*

Partiendo de los ficheros de voz de la base IESSDB, se procedió a crear una base de unidades para cada emoción y locutor, y se incorporaron al sistema TTS como voces independientes. Es decir, cada locutor pasaba a tener hasta 11 voces diferentes seleccionables como se ha visto anteriormente. Al tener menos ficheros disponibles (entre 10 y 24 minutos), las bases de unidades resultantes son pequeñas y, aunque están presentes todas las unidades comunes del corpus, no existen muchas repeticiones de las mismas para conseguir una calidad óptima.

5. SÍNTESIS Y RESULTADOS

Se realizaron unas primeras pruebas de síntesis con las bases de unidades independientes para cada emoción. El sistema TTS está desarrollado con único módulo de prosodia que proporciona un estilo de habla neutro, por lo que se asigna una prosodia igual para todas las emociones. Aun así, la voz resultante suena bastante con la emoción o estilo de habla de la base de unidades de la cual proviene.

Por otra parte, los valores de duración y F0 de las unidades seleccionadas por el sintetizador pueden estar muy alejados de los valores "objetivo" determinados por el módulo de generación de la prosodia, en especial para aquellas emociones con grandes variaciones de estos valores como son la alegría o la tristeza. Las técnicas para adaptar la prosodia no siempre consiguen alcanzar dichos valores puesto que se establecen límites a los factores de modificación, y en consecuencia las señales sintetizadas conservan en buena medida los valores de F0 y duración de las frases con emoción. Por esta razón es posible que las frases sintetizadas no lleguen a conseguir los valores de la prosodia neutra determinada por el sistema.

Es curioso observar como, aunque las frases que se sintetizan formen parte del propio corpus, un 8% de las unidades seleccionadas provienen de otras frases. Esto es debido a que en estos casos predomina el factor de los valores del coste "objetivo" por encima del coste de "concatenación" en el proceso de selección de las unidades.

5.1. Síntesis con prosodia natural forzada

Para comprobar el efecto de la prosodia por defecto del sistema, se procedió a sintetizar unas frases con su prosodia real. Estas frases de prueba se eliminaron en la fase de generación de las bases de unidades. Al forzar la duración y frecuencia fundamental de las unidades a los valores naturales, se consiguen frases muy parecidas a las locuciones originales, tanto en prosodia como en calidad de la voz.

Se procedió igualmente a sintetizar dichas frases con el sistema con voces estándar y forzando su prosodia a los valores analizados de las frases reales con emoción. Comparándolas con las anteriores, estas suenan mucho más naturales aunque es difícil percibir las emociones por la prosodia. Esto se contradice en parte con los resultados del trabajo de investigación previo de resíntesis con trasplante de prosodias [14], donde se mostraba que era posible reconocer las emociones en frases neutras a las cuales se les había forzado una prosodia con emoción.

Una posible causa de esta diferencia puede ser el hecho que al trasplantar la prosodia se preservaban todas las pequeñas variaciones de entonación (microprosodia) mientras que al hacer la síntesis por concatenación el patrón de prosodia se reduce a un único valor por fonema, y esto puede ser insuficiente para caracterizar la evolución de la entonación en las emociones.

5.2. Evaluación

Con el fin de conocer mejor la importancia relativa de disponer de una buena prosodia o de unas buenas unidades con emoción, se procedió a realizar un test perceptivo. En él se evaluaba la calidad y el grado de reconocimiento de las emociones y estilos en las señales sintetizadas mediante los tres métodos de síntesis descritos hasta ahora. Estos se denominaran de la siguiente manera:

- Método ES: bases de unidades independientes por emoción, y prosodia por defecto del sistema.
- Método EN: bases de unidades independientes por emoción, y prosodia natural.
- Método DN: base de unidades por defecto del sistema, y prosodia natural.

El test se realizó a través de Internet con 25 voluntarios, de manera no supervisada. Consistía en una introducción explicativa y tres formularios. En el primero se presentaban 12 señales de audio y se pedía identificar la emoción entre un conjunto de 7 posibilidades, e indicar la calidad subjetiva según una escala tipo MOS de 5 valores. La segunda parte del test era similar a la anterior para los estilos de habla, y se presentaban 6 señales. En la tercera parte, se comparaban los métodos de dos en dos mediante un test tipo AB; se presentaban dos ficheros de audio junto con una indicación de la emoción o estilo al que correspondían, y se tenía que marcar cuál de los dos ficheros era el que mejor expresaba dicha emoción o bien si se consideraba que eran igualmente válidos.

Para las emociones, las matrices de confusión entre estímulo y respuesta muestran como para los métodos ES y EN no hay una preferencia clara hacia la emoción correcta respecto a las demás, aunque siempre es la obtiene la máxima valoración. Por el contrario, en el método DN las respuestas son mayoritariamente en el sentido de marcar la opción "neutral". Es decir, a pesar de haberse sintetizado con prosodia natural, ésta no es suficiente para expresar la emoción de forma convincente. Las tasas de reconocimiento de las emociones reflejan esta situación (tabla 4a) y corroboran las primeras impresiones comentadas anteriormente al principio de esta sección. Es curioso constatar que en algunos casos el método EN con prosodia natural da unos resultados peores que el método ES con la prosodia por defecto neutral.

En cuanto a los estilos de habla se observa que para el habla rápida o lenta se obtienen muy buenos resultados (tabla 4b) debido principalmente a que todas las unidades disponibles en las bases tienen una duración menor o mayor respectivamente al habla normal. No así, para los dos otros estilos los resultados son bastante malos. En particular se tiende a confundir el estilo "alto" por "rápido" y el "bajo" por "lento". Aquí cabe comentar que las etiquetas de estos dos estilos pudieron llevar a confusión entre los evaluadores puesto que no se especificó en las instrucciones si correspondían a altitud en entonación o en intensidad. De forma similar a las emociones, en el método DN se continua identificando mayoritariamente todas las señales como neutras.

	ES	EN	DN
A	68,8%	61,1%	0,0%
D	40,0%	29,4%	14,3%
F	35,3%	25,0%	0,0%
J	44,4%	36,8%	7,7%
S	33,3%	52,9%	0,0%
T	68,8%	86,7%	10,5%
N	83,3%	50,0%	78,6%

(a)

	ES	EN	DN
H	35,7%	46,2%	14,3%
L	20,0%	41,7%	25,0%
Z	100,0%	92,3%	53,8%
W	90,0%	93,3%	0,0%
N	33,3%	71,4%	81,8%

(b)

Tabla 4. *Resultados del test de reconocimiento de emociones (a) y estilos (b)*

Por lo que respecta a la valoración de la calidad, el método que recoge el valor MOS más alto es el DN, sin duda porque es el que utiliza una base de unidades más grande. En general, con el método EN se obtienen valores más uniformes para el conjunto de emociones, mientras que para el método ES se da un valor hay algún caso más dispares.

Finalmente, en la comparación de los métodos, hay una preferencia clara de los métodos con unidades con emoción (EN y ES) respecto al método por defecto (DN), excepto para el caso de la emoción "neutra" en que se prefiere el método DN porque además de llevar prosodia natural se ha sintetizado con una base de unidades mayor.

Respecto a la relación "prosodia sintética" versus "prosodia natural", se presuponía que el método EN daría mejores resultados que el método ES. Esto de verifica para la mayoría de emociones y estilos, con excepción de "asco" (D), "miedo" (F), "rápido" (Z) y

"lento" (W) donde la respuesta mayoritaria fue la indiferencia entre los dos métodos. Se debería investigar más a fondo si estas dos emociones tienen una caracterización basada en el rol de la duración.

Mediante un sistema de puntuación que asigna 3 puntos al método preferido, y un punto en caso de no preferencia por ninguno de los dos, se obtiene la siguiente gráfica (figura 1).

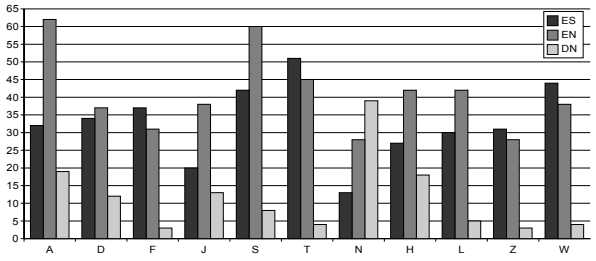


Figura 1. *Resultados de la comparación de métodos*

Para las emociones "enfado" (A), "alegría" (J), "sorpresa" (S), "alto" (H) y "bajo" (L) el método preferido es el de unidades con prosodia natural. Para los demás, no se aprecia mucha diferencia entre los métodos con prosodia natural o sintética, cosa que nos lleva a pensar que la expresión de las emociones se realiza principalmente por la calidad de la voz en lugar de por la prosodia.

6. CONCLUSIONES

Con los experimentos realizados se ha visto como es posible sintetizar emociones con bases de unidades provenientes de frases grabadas con un diferentes estilos de habla. Las tasas de reconocimiento todavía no son lo suficientemente altas como sería de desear, seguramente debido a que las bases utilizadas no son lo bastante grandes y, por otra parte, los modelos de prosodia todavía están incorporados en el sistema TTS.

Actualmente se está trabajando en la incorporación de la característica de emoción/estilo dentro de las etiquetas de selección de unidades. Esto permitirá generar una única base y escoger las unidades no sólo en función de sus características acústicas, fonológicas o prosódicas, sino también en base a parámetros lingüísticos más abstractos como es el estilo de habla. Así mismo se trabaja en el desarrollo de módulos prosódicos que modelen las diferentes emociones.

7. AGRADECIMIENTOS

El autor quiere expresar su gratitud a las personas del departamento y externas que han participado en las pruebas de evaluación. Este trabajo ha sido parcialmente financiado por la Unión Europea bajo el proyecto TC-STAR (http://www.tc-star.org/).

8. REFERENCIAS

[1] N. Campbell, *"Towards synthesising expressive speech; designing and collecting expressive speech data"*. Proc. 8th European Conference on Speech Communication and Technology (EUROSPEECH), 2003

[2] I.R. Murria, J. L. Arnott, *"Toward the simulation of emotion in synthetic speech: a review of the literature on human vocal emotion"*. *JASA*, vol.2, pp. 1097-1108, 1993

[3] M. Schröder, *"Emotional Speech Synthesis: a Review"*, Proc. 7th European Conference on Speech Communication and Technology (EUROSPEECH), pp. 561-564, 2001

[4] J.E. Cahn, *"Generating expression in synthesized speech"*, Master's thesis, Massachusets Institute of Technology, 1990

[5] I. R. Murray, J. L. Arnott, *"Implementation and testing of a system for producing emotion-by-rule in synthetic speech"*, *Speech Communication*, vol. 16, pp. 369-390, 1995

[6] A. W. Black, *"Unit Selection and Emotional Speech"*, Proc. of 8th European Conference on Speech Communication and Technology (EUROSPEECH), 2003

[7] A. Rodríguez-Bravo, P. Lázaro, N. Montoya, J. M. Blanco, D. Bernadas, J. M. Oliver, y L. Longhi, *"Modelización acústica de la expresión emocional en el español"*. *Procesamiento del Lenguaje Natural*, Revista SEPLN, pp. 159-166, 1999

[8] J.M. Montero, *"Estrategias para la mejora de la naturalidad y la incorporación de variedad emocional a la conversión texto a voz en castellano"*, Tesis doctoral, Universidad Politécnica de Madrid, 2003

[9] E. Navas, I. Hernáez, I. Luengo, J. Sánchez, *"Análisis Acústico de una Base de Datos de Habla Emocional"*, III Jornadas en Tecnologías del Habla (Valencia), pp. 233-236, nov. 2004

[10] I. Iriondo, F. Alías, J. Melenchón, M.A. Llorca, *"Modeling and Synthesizing Emotional Speech for Catalan Text-to-Speech Synthesis"*, Proc. Workshop on Affective Dialogue Systems, (Kloster Irsee, Germany), pp. 197-208, 2004

[11] A. Bonafonte, P.D. Agüero, J. Adell, J. Pérez, A. Moreno, *"OGMIOS: The UPC Text-to-Speech Synthesis System for Spoken Translation"*, Proc. of TC-STAR Workshop on Speech-to-Speech Translation (Barcelona), pp.199-204, Junio 2006

[12] V. Hozjan, Z. Kacic, A. Moreno, A. Bonafonte, A. Nogueiras, *"INTERFACE Databases: Design and collection of a multilingual emotional speech database"*, Proc. of the 3rd International Conference on Language Resources and Evaluation (LREC), pp. 2024-2028, 2002

[13] J. Adell, A. Bonafonte, *"Towards phone segmentation for concatenative speech synthesis"*. Proc. of 5th ISCA Speech Synthesis Workshop (Pittsburgh), 2004

[14] I. Esquerra, A. Bonafonte, *"Habla emocional mediante métodos de re-síntesis y selección de unidades"*, Actas del XIX Simposium URSI (Barcelona), 2004

A METHOD FOR AVOIDING F0 DISCONTINUITIES IN A CONCATENATIVE INTONATION MODEL

Francisco Campillo[*], *Taniya Mishra*[†], *Jan van Santen*[†]
Eduardo R. Banga[*]

[*] Signal Theory Group
Dpto. Teoría de la Señal y Comunicaciones
University of Vigo, SPAIN
[†] Center for Spoken Language Understanding
OGI School of Science & Engineering
20000 NW Walker Road, Beaverton, OR 97006, USA.
{campillo, erbanga}@gts.tsc.uvigo.es, {mishra, vansanten}@cslu.ogi.edu

ABSTRACT

This paper presents a novel approach for avoiding f0 discontinuities in the framework of a unit selection model of intonation with the accent group contour as the basic unit for concatenation. Taking as input the resulting contour of the model, it is modified on an accent group basis according to the typical decomposition into phrase and accent curves of the superpositional models. Results are presented that show a good performance of the proposed method, what implies that it can be a good approach for the treatment of the discontinuities, the main drawback of the concatenative models.

1. INTRODUCTION

Intonation modeling is widely acknowledged as one of the main factors regarding to the final quality of a TTS (Text-to-speech) system, independently of the underlying technology for generating the synthetic speech. This way, most current speech synthesizers produce a highly intelligible speech, but their naturalness is still far from being the desired one. In the literature we can find very different approaches to the problem of intonation, from the simple implicit modeling of [1], where the own intonation of the acoustic units resulting from unit selection is accepted, to other more complex methods. Particularly, in this paper we will focus on two classes: the superpositional and the concatenative models.

In the superpositional models ([2], [3]) the pitch curve is decomposed into several contributions corresponding to phonological entities with different temporal scope. The type of component curves, as well as the underlying assumptions on their possible shapes, make the distinction among the different superpositional models. For example, the Fujisaki model ([2]) considers three components: phrase curves, accent curves and a baseline constant value. However, the Linear Alignment Model ([3]) considers phrase curves, accent curves, and perturbation or residual curves. Independently of the set of curves, it is necessary to find an automatic method for decomposing the pitch contour to obtain every component, as well as a way for combining all this information and generating a target synthetic contour.

The concatenative models ([4], [5], [6]) do not make any assumption on the nature of the intonation, and try to generate the pitch contour by concatenating natural contours extracted from similar contexts to those where they are applied, in a process identical to the acoustic unit selection. These models are very simple and the task of analyzing the natural pitch contour is reduced to chunking it into consecutive sections according to the temporal scope of the phonological entity the basic unit is tied to. However, this models have an important flaw. If the available intonation corpus is not large enough, or the unit selection algorithm is not properly designed, there can be discontinuities in the concatenation points, what severely degrades the performance of the model. In this paper we will address this problem, proposing a novel method that employs the different components assumed by the superpositional models for eliminating the remaining discontinuities.

The outline of this paper is as follows. Sections 2 and 3 describe the main characteristics of the superpositional and concatenative models, respectively. Then, section 4 presents a comparison of both models, emphasizing their common points and their differences. Section 5 shows a method for modifying the pitch contour of an accent group in order to avoid the possible f0 discontinuities in the points of concatenation with the surrounding accent groups. Sections 6 and 7 describe a perceptual test conducted to check the performance of the proposed method, and a discussion of the results. Finally, section 8 is dedicated to the conclusions and some suggestions for future research.

2. SUPERPOSITIONAL MODELS

The superpositional models consider the pitch curve as the contribution of several component curves related to different phonological domains. For example, in the Fujisaki model ([2]), the most known superpositional model, the pitch curve is the result of the addition in the logarithmic domain of phrase curves (tied to the phonological phrase), accent curves (tied to the accented syllables) and a constant minimum level dependent on the speaker. Another example, the General Superpositional Model of Intonation ([3]), assumes the pitch curve to be the result of the addition in the log domain of three different components: the phrase and accent curves, and a new one, the residual or perturbation curve, related to the phonemes, which permits a more natural approximation of the pitch curve on taking into account the own microprosody of the segments. With respect to the previous model, where very hard constraints are held into the possible shape of the curves, in the General Superpositional Model these shapes are not specified. As a result, the analysis and extraction of the different components is a very hard task,

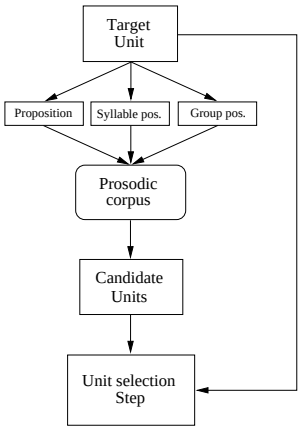


Figura 1. *Clustering of accent groups in the intonation unit selection*

although there are several promising approaches ([7], [8]).

3. CONCATENATIVE MODELS

Based on the same principles as the acoustic unit concatenation systems, these methods try to model the pitch contour as the result of the concatenation of natural contours. The underlying assumption is that an intonation unit applied in a very similar context to its original one should be almost indistinguishable from a natural contour. There are several approaches ([5], [6], [4]) with differences in the basic units for concatenation and the features modeling the context, but in this paper we will focus solely on the first one.

In [5] the process for generating the global pitch contour is formally equivalent to the acoustic unit selection of [9], which means that it is necessary to decide a basic unit for concatenation, a way of choosing a subset of candidate units for every target unit, and design a target and a concatenation cost functions for unit selection. In this case the basic intonational unit is the accent group, defined as a sequence of non accented words ending in an accented word. The subset of candidate groups is decided according to previous knowledge of the Spanish intonation ([10]): the accent groups are clustered into 48 classes as a function of three features: the position of the accent group within the phonic group (initial, final, medial and initial-final), the type of proposition (declarative, interrogative, exclamatory and ellipsis) and the position of the accented syllable within the accent group (ultimate, penultimate and antepenultimate). Figure 1 shows a diagram of this first hard constraint that decides the subset of candidate units for a unit selection step. After that, the target cost function considers other features, such as the number of syllables, the duration of the accent group, the position of the phonic group within the sentence and the type of break at the end of the group, while the concatenation cost measures mainly the continuity of f0 at the joint point. Finally, the sequence of accent groups with the best score taking into account all the target and concatenation costs is selected.

4. COMPARISON OF METHODS

Comparing two different approaches describing the same phenomenon is always interesting, as it can provide some very useful insight on the nature of the problem. In the superpositional models it is necessary to decompose the global pitch contour into the different contributions at analysis time, which, as mentioned before, it is not a trivial problem. At synthesis time, how to find the best combination of phrase curves and accent curves is still an open question, although there are interesting approaches like [11].

In a concatenative system the analysis of the pitch contour is a trivial problem, as you only need to find the join points, what depends directly on the chosen unit. At synthesis time it is necessary to find a suitable set of features for describing the context (what it is likely to be crucial for every intonation model), and the design of the cost functions for selection, which introduces the currently unsolved problem of the weight optimization of every subcost contribution to the global score.

The accent group contour chosen in [5] can be considered as the addition of an accent curve and a local phrase curve, in terms of the superpositional models. The clustering shown in Figure 1, that assures initial and final accent groups for the beginning and end of the phrase, respectively, implies an implicit phrase curve selection, with the intermediate groups looking for continuity with the concatenation cost function. In fact, in [3] the phrase curve is modeled as a two-part curve obtained by non linear interpolation between three points: the start of the phrase, the start of the last accent group in the phrase, and the end of the phrase. So, on considering these points in the clustering, an appropriate declination according to the type of sentence is chosen, avoiding the need for decomposing the original contour and finally adding the different contributions in a process very likely to be prone to errors.

However, there is an important flaw in the concatenative methods: their behavior in the presence of discontinuities. In a superpositional framework this problem does not exit, as the explicit use of the phrase curve assures continuity in every point of the pitch contour. The intonation unit selection approach in [5] minimizes this problem on considering multiple candidates for every target unit, and it has shown a very good performance in the domain of neutral speech with a reasonable sized prosodic corpus (about one hour), but the problem of infrequent pitch jumps is potentially still there. Moreover, it can be argued that it is just a problem of corpus coverage, but the application to other domains such as expressive speech, where the larger variability would likely imply more discontinuities, makes it important to research some method for treating these cases where the unit selection algorithm can not find a suitable sequence of intonation units.

5. PITCH CONTOUR MODIFICATION

Summing up, it would be desirable to find a method that uses natural units as often as possible, while avoiding discontinuities. As mentioned before, the accent group employed in [5] includes both phrase and accent curves from the superpositional model. This suggests the possibility of combining the basics of both methods, performing a unit selection for obtaining the better sequence of natural contours and applying the general assumption of the superpositional models of the different components for fixing the existing discontinuities. So, in the presence of a discontinuity, we could maintain the accent curve component of the chosen accent group, and add it to a new local phrase curve computed in order to balance the discontinuity.

This approach would solve the problem of the discontinuities, but we would still need to decompose the candidate pitch contours. Moreover, now we would also have to make a local phrase curve estimation from the implicit one defined by the surrounding accent groups.

In [8] it is assumed that the phrase curve can be approxi-

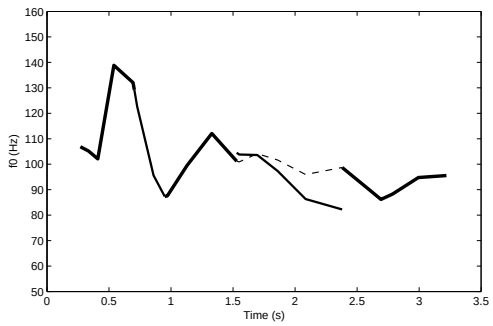


Figura 2. *Modification of an accent group pitch contour*

mated by n line segments, where n is the number of feet in the phrase. Although the left-headed foot used there is defined as an accented syllable followed by one or more unaccented syllables, and therefore its temporal scope does not coincide with the accent group defined in section 3, we will still assume this approximation to be valid for our case. Then, if the phrase curve is a line segment in the scope of every accent group, the pitch contour modification can be extremely easy. First, if the intonation unit selection algorithm is well designed, and the prosodic corpus can provide a sequence of accent groups without very large discontinuities between groups, the desired local phrase curve can be reasonably approximated by the segment line joining the desired two points of concatenation with the surrounding contours. And second, being the original phrase curve of the accent group a segment line, and given that the addition of two segment lines is another segment line, the pitch contour modification is reduced to subtract from the accent group the segment line that compensates the differences in f0 in both concatenation points. In other words, it is not necessary to compute the target local phrase curve and the original phrase curve.

Equations (1), (2) and (3) show the algorithm for computing the pitch contour modification $\Delta(t)$ as a function of the differences $D_{initial}$ and D_{final} at times $t_{initial}$ and t_{final}, corresponding to the discontinuities at the beginning and end of the accent group. The differences $D_{initial}$ and D_{final} are computed subtracting the original value of the accent group from the desired value at the concatenation points.

$$\Delta(t) = m \times t + b \tag{1}$$

where the slope m is

$$m = \frac{D_{final} - D_{initial}}{t_{final} - t_{initial}} \tag{2}$$

and the intercept b

$$b = D_{initial} - m \times t_{initial} \tag{3}$$

Then, the pitch contour f_0 of the accent group is modified in the following manner:

$$f(t) = f_0(t) + \Delta(t) \tag{4}$$

Figure 2 shows an example of the application of this modification method. The pitch contour resulting from the intonation unit selection is represented by a solid line. The accent group located around 1.5 and 2.5 seconds exhibits a discontinuity of about -4 Hz at the beginning and 15 Hz at the end. The modified contour is represented by a dotted line.

6. PERCEPTUAL TESTS

A listening test was conducted in order to check the performance of the modification procedure. 30 sentences were manually extracted from our Galician speech corpus, looking for accent groups belonging to the same class (see Figure 1) and with similar context (number of syllables, position of the accent group into the phonic group, number of words, etc), and big discontinuities $D_{initial}$ and D_{final} in the points of concatenation. From this set, 12 sentences were declarative, 10 interrogative, 6 exclamatory and 2 ellipsis. Table 1 shows the statistics of the discontinuities $D_{initial}$ and D_{final} before modification, while the $Slope$ term is the difference between both discontinuities divided by the duration of the new contour in its original context, and gives an idea of the rate of change.

Tabla 1. *Perceptual test sentences statistics*

	Mean	Std. Dev	$\|Max\|$	$\|Min\|$
$D_{initial}\ (Hz)$	-4.53	14.18	30.97	2.27
$D_{final}\ (Hz)$	-2.78	12.45	21.39	4.72
$Slope\ (Hz/s)$	-2.04	32.55	114.37	2.62

For every sentence two stimuli were obtained as the output of a unit-selection speech synthesizer ([12], [13]), the first one imposing its original intonation contour, and the second one imposing the new one resulting from the change of an accent group contour and the subsequent application of the proposed modification method.

The perceptual test was a typical CMOS (Comparison Mean Opinion Score) where every listener was asked to evaluate each pair of stimuli according to a 5 point scale, ranging from "Stimulus A much better than B" to "Stimulus B much better than A", taking into account only the naturalness of the synthetic intonation curves (Table 2). The order of presentation of every pair of stimuli was randomized, as well as the stimuli themselves within each pair. 14 native or bilingual listeners participated in the test, 6 of them involved in speech research.

1	Stimulus A much better than B
2	Stimulus A better than B
3	Similar or equal
4	Stimulus B better than A
5	Stimulus B much better than A

Tabla 2. Perceptual test scale

7. DISCUSSION

Figure 3 shows the results of the perceptual test, taking the natural contours as the reference. About 80% of the evaluations considered that the modified stimuli were not at least worst than the original ones, which is a very good result if we take into account that the modification method had to deal with discontinuities as large as 30 Hz in some cases (see Table 1). Although the differences between the percentages of evaluations considering worse (including much worse) and better (including much better) were found to be statistically significant after performing a one-sample binomial proportion test (p-value = 0.0133), the authors argue that some of the differences found in the test can be explained by listeners' evaluation based more on their own preferences than on the naturalness of the synthetic contours, as suggest the funny result of about 11% of modified contours being more natural than the original ones. Moreover, another source of

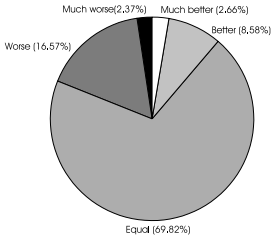


Figura 3. *Results from the perceptual test*

distortion comes from the use of a corpus based speech system for obtaining the stimuli. On using the original contours for the acoustic unit selection it was very likely to get the sequence of units extracted from that same sentence, producing a synthetic stimulus with a higher global quality than the stimulus with the modified contour. Finally, with respect to the sentences, only the modified versions of 2 of the remaining 30 were considered to be worse by the majority of the listeners.

8. CONCLUSIONS

In this paper we presented a novel method for modifying the pitch contour of an accent group, in order to avoid the discontinuities that degrade drastically the naturalness of the global synthetic curve. With this goal in mind, we combined the output of an intonation unit selection system ([5]) with the typical decomposition of the superpositional models ([3]), and proposed a local modification of the accent group phrase curve for eliminating the discontinuities. This way, natural contours are used as far as possible, and some of them are modified when required without having to decompose them explicitly into their different components.

The possibility of modifying the pitch contours gives more freedom to a unit selection intonation model, because the f0 continuity is a very hard constraint in the design and optimization of the cost functions. Nevertheless, it is still an important factor, as the local phrase curve estimation described in section 5 requires some continuity in the sequence of chosen accent group contours to be a reasonable approximation. Anyway, this point deserves more research effort.

Note that the choice of the accent group as the basic unit permits this simple modification technique. For instance, smoothing discontinuities using the segment instead ([6]) would possibly be much harder.

The subjective results showed a very good performance of the modification method, even in the presence of quite large discontinuities. Although different types of sentences were included in the test, looking for variability, the experiment was restricted to neutral speech. A similar perceptual test with expressive speech remains as a very interesting future work. A good behavior of the modification method there would present the unit selection technique as a very good candidate for intonation modeling for every domain and without relying on the availability of a very large prosodic corpus.

9. ACKNOWLEDGMENTS

The work reported here was carried out while a visiting research post doc at Center for Spoken Language Understanding of the first author, with funds from Dirección Xeral de Investigación, Desenvolvemento e Innovación, Consellería de Innovación e Industria, Xunta de Galicia, and support from NSF grant 0205731 "ITR: Prosody Generation for Child Oriented Speech Synthesis" to Jan van Santen.

10. REFERENCES

[1] Black, A. and Taylor, P. "Automatically clustering similar units for unit selection in speech synthesis". In Proceedings of EUROSPEECH'97, p601–604, Rhodes, 1997.

[2] Fujisaki, H., "Dynamic characteristics of voice fundamental frequency in speech and singing". In Peter MacNeilage, editor, The Production of Speech, p39–55, Springer, New York, NY, 1983.

[3] van Santen, J., and Möbius, B., "A quantitative model of f0 generation and alignment". In Botinis, editor, Intonation Analysis, Modelling and Technology, chapter 12, p269–288, Kluwer Academic Publishers, Netherlands, 1999.

[4] Meron, J., "Prosodic unit selection using an imitation speech database". In 4th ISCA Tutorial and Research Workshop on Speech Synthesis, Perthshire, Scotland, 2001.

[5] Campillo, F. and R. Banga, E., "Combined prosody and unit selection for Corpus-based text-to-speech systems". In Proceedings of ICSLP'02, p141–144, Denver, 2002.

[6] Raux, A. and Black, A., "A unit selection approach to F0 modeling and its applications to emphasis". In Proceedings of ASRU 2003, St Thomas, US Virgin Islands, 2003.

[7] van Santen, J., Mishra, T. and Klabbers, E., "Estimating phrase curves in the general superpositional intonation model". In Proceedings of the ISCA Speech Synthesis Workshop'04, Pittsburgh, PA, 2004.

[8] Mishra, T., van Santen, J. and Klabbers, E., "Decomposition of pitch curves in the general superpositional intonation model". Prosody 2006, Dresden, Germany, 2006.

[9] Black, A. and Campbell, N., "Optimising selection of units from speech databases for concatenative synthesis". In Proceedings of EUROSPEECH'95, p581–584, Madrid, 1995.

[10] López, E., "Estudio de técnicas de procesado lingüístico y acústico para sistemas de conversión texto voz en Español basados en concatenación de unidades". PhD thesis, E.T.S.I. de Telecomunicaciones, Universidad Politécnica de Madrid, España, 1993.

[11] van Santen, J., Kain, A., Klabbers, E. and Mishra, T. "Synthesis of prosody using multi-level unit sequences". In Speech Communication, Volume 46, p365–375, 2005.

[12] Campillo, F. and Banga, E. R., "On the design of the cost functions for a unit selection speech synthesis". In Proceedings of EUROSPEECH'03, p289–292, Geneva, 2003.

[13] Campillo, F. and R. Banga, E., "A method for combining intonation modelling and speech unit selection in corpus-based speech synthesis systems". Speech Communication 48, pag. 941-956, 2006.

LETTER-TO-SOUND CONVERSION FOR GALICIAN TTS SYSTEMS

Daniela Braga[1] and Luís Coelho[2]

[1] Universidade da Coruña, [2] Instituto Politécnico do Porto

ABSTRACT

In this paper, a linguistically rule-based letter-to-sound (LTS) conversion algorithm is described for Galician language. A complete set of phonological transcription rules regarding the Galician standard variety is presented. A SAMPA computer readable phonetic alphabet for Galician is also proposed. The algorithm was implemented and tested by using CORGA text materials. The obtained experimental results gave rise to 98.5% of accuracy rate. The main errors were identified and possible solutions are discussed in order to increase this value. It is also demonstrated in our work that there is a very good matching between Galician, European Portuguese and Brazilian Portuguese LTS conversion algorithms. Our purpose with this work is to develop a module that can improve synthetic speech naturalness in Galician. Other applications of this system can be expected like language teaching/learning. These results, have proved the dramatic importance of linguistic knowledge on the development of Text-to-Speech systems (TTS).

1. INTRODUCTION

Several frameworks have been proposed to tackle the grapheme-to-phone transcription module of a Text-to-Speech system, among which it can be mentioned: information theoretic systems such as decision trees [1], automatically-trained decision trees [2], table look-up models [3], dictionary-based approaches [4], linguistically rule-based modules [5], hybrid systems [6], neural networks approaches [7] Finite State Transducers (FST) [8] statistical approaches [9] and HMMs [10]. A comparison of different techniques and results is made by Damper et al [11].

One of the most preferred approaches is dictionary-based, which uses a large dictionary containing the phonetic transcription of a given number of words. This technique has been widely applied to languages whose orthography is roughly phonetically-based, such as English or French. But this approach fails when new words that are not in the dictionary come up.

For European Portuguese language (EP), some synthesizers have been proposed since the early 90's, based on different synthesis techniques, such as the formant-based synthesizer DIXI [12] and the Multivox [13], the articulatory-based synthesizer [14], or the concatenative-based synthesizer [15]. In every case, the LTS is an independent module of the TTS architecture and it is mostly built by using a rule-based approach [13], [16] and a Finite State Transducers approach [17].

For Galician, as far as we know, there is *Cotovía* [18], a bilingual concatenative TTS system for Spanish and Galician. However, regarding the rule-based approaches, there seem to be no publication of the complete set of rules for Galician available

Symbol	Meaning
...	Any grapheme
< x >	Grapheme or set of graphemes x
[y]	Phone or set of phones y
,	Separates options
{ x_1, x_2, x_3 }	Set of graphemes
< x_1 {x_2, x_3} >	< x_1 x_2 > or < x_1 x_3 >
< C / y >	Consonant except <y>
< C / {w, z} >	Consonant except < w> and < z>
V	Any graphic vowel (e.g. a, e, i, o, u)
C	Any graphic consonant (e.g. p, t, k, b, d, g...)
Pont	Punctuation mark (e.g. , . ! ? () - ; sp)
Ltr	Characters that are letters (e.g. a, b, c, …)
SP	Space between words
Hf	Hyphen
<(case) x >	Certain case that modifies the grapheme <x>
<(C) x>	<x> is a consonant
<(V) x>	<x> is a vowel
<(UV) x>	<x> is unvoiced
<(VO) x>	<x> is voiced
<(US) x>	<x> is an unstressed vowel
<(S) x>	<x> is a stressed vowel
<(W_bgn) x>	<x> is in word beginning
<(Prn_D) x>	Grapheme <x> in Demonstrative Pronoun (e.g. este(s), ese(s), aquel(s))

Table 1. Annotation symbols and conventions used in the LTS algorithm for Galician.

There is a large tradition of phonological and phonetic studies in Galician with recognized theoretical and experimental results [19], [20], [21]. However, when regarding the application of this knowledge to TTS systems in Galician, a big gap remains to be filled, specifically in what concerns practical rules that allow building a LTS module.

As far as we know through publications, this phonological knowledge is only being applied to EP and BP TTS systems [22], [23] respectively.

We justify our rule-based approach mainly because: we consider Galician as a variety of Portuguese, (together with a large list of authors); Galician, like EP and BP varieties, has a good phonological regularity; Galician, unlike Portuguese, has less orthographic deviation, due to the late orthography fixation and ongoing normalization; a rule approach is more economic than a phonetic dictionary in terms of space required to store the function; a rule approach is always able to read a new word; a good set of rules based in a linguistic description can cover almost any transcription problem; our rule approach proved to have a good rate of accuracy in EP (98.80%) [22] and BP (97.44%) [23] and most of the rules can be adapted to Galician and other varieties of Portuguese..

For our algorithm construction, studies on recent Galician phonology and phonetics were considered [19], [20], [21] and some sandhi phenomena were already included.

2. LETTER-TO-SOUND ALGORITHM

2.1. Annotation Conventions

In Table 1, the writing conventions used in the LTS algorithm proposed in section 2.2. are presented.

phoneme class	symbol	example
Vowels	/a/	casa, cama, cal, acto
	/i/	río, fío, ir, digo
	/u/	xunta, último
	/E/	ferro, almacén, mel, exame,
	/e/	rochedo, ver, vello
	/O/	voz, caracol
	/o/	curioso, don
Semivowels	/j/	queixo, loita, piano, fiestra
	/w/	louvar, mouro, ruela, suor
Plosive consonants	/p/	pé, campo
	/t/	trato
	/k/	carpa, aqui, kantiano
	/b/	bebo,
	/d/	caldo, morder
	/g/	gato, figo
Affricate c.	/tS/	chamar, achar
Fricate consonants	/f/	faro, feliz, café
	/T/	fácil, zarzallo, macio, luz
	/s/	selo, cousa, cuspir
	/S/	xa, xente, muxica
Nasal consonants	/m/	memoria, campo
	/n/	nada, onte
	/J/	brańa
	/N/	unha, algunha
Liquid consonants	/l/	lúa, álxebra, colgar
	/L/	valla, muller
	/r/	coro, carta
	/rr/	rato, melro, aforrar

Table 2. SAMPA for Galician.

Some cases were previously defined in the system, such as Vowels, Consonants, Voiced and Unvoiced Consonants, Stressed and Unstressed Vowels/Syllables and Demonstrative Adjectives.

For phonetic annotation, we proposed a SAMPA transcription set of symbols for Galician (Table 2), based on SAMPA alphabets [21] for Portuguese and Spanish, with some phonetic extensions: [l*] to represent the velar lateral consonant <sal>, "*salt*"; /~/, a diacritic to represent the vowel nasalization caused by the proximity of nasal consonants; /z/ and /T*/ to assign voicing to fricatives /s/ and /T/ respectively.

The Galician SAMPA we proposed was mainly phonological-based, differently from what we previously used for EP and BP which was more phonetic-based, due to the great number of allophones considered in standard Galician [19]. This allophonic variation was eliminated because these context dependent parameters are automatically extracted by the voice font unit selection training system that will feed our synthesis engine.

2.2. Transcription Rules

In the following Tables 3 to 10 a complete set of transcription rules for every grapheme used in Galician is proposed and some examples of occurrence are given.

#	graphical pattern <a>	phon	example
1	... < a, á, à >...	/a/	casa, cama, cal, pau, acto, cá
#	**graphical pattern <b>**	**phon**	**example**
1	... <b>...	/b/	ben, pobre
#	**graphical pattern <c>**	**phon**	**example**
1	...<ch>...	/tS/	chuvia, achar, che
2	... < c > < e, i >...	/T/	certo, cinco
3	... <c>...	/k/	cando, caeu, actos, dialectal
#	**graphical pattern <d>**	**phon**	**example**
1	... <d>...	/d/	despois, caldo, vida

Table 3. Transcription rules for <a>, <b>, <c>, <d>.

The Galician alphabet is composed of 23 graphemes and 6 digraphs (<<ch>, <gu>, <ll>, <nh>, <qu>, <rr>), which were included in the single grapheme patterns. All graphical patterns were considered, including graphemes that are likely to occur in foreign words (such as <j>, <k>, <y>, <w>, cfr. Table 10).

Raising and falling diphthongs were also beard in mind, although some of them are incorporated in the rules of graphemes <i> and <u>. Most of the rules presented are stated in literature [22] for the Galician standard variety. Every phonetic output was confirmed by a native Galician linguist, this way assuring the correct transcription of all graphic patterns.

#	graphical pattern <e>	phon	Example
1	... <SP, Pont> <e> <SP>...	/E/	ti e eu
2	...<(Prn_D), (S) e>...	/e/	este, ese
3	...<(W_bgn) e><x>...	/E/	extra, exame
4	...<(S) e><C>< i,u><V/ {i,u}><Pont>...	/E/	seria, recua,
5	...<(S) e><C><C/ {m,n}>< i,u><V/i,u><Pont>...	/E/	modestia, controversia
6	...<e>< n><C, Pont>...	/e/	meten, cóllenno
7	...<é>< n><Pont>...	/E/	tamén, vén
8	... <e> <n> < za, cio, cia><s, Pont>...	/E/	ciencia, pertenza
9	... < é > <s><Pont>...	/e/	lugués
10	... < e > <s, z> <a><s, Pont.>...	/e/	luguesa, avareza
11	... < e > <do><s, Pont.>...	/e/	penedo,
12	...<e><m,n><e><Pont>. .	/E/	gene, leme
13	...<(S) e><r><Pont>...	/e/	ver, saber
14	...<(S) e><u, o><Pont>....	/e/	seo, freo, temeu
15	...< (S) e ><la><Pont>...	/E/	vela, cadela, aquela
16	...<e><l><C, Pont>...	/E/	papel, pel, felpa
17	...<i><e>...	/E/	fiestra
18	...<(S) e><i>...	/e/	cheira, queira
19	... < é >...	/E/	léxico, vixésimo
20	...<é, e><(ct, cn,)>...	/E/	dialecto, técnica
21	...<(US) e><o, a>...	/e/	aldea, real, feo
23	...<(S) e><ll, ñ, ch, x>...	/e/	vello, pecho
24	... < (US) e >...	/e/	despois, español
25	... < (US) e ><Ltr, Pont>...	/e/	chocolate

Table 4. Transcription rules for <e>.

When conceiving these rules, we tried to design them, as much as possible, according to graphic patterns, this way reducing their dependence from other modules of text processing we are developing, such as the tonic syllable marker and the syllable divisor. Main exceptions to some of the rules are also presented.

The transcription of the 27 graphemes and 6 digraphs that are likely to appear in Galician texts is performed by a total amount of 93 rules.

Most of the problems in Galician LTS algorithm, similarly to what happens in Portuguese language, have to do with the vocalic alternations between /e/ and /E/ and /o/ and /O/.

#	graphical pattern <f>	phon	Example
1	... < f >...	/f/	fai, café, flor, naíf
#	**graphical pattern <g>**	**phon**	**Example**
1	... < g u > < e, i >...	/g/	guerra, guindaste
2	... < g >...	/g/	figo, Galiza, fungo
#	**graphical pattern <h>**	**phon**	**Example**
1	... < h >...	/ /	hoxe, haber
#	**graphical pattern <i>**	**phon**	**Example**
1	...<i><ño, ña>...	/i~/	padriño, morriña
2	... <V/{i, u}> <i>...	/j/	caixa, papeis, loita
3	...<i> <V/{i, u}>...	/j/	fiestra, liorta, canción
4	... < i, í >...	/i/	río, fío, ir, digo

Table 5. Transcription rules for <f>, <g>, <h>,<i>.

#	graphical pattern <l>	phon	Example
1	... < l > <C/h, Pont>...	/l*/	calma, último, alzar, colgar
2	... < ll >...	/L/	muller, mollado, vello
3	...<l>...	/l/	lúa, valado, vela
#	**graphical pattern <m>**	**phon**	**Example**
1	...<V><m><C>...	/~m/	cumprido, importante, tempo, ambos
2	... < m >...	/m/	memoria, campo
#	**graphical pattern <n>**	**phon**	**Example**
1	... <u><nh><a>...	/N~/	unha, algunha
2	...<V><n><C, Pont>...	/~n/	onte, antes, unha, don,
3	... <n>...	/n/	nai, neno, onte, once, onda
#	**graphical pattern <ñ>**	**phon**	**Example**
1	...<ñ>...	/~J/	braña, niño, mariñeiro

Table 6. Transcription rules for <l>, <m>, <n>, <ñ>.

They may occur for the following four reasons: 1) verbal morphology (e.g. <meto> /e/ vs <metes> /E/; <acordo> /o/ vs <acordo> /O/); 2) disambiguation of pairs of homographs either with a different grammatical category (noun/verb: <comezo> /e/ vs <comezo> /E/, <acordo> /o/ vs <acordo> /O/) or with the same grammatical category (noun/noun: <besta> /e/ vs <besta> /E/; 3) etymological reasons that can not be ruled (e.g. the tonic vowel in <ferro> is /E/ while in <negro> is /e/); 4) metaphony, a phonetic phenomenon widely spread in Galician and Portuguese, that consists in the regressive and assimilative influence performed by the final unstressed word vowel over the opened tonic vowel. The final closed vowel closes the tonic

vowel a little bit /E/>/e/ (e.g. <medo>) and /O/>/o/ (e.g. novo).

#	graphical pattern <o>	phon	Example
1	....<ó>...	/O/[1]	código, só, lóxico
2	…< (W_bgn) h> <o> <r, s, t>…	/O/[2]	horario, hora, hotel
3	...<o, ó><n>><es, C, Pont>...	[o]	don, acción, papón
4	…<(S) o><r><es, Pont>…	/o/	cor, dor, mellor, mellores
5	…<o><z>< Pont>…	/O/	voz, atroz
6	…<o><ces>< Pont>…	/O/	voces, noces
7	…<o><so><s, Pont>…	/o/	curioso, desexoso
8	…<o><sa> <s, Pont>…	/O/	desexosa, xeitosa
9	…<o><l, la><C, Pont>…	/O/	caracol, mol, farola
10	…<o><i><s><Pont>…	/O/	caracois
11	.. ..<(S) o><C>< i ><V/ {i,u}><Pont>...	/O/	ocio, odio, serodio
12	...<o><u>...	/o/	mouro, ouro
13	...<o><i>...	/o/	loita, coitado
14	…<Ltr><o><l><C/h> …	/o/	soltar, volver
15	...<(S) o><a>...	/o/	coroa, boa
16	...<(S) o><a>...	/o/	coroa, boa
17	...<o><(pt, bt, bx, ct, cl, gn)>...	/O/	optar, obter, obxectar,
18	... <(US) o > <s, Ltr, Pont, SP>...	/o/	veciño, carro, europeo
19	... <(US) o >...	/o/	asobío, polir
#	graphical pattern <p>	phon	Example
1	... < p >...	/p/	pato, polo, carpa

Table 7. Transcription rules for <o>, <p>.

#	graphical pattern <q>	phon	Example
1	...<qu><e,i>...	/k/	esquina, quen, queixo
2	... < q >...	/k/	quorum
#	graphical pattern <r>	phon	Example
1	... < r r > ...	/rr/	aforrar, curro
2	...<(W_bgn) r>...	/rr/	rúa, ría
3	... <l,n><r>…	/rr/	honra, melro
4	... < r >...	/r/	mar, cara, fraco
#	graphical pattern <s>	phon	Example
1	…<s><C_VO>…	/z/[3]	prosma, fasllo
2	…<s><SP><C_VO>…	/z/	estás desperto
3	...<s>...	/s/	selo, cousa, cuspir
#	graphical pattern <t>	phon	Example
1	… <t>…	/t/	corte, tamén

Table 8. Transcription rules for <q>, <r>, <s>, <t>.

[1] Exception to this rule: infinitive forms of <pór> /o/.
[2] Exception to this rule: <hoxe> /o/.
[3] SAMPA extension for the voiced alveolar fricative.

#	graphical pattern <u>	phon	Example
1	... < ü > ...	/w/	lingüista, pingüe
2	... <V/u>< u >...	/w/	cauto, ceu, pouco, fuxiu
3	...<u><V/u>	/w/	minguar, ruela, tenue, suor, frecuente
4	... < ú, u >...	/u/	túa, rúa, pulo, miúdo
#	graphical pattern <v>	phon	Example
1	... < v >...	/b/	avogado, volveu, vostede
#	graphical pattern <x>	phon	Example
1	List of exceptions	/ks/	axila
2	...<(W_bgn) ex, exo, extra, taxi, xeno, xilo>...	/ks/	extra, exame, excedente, xenofobia, xilófono
3	...< (W_bgn) ex><V><x>...	/ks/	exaxerar, exixir, exexese, exixencia
4	...< x>...	/S/	xamón, xente, obxecto, relox
#	graphical pattern <z>	phon	Example
1	... < z SP> <C_VO>...	/T*/[4]	cruz verde, cruz branca
2	... <z>...	/T/	zunir, prezo, luz

Table 9. Transcription rules for <u>, <v>, <x>, <z>.

#	graphical pattern <j>	phon	Example
1	... <j>...	/dZ/	jeep, jazz, judo
#	graphical pattern <k>	phon	Example
1	... < k>...	/k/	karate, kamikaze
#	graphical pattern <y>	phon	Example
1	….<(W_bgn) y>…	/jj/	yen, yang
2	...<y>...	/i/	baby, bodyboard
#	graphical pattern <w>	phon	Example
1	…<w>…	/w/	walkie-talkie, whisky

Table 10. Transcription rules for foreign graphemes <j>, <j>, <y>, <w>.

Grapheme <x> is also of difficult transcription because of etymological reasons. The following list of exceptions and their morphological variations was created in which <x> is pronounced /ks/: *anglosaxón*,

[4] SAMPA extension for the voiced dental fricative.

aproximar, asfixia, axila, axilar,axioma, bórax, clímax, complexo, convexo, crucifixo, elixir, exacto, exame, exaxerar, exceder, excelente, excepto, exclamar, exilio, eximir, éxodo, exótico, explosión, expurgar, extenso, extensión, extensor, extra, fax, galaxia, hexágono, laxo, léxico, maxilar, nexo, reflexionar, prefixo, reflexo, saxón, sexo, sexual, sílex, sintaxe, sufixo, taxi, texto, textual, tórax, tóxico.

2.3. Algorithm Implementation

The time needed to generate an acoustic utterance from a text input should be as short as possible. In order to achieve the best performance in the implementation process, we tried to group the rules/patterns according to the type of logical test and common characters. In our system, an average of three tree nodes evaluations is needed for reaching the output pattern.

As much as possible, the rules that were leading to the default output were not implemented. The default output is chosen to be the most frequent among the total outputs for a given grapheme. With this procedure, from the total 93 rules presented rules only nearly 75 are needed. Temporal performance tests with other LTS systems are foreseen.

Compared with the dictionary based approach, our system requires less memory space and can cope with any new word which makes it a better option for large vocabulary Galician TTS systems. Statistical approaches devise transcription rules from a given set of training utterances, but this usually leads to scattered and complex conditions that should be re-organized. This is a good and universal option when little or no knowledge of the language exists.

3. RESULTS AND DISCUSSION

The proposed algorithm was implemented and tested using texts from the CORGA (Corpus de Referencia do Galego Actual) [25]. This corpus is composed by several types of texts, extracted from newspapers, books magazines, journals and transcriptions of oral data, covering different areas from politics to technology or fiction. A training text containing a total number of 10289 characters distributed among 2121 words was used as the text input of our system. The phones originated by our system were manually checked and 98.50% were correctly converted.

The errors were marked and classified as shown on Table 11. From a total of 1.50% of errors (182 wrongly transcribed graphemes), 0.21% are exceptions that were still not implemented for lack of time. Most of the errors come from <e> and <o> grapheme transcription, since these graphemes have a high variability in Galician, as in Portuguese. Many of them occur in homographs (e.g. noun <selo> e/ vs verb <selo> /E/) and words with sub-morphemic alternation caused by metaphony (e.g. <sogro> /o/ vs <sogros> /O/).

Other mistakes in <e> and <o> grapheme's transcription are due to vocalic alternations which occur along the conjugation of the verbs (e.g. <eu bebo> /e/, <ti bebes> /E/, <el bebe> /E/). This alternation is well known and can also be predicted by using linguistic information. Foreign words also represent a problem to be considered in Galician LTS conversion. However, Galician seems to admit less foreign words than EP and BP.

Type of error	# occur.	% occur.
Homographs with <e>	26	0.21
Mistakes in <e> along verbal conjugation	15	0.12
Mistakes in <o> along verbal conjugation	18	0.15
Mistake in <e> transcription	32	0.26
Mistake in <o> transcription	32	0.26
Mistake in <u> transcription?	10	0.08
Foreign words	12	0.10
Non implemented exceptions	26	0.21
Acronyms	11	0.09
Total	**182**	**98.50**

Table 11. Errors in absolute frequency and percentage relative to total grapheme nr.

Comparisons with other LTS modules for Portuguese TTS systems seem to confirm the good performance of our rule-based approach. Besides, our system is more optimized when compared with other similar frameworks, since we only deal with 93 rules, when others need around 600 rules, as stated in [13].

Our good results are very comparable to the LTS for EP [22], with an accuracy rate of 98.80%. The LTS system [23] has an accuracy rate of 97.44%. Other rule-based frameworks don't mention accuracy rates [12], [13], [26].

4. CONCLUSIONS

In this paper, we described a LTS transcriber linguistically rule-based that is perfectly able to be applied to any Galician TTS system. The proposed rules were implemented and tested using texts from CORGA corpus. Evaluation of our system resulted in 98.50% of correct transcribed graphemes. Most of the errors were found in homographs, verbal conjugations, metaphony affected words and foreign words.

From the obtained results, it is our belief that morphological decision in terms of verb/noun will drastically improve the efficiency of our LTS system. This study is already ongoing. Phonological information regarding the verbal flexion of the verbs can also be included in future developments. These problems are

fully shared by EP, BP and Galician. A careful research on the phonological behavior of the foreign words in Galician is also foreseen. Although some sandhi phenomena occurring between consonants in word boundaries were included, other phenomena regarding vocalic neighboring are soon expected to be implemented

The proposed algorithm is an adaptation of the LTS for EP proposed elsewhere [22]. From the overall set of rules presented, 80.2% are fully applicable, 19.1% have the same input pattern but a different output. This means that we have 95.3% of pattern compatibility. We also intend to evaluate the compatibility of our patterns with other varieties of Portuguese.

A comparison of our system with another one using automatically-generated rules obtained by decision trees, for instance, is foreseen.

9. ACKNOWLEDMENTS

We would like to thank to Professor Xosé Ramón Freixeiro Mato for his unconditional scientific support on this work. We would also like to acknowledge *Centro Ramón Piñeiro para a Investigación en Humanidades* for having allowed us to use full texts of CORGA corpus.

10. REFERENCES

[1] J. M. Lucassen, R. L. Mercer, "Discovering phonemic based forms automatically: an information theoretic approach" in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 42.5.1-42.5.4, 1984.

[2] Black, A., Lenzo, K. and Pagel, V. "Issues in Building General Letter to Sound Rules", in *3rd ESCA Workshop on Speech Synthesis*, Jenolan Caves, Australia, pp. 77-80, 1988.

[3] A. Bosch and W. Daelemans, "Data-oriented methods for grapheme-to-phoneme conversion". in *Sixth Conference of the European Chapter of the Association for Computational Linguistics (EACL'93)*, pp. 45-53, 1993.

[4] Cecil H. Coker, Kenneth W. Church, and Mark Y. Liberman, "Morphology and rhyming: Two powerful alternatives to letter-to-sound rules for speech synthesis". in *Conference on Speech Synthesis. European Speech Communication Association*, 1990.

[5] R. M. Kaplan and M. Kay, "Regular models of phonological rule systems", in *Computational Linguistics 20(3)*, pp. 331-378, 1994.

[6] H. M. Meng, S. Seneff, and V. Zue, "Phonological parsing for bi-directional letter-to-sound/sound-to-letter generation", in *ARPA Human Language Technology Workshop*, Princeton, USA, 1994.

[7] T.J. Sejnowski and C. R. Rosenberg. "Parallel networks that learn to pronounce English Text". In *Complex Systems, 1*, pp. 145-168, 1987,.

[8] E. Roche and Y. Schabes, *Exact Generalization of Finite-State Transductions: Application to Grapheme-to-Phoneme Transcription*. Technical Report TR-95-08, Mitsubishi Electric Research Laboratories, Cambridge, USA, March 1995.

[9] A. Chotimongkol, and A. Black, "Statistically trained orthographic to sound models for Thai", in *ICSLP2000*, Beijing, China, 2000.

[10] P. Taylor**,** Grapheme-to-Phoneme conversion using Hidden Markov models, in *Proc. Interspeech 2005*, Lisbon, Portugal, September 2005.

[11] R.I. Damper, Y. Marchand, M.J. Adamson, K. Gustafson, "Comparative Evaluation of Letter-To-Sound Conversion Techniques For English Text-To-Speech Synthesis", In *Proceedings of 3rd ESCA/COCOSDA International Workshop on Speech Synthesis*, Jenolan Caves, Australia, pp. 53-58, 1998.

[12] L. C. Oliveira, *Síntese de Fala a Partir de Texto*, Ph.D. dissertation, Instituto Superior Técnico, Universidade Técnica de Lisboa, 1996.

[13] J. P. Teixeira, D. Freitas, P. Gouveia, G. Olaszy, G. Németh, "Multivox: Conversor Texto-Fala para Português", in *Proceedings of PROPOR'98*, Porto Alegre, Brazil, November 1998.

[14] A. J. S. Teixeira, Síntese Articulatória das vogais nasais do Português Europeu, Ph.D. dissertation, Universidade de Aveiro, 2000.

[15] M. J. Barros, *Estudo comparativo e técnicas de geração de sinal para a síntese da fala*, M.S. thesis, Faculdade de Engenharia da Universidade do Porto, 2001.

[16] L. C. Oliveira, M. C. Viana, I. M. Trancoso, "A Rule-Based Text-to-Speech System for Portuguese" *Proc. ICASSP'92 - Int. Conf. on Acoustics, Speech, and Signal Processing*, San Francisco, U.S.A., March 1992.

[17] D. A. Caseiro, I. Trancoso, "Grapheme-to-Phone Using Finite-State Transducers", in *2002 IEEE Workshop on Speech Synthesis*, September, 2002.

[18] Cotovía available at: http://corpus.cirp.es/corgaxml/

[19] X.R.Freixeiro Mato, *Gramática da Lingua Galega I. Fonética e Fonoloxía*. Edicións a Nosa Terra, Vigo, 1998.

[20] R. Alvarez Blanco, "O sistema fonolóxico do galego. Comparación co do portugués". En D. KREMER (ed.): *Actes du XVIIIème Congrès International de Linguistique et Philologie Romanes, Université de Trier, 1986)*. Vol. 3. Tübingen: Max Niemeyer Verlag, 517-530, 1991.

[21] X. L. Regueira, "Elementos para a definición dun modelo fonético estándar da lingua galega". En B. FERNÁNDEZ SALGADO: *Actas do IV Congreso Internacional de Estudios Galegos*. Vol. 1. Oxford: Centro de Estudios Galegos, 179-194, 1997.

[22] D. Braga, L. Coelho, F. G. V.Resende Jr, "A rule-based grapheme-to-phone converter for TTS systems in European Portuguese", *VI International Telecommunications Symposium (ITS2006)*, Fortaleza-CE, Brazil, September, 2006.

[23] D. Silva, A. Lima, R. Maia, D. Braga, J. F. Moraes, J. A. Moraes, F. Resende Jr. , "A rule-based grapheme-phone converter and stress determination for Brazilian Portuguese natural language processing", *VI International Telecommunications Symposium (ITS2006)*, Fortaleza-CE, Brazil, September, 2006.

[24] SAMPA available at: http://www.phon.ucl.ac.uk/home/sampa/index.htm

[25] CORGA (Corpus de Referencia do Galego Actual) available at: http://corpus.cirp.es/corgaxml/

[26] C. Oliveira, L. Moutinho, A. Teixeira, "Um novo sistema de conversão grafema-fone para PE baseado em transdutores", in *Actas do II Congresso Internacional de Fonética e Fonologia*, Maranhão, Brazil, 2004.

MAXIMUM ENTROPY MOTIVATED GRAPHEME-TO-PHONEME, STRESS AND SYLLABLE BOUNDARY PREDICTION FOR PORTUGUESE TEXT-TO-SPEECH

Maria João Barros[1] *& Christian Weiss*[2]
[1]ISEL, Polytechnic Institute of Lisbon, Portugal, [2]L2F, Inesc Id, Portugal
mjbarros@cc.isel.ipl.pt, christian.weiss@l2f.inesc-id.pt

ABSTRACT

In this paper we present a framework for grapheme-to-phoneme (G2P) conversion, stress and syllable boundary prediction for European Portuguese (EP) Text-to-Speech (TTS) Systems. For all prediction tasks Maximum-Entropy models were used for classification. Due to the need of expensive work by experts to implement rule based G2P converters there was interest in developing probabilistic models with the Maximum-Entropy approach to solve the previous mentioned symbolic pre-processing within a TTS system. The system presented in this work is a fast and flexible approach which gives good results in each of the prediction tasks, optimal for fast application development in the TTS domain. The data used for training the G2P conversion model is manually labelled from continuous speech with natural vocalic reduction and co-articulation between words effects, common in Portuguese continuous speech. The framework is used for EP but is also usable for Brazilian Portuguese (BP) where minor changes have to be done in the G2P training data whereas stress and syllable models are the same.[1]

1. INTRODUCTION

Portuguese is the sixth most spoken language in the world, with 200 million native speakers. It is the second most spoken Latin language and the third language spoken in Occidental world. It is a Romanic language that is spoken in Angola, Brazil, Cape Verde, Indian Union, Guinea-Bissau, Macau, Mozambique, Portugal, Sao Tome & Principe and Timor Lorosa'e. It is the official language in eight countries and largely used as second language in many others.

Automatically translation of any orthographic word into its phonetic representation is needed. This is mainly known as G2P conversion. The challenge is to convert words in a way that each token sounds natural for each rhythm and not only according to the rules that are followed for paused speech or isolated word reading, this means considering coarticulation between words effects and phonetic reductions of continuous speech, known as Sandhi effects [1], [2].

Rule-based systems are useful to generate standard transcriptions to synthesis applications or to build new corpora for machine learning systems, but these systems are expensive while needing a linguist expert to setup all rules and exceptions needed to produce the results. Probabilistic systems are not that cost intensive and can be setup even without linguist knowledge. They show to be more flexible according to natural sounding synthetic speech of continuous speech, once their statistical models can be trained with data in such manner they were spoken in determined speaking rhythm that is intended to be used in each application, with corresponding allophones from coarticulation effects and phonetic reductions.

There is much work done in the speech processing domain like TTS or Automatic Speech recognition (ASR) for EP and BP. However, there is only few works done in probabilistic motivated G2P conversion, stress prediction and syllabification. But, as it was demonstrated, these are essential tasks for natural language processing with the major aspect on building TTS systems or speech recognition systems.

In this paper we introduce our approach to G2P, stress prediction and syllabification based on the Maximum-Entropy Framework introduced by [3], [4] for Natural Language Processing (NLP).

So far there are two rule based approaches to EP G2P conversion in the literature, one described in [5], [6] and another in [7], [8]. A neural network was introduced by [9] with fairly good results and leading to a CART based G2P conversion which was developed within the DIXI+ framework. A newer approach to G2P conversion was introduced by [10] where a weighted finite state transducer was implemented using the rules of the DIXI+. The disadvantage of generating many thousands of weighted finite-state-machines (WFSTs) resulting from this transformation the authors considered a hybrid approach using a combined knowledge-based and data driven-driven approach. There is also some work made for EP stress and syllable boundary prediction, using rule based approach as in [11] for syllable and [12] for stress prediction, or using neural networks probabilistic approach as in [13] for syllable and [14] for stress prediction.

This work was sponsored by Canon Foundation in Europe

This paper is organized as follows. In Section 2 we describe the Maximum-Entropy approach and refer to its difficulties. In Section 3 we explain the training corpus used for the EP models. A summary of G2P EP conversion is made in Section 4, of syllable boundary detection in Section 5 and of stress prediction in Section 6. Section 7 presents the actual results of our trained models and a conclusion is given in Section 8.

2. MAXIMUM-ENTROPY BASED MODEL

As noted above the Maximum Entropy Framework is a well known approach for ambiguities resolution in natural language processing where many problems can be reformulated as a classification problem. The task of such a reformulation is to include a context and to predict a correct class. The objective is to estimate a function $X \rightarrow Y$, which predicts an object $x \in X$ to its class $y \in Y$. Y represents the predefined classes for either each task of our prediction problem.

In the case of G2P conversion each phoneme of the phoneme inventory represents a class. In European Portuguese there are 38 Phonemes (see Table 1) which means that we have 38 classes.

In the field of stress prediction we are dealing with a binary classification where the class is true for stressed syllables and false for non-stressed.

The same binary classification task has to be solved in the domain of syllabification where we have a syllable boundary or not.

X consists of linguistic features where we include the context and the resulting input for the classification is a feature vector containing the object itself which has to be classified as well as the context. The classifier $X \rightarrow Y$ can be seen as a conditional probability model in the sense of

$$C(x) = \arg\max_{y} p(y|x) \qquad (1)$$

where x is the object to be classified and y is the class. Including the context we get a more complex classifier

$$C(x_1, x_2, ..., x_n) = \arg\max_{y_1 ... y_n} \prod_{i=1}^{n} p(y_i | x_1 ... x_n, y_1 ... y_{i-1}) \quad (2)$$

Where $x_1 ... x_n, y_1 ... y_{i-1}$ is the context at the i^{th} decision and y_i is the outcome.

3. THE TRAINING CORPUS

The training corpus for G2P conversion consists of 7352 orthographic words with their phonetic transcription, which although presented to the system as isolated words obey to be a transcription that considers the context of the word, or the coarticulation between words, and include all the phonetic grammar rules. The syllable boundary detection and the stress prediction corpora have respectively 4283 and 4219 phonetic words, the first with their syllable boundary information and the second with their syllable stress classification. These two corpora have a binary classification: in or not in boundary and stressed or non-stressed syllable. These classifications were annotated as *0* for no boundary or no stress; and 1, for boundary or stress.

The G2P conversion is more complex in the sense that instead of predicting binary classes, the system needs to classify 44 different classes, as can be seen in Table 2.

Some classes are a combination of more than one phoneme which is the reason for having 44 classes instead of the 38 corresponding to the number of phonemes for Portuguese, as it was explained before.

4. GRAPHEME-TO-PHONEME CONVERSION

The most important aspect of G2P conversion is the choice of the symbol inventory used for the transcription system. Although International Phonetic Alphabet (IPA) [15] is the most complete and most widely used in transcription systems, as dictionaries, the Computer Readable Phonetic Alphabet (SAMPA) [16] is usually adopted for computational systems. The reason for this is its reduced phonetic set, relating to IPA, which is enough for G2P transcriptions and reduces the systems complexity.

In Portuguese there are 20 phonetic consonants, 14 phonetic vowels and 4 semi-vowels, which are shown in Table 1, opposed to the 18 graphic consonants and 5 graphic vowels.

Oral Vowels and Semi-vowels
6, a, E, e, @, O, o u, i, j, w
Nasal Vowels and Semi-vowels
6~, e~, o~, u~, i~, j~, w~
Fricative Consonants
v, f, z, s, S, Z
Liquid Consonants
L, l, l~
Vibrant Consonants
r, R
Plosive Consonants
b, p, t, k, g, d
Nasal Consonants
m, n, J

Table1. *European Portuguese Phoneme Inventory, (in SAMPA)*

4.1. Portuguese Vowels

From the 5 graphic vowels that exist for Portuguese, <a>, <e>, <i>, <o> and <u>, the <o> with 7 possible transcriptions, /o/, /O/, /u/, /w/, /o~/, /o~j~/, /w~/, and the <e> with 11 possible transcriptions, /E/,

/e/, /@/, /6/, /i/, /j/, /6j/, /e~/, /6~j~/, /6~j~6~j~/ and "&", show to be the most complex ones according to phonetic transcription. There are about 30 grammatical rules for each, only related to isolated words, and even more rules related to the allophones that come from coarticulation effects between words.

The symbol "&" represents a dummy class and appears in those situations of mute graphemes, in our model classified as "&". For instance the word <que> can be transcribed as /k@/ being the <u> mute, or even as /k/ being the /@/ suppressed. The grapheme <u> in both situations and <e> in the last one would be transcribed as "&".

Another problem according to the graphemes phonetic transcription is homonyms, once their transcription gives many times the meaning of homonym words. For example <sede> that can be /sed@/ or /sEd@/, or <acordo> that can be /6kordw/ or /6kOrdw/ depending on the meaning of the word.

The <a> is a grapheme that, although not so complex as the <o> or the <e>, still has 3 phonetic transcriptions: /a/, /6/ and /6~/. The <i> and the <u> represent not only the phonetic vowels <i> and <u> but also the semivowels <j> and <w> respectively and all of these four in their nasal version: <i~>, <j~>, <u~> and <w~>. The <i> can also be transcribed as a <@>, for instance when there is a sequence of two syllables with <i> as in <feminino> that is transcribed as /f@m@ninw/.

4.2. Portuguese Consonants

From the 18 graphic consonants existent for Portuguese there are some particular cases that have to be well considered.

The <h> is a consonant that by itself doesn't have a sonorous transcription as it lost its aspiration with the evolution of the language, but when preceded by one of the consonants <c>, <n> or <l>, gives the consonants combinations <ch>, <nh> and <lh> that have their own particular phonetic transcriptions, /S/, /J/ and /L/ respectively.

The <m> and <n> between a vowel and a consonant or the <m> in the end of a word, don't have a phonetic transcription but they nasalize the precedent vowels.

The most difficult consonants are the <x> which has 4 possible phonetic transcriptions: /s/, /z/, /S/ or /ks/ and the <s> that can be transcribed into /s/, /z/, /S/, or even /Z/ if coarticulation between words is considered.

Also the <l> can be /l/ if followed by a vowel or /l~/ if followed by a consonant or in the end of a word. The <g> can be transcribed as /g/ or /Z/ depending on the context.

There are also cases, although these are not important to the G2P transcription problem, of two graphemes with the same phonetic transcription. For example, both <g> and <j> can be transcribed as /Z/ and both <s> and <c> as /s/.

4.3. Probabilistic Transcription

From the training results we observe the correspondence between Portuguese graphemes and phonemes that is presented in Table 2, where "&" is a dummy as explained in Subsection 4.1.

Grapheme	Phonemes
ú	u~, u
õ	o~
ô	o
ó	O
í	i~, i
ê	6~j~6~j~, e~, 6j, 6~j~, e
é	E, 6, 6~j~, e
ç	s
ã	6, 6~
â	6, 6~
á	a, &
à	a
z	Z, S, z
x	S, ks, z, s
v	v
u	@, u~, &, w, u, w~
t	t
s	Z, S, &, z, s
r	R, &, r
q	k
p	&, p
o	O, o~, @, &, w, u, w~, o
n	J, &, n
m	&, m
l	L, l~, l
k	k
j	Z
i	i~, @, &, j~, j, i
h	&
g	Z, g
f	f
e	i~, E, 6, @, e~, 6~, &, 6j, 6~j~, j~, j, i, e
d	d
c	S, &, s, k
b	b
a	a, 6, 6~, &, 6~w~, o

Table2. *Portuguese Graphemes Phonemic Transcription*

By observation of the table we can see that some graphemes transcriptions have more than one phoneme.

These phonemes sequences were considered as individual classes, what justifies having 44 classes instead the 38 correspondent to the number of EP phonemes.

Examples of these situations or others that can be not so clear are: <tem>, where the <e> is transcribed as /6~j~/ while the <m> is mute: /t6~j~/; <têm>, where the

<ê> is transcribed as /6~j~6~j~/ while the <m> is mute: /t6~j~6~j~/; <mandam>, where the second <a> is transcribed as /6~w~/: /m6~d6~w~/; <excelente>, where the first <e> is transcribed as /6j/: /6jSs@le~t@/; <além>, where <é> is transcribed as /6~j~/ while the <m> is mute: /al6~j~/; <óptimo>, where the <p> is mute: /Otimu/; situations with double consonants being transcribed as one phoneme: <fosse> that is transcribed as /fos@/ or <torre> that is transcribed as /toR@/; or consonants followed by <h> that lead to a particular phoneme: <acho> that is transcribed as /aSw/, <ilha> that is transcribed as /iL6/, <ninho> that is transcribed as /niJu/.

5. SYLLABLE BOUNDARY DETECTION

Syllable is what denominates the part of the word which is pronounced in only one voice emission and can be classified in non-stressed, pos-stressed, pre-stressed or stressed.

Portuguese syllabic segmentation follows a set of few grammatical rules, from which 6 are related to the vowels, 6 to the consonants and 3 to others cases which has to be considered [17].

Although syllable boundary detection is not a difficult task to achieve using rule-based systems, the probabilistic approach is very simple because it is a problem of binary classification: in boundary or not in boundary of syllable, and it allows considering Sandhi effects, which are very prominent in the EP language [1], [2].

6. STRESS PREDICTION

In the Portuguese grammar the words can be classified according to the stressed syllable position in the word as oxytone, paroxytone or proparoxytone, depending if the stressed syllable is the last syllable, the one before the last or the third from the end of the word, respectively.

Many times in Portuguese the stress can be given by graphic accents, but these don't always exist in words. The stress prediction, as the syllable boundary detection, is also a problem of binary classification: stressed or non-stressed syllable.

The accents rules for Portuguese can be found in the Inter-institutional style guide from the European Union Publications Office [18].

7. RESULTS

The results for the three tasks are presented by giving the Log likelihood and the performance of the system.

The G2P transcription's model results are shown in Table 3, the syllable boundary detection's model results in Table 4 and the stress prediction's model ones in Table 5.

The performance can be seen as the number of correct classified elements divided by the number of overall elements multiplied by one hundred.

	Result
Log likelihood	-12329.53
Performance	88.94 %

Table3. *Prediction results of G2P transcription*

	Result
Log likelihood	-1200.10
Performance	97.64 %

Table4. *Prediction results of syllable boundary detection*

	Result
Log likelihood	-5465.59
Performance	85.57 %

Table5. *Prediction results of stress prediction*

Besides these results another test was made to the G2P converter once this module is more complex for classification than the other two, having 44 classes or possible classifications while the other two have a binary classification.

The corpus for this test has 550 words comprising 3430 phonemes. All the phonemes from the EP inventory were covered, with different number of occurrences.

The test consisted in comparing the system results to the test corpus with the entries from the training corpus, giving the average number of phoneme errors taking into account three kind of errors: when the system replaced the phoneme by another (substitution); when the system gave a result to a phoneme that in the manually transcription is missed (insertion); and when the system gave nothing and the manually transcription gives a phoneme (deletion). This kind of measure is known in the speech recognition domain by word error rate (WER).

The results, given by the number of correct classified phonemes divided by the number of overall phonemes multiplied by one hundred, are shown in Table 6 and Table 7, using the confusion matrix method [19], where the rows are the actual classes and the columns the predicted classes.

As explained in Subsection 4.3, some classes are a combination of phonemes in order to represent all possible graphemes transcriptions.

-	ks	S	d	Z	k	g	t	J	v	s	b	&	z	r	l~	L	f	n	m	l	p	R
ks	87	10	-	-	-	-	-	-	-	-	-	-	3	-	-	-	-	-	-	-	-	-
S	1	96	-	-	-	-	-	-	-	2	-	-	1	-	-	-	-	-	-	-	-	-
d	-	-	99	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Z	-	4	-	93	-	-	-	-	-	-	-	-	3	-	-	-	-	-	-	-	-	-
k	-	-	-	-	97	-	-	-	-	-	1	2	-	-	-	-	-	-	-	-	-	-
g	-	-	-	2	-	98	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
t	-	-	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
J	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
v	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-
s	1	4	-	-	-	-	-	-	-	93	-	1	1	-	-	-	-	-	-	-	-	-
b	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-
z	-	3	-	-	-	-	-	-	-	3	-	-	94	-	-	-	-	-	-	-	-	-
r	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-	-
l~	-	-	-	-	-	-	-	-	-	-	-	-	-	-	83	-	-	-	-	17	-	-
L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-
f	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-
n	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	98	-	-	-	-
m	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-
l	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-	98	-	-
p	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	99	-
R	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100

Table 6. *Confusion matrix for consonants (in percent.)*

-	e~	a	E	j	j~	u	6~j~	e	6~j~6~j~	u~	&	i~	w	w~	@	i	6	O	o~	6~	6j	o
e~	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
a	-	83	-	-	-	-	-	-	-	-	-	-	-	-	-	-	17	-	-	-	-	-
E	-	-	78	1	-	-	-	8	-	-	4	-	-	-	5	4	-	-	-	-	-	-
j	-	-	-	77	2	-	-	4	-	-	6	-	-	-	-	11	-	-	-	-	-	-
j~	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
u	-	-	-	-	-	63	-	-	-	-	12	-	17	-	4	-	-	-	-	-	-	4
6~j~	-	-	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
e	-	-	-	-	-	-	-	66	-	-	27	-	-	-	7	-	-	-	-	-	-	-
6~j~6~j~	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-
u~	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-
&	-	-	1	-	-	1	-	1	-	-	90	-	3	-	2	2	-	-	-	-	-	-
i~	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-
w	-	-	-	-	-	18	-	-	-	-	17	-	59	-	-	-	-	1	-	-	-	5
w~	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-	-
@	-	-	-	-	-	-	-	4	-	-	40	-	12	-	36	4	-	-	-	-	-	4
i	-	-	-	-	-	-	-	3	-	-	-	-	-	-	-	97	-	-	-	-	-	-
6	-	8	-	-	-	-	-	-	-	-	3	-	-	-	-	-	89	-	-	-	-	-
O	-	-	-	-	-	3	-	-	-	-	3	-	5	-	-	-	-	75	-	-	-	14
o~	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-
6~	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-
6j	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-
o	-	-	-	-	-	3	-	-	-	-	2	-	2	-	-	-	-	13	-	-	-	80

Table 7. *Confusion matrix for vowels (in percentage)*

8. CONCLUSION

The vowels show to be more difficult than the consonants due to the fact of most of them having several possible transcriptions. But there are also very complex consonants as the <x> and the <s> that even the rules don't cover all of their possible cases.

As one can see in Table 6 the /ks/, /S/, /Z/, /s/ and /z/ are the most difficult cases. It is convenient to refer that although the /l/ and the /l~/ were sometimes confused with each other we don't consider these as difficult cases, because this is not perceptually relevant and even some systems don't distinguish between them.

Regarding to the vowels, from the analysis of Table 7, we see that some of the phoneme substitutions are not even errors because have no perceptual significance or are both acceptable transcriptions. For example, substituting an /i/ by an /j/ and vice versa, or an /u/ by an /w/ and vice versa is completely acceptable and it is even acceptable that these are substituted by /@/ or "&", because the system considers the phonemic transcription. This means that sometimes there are more than one possible choice.

Suppressing the /@/ means substituting the /@/ by "&", what is also reasonable and the only thing that is affected is the rhythm of the speech, not the meaning. If we attend to these considerations we can consider that the /u/ has 96% of correct values, the /w/ has 94%, the /j/ has 88% and the /@/ has 88%.

The system represents a first approach to the given tasks and can be improved by adding more data in the training corpora, but attending to the considerations made about the results in the confusion matrices the system shows to be a reliable solution and the probabilistic Maximum-Entropy Framework shows to be a good and simple approach.

9. REFERENCES

[1] D. Braga, D. Freitas, H. Ferreira: Processamento Linguístico Aplicado à Síntese da Fala. In: 3º Congresso Luso-Moçambicano de Engenharia, Maputo, Mozambique (Aug. 2003).

[2] R. Amaral, P. Carvalho, D. Caseiro, I. Trancoso, L. Oliveira: Anotação fonética automática de corpora de fala transcritos ortograficamente. In: PROPOR'99 - IV Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada, Evora, Portugal (Sep. 1999).

[3] A. Berger, S.A. Della Pietra, V.J. Della Pietra: A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics 22(1) (1996).

[4] A. Ratnarparkhi: Maximum Entropy Models for Natural Language Ambiguity Resolution. PhD Dissertation, University of Pennsylvania (1998).

[5] L. Oliveira, M. Viana, I. Trancoso: A Rule-Based Text-to-Speech System for Portuguese. In: Proc. ICASSP'92, San Francisco, USA (Mar. 1992).

[6] F. Barbosa, G. Pinto, F. G. Resende, C. A. Gonçalves, R. Monserrat, M. C. Rosa: Grapheme-Phone Transcription Algorithm for a Brazilian Portuguese TTS. In: Computational Processing of the Portuguese Language, 6th International Workshop (PROPOR 2003), Springer Verlag, Faro, Portugal (Jun. 2003) 23-30.

[7] J. P. Teixeira, D. R. Freitas, P. D. Gouveia, G. Olaszy, G. Németh: MULTIVOX – Conversor Texto Fala Para Português. In: III Encontro Para o Processamento Computacional da Língua Portuguesa Escrita e Falada, Porto Alegre, Brasil (Nov. 1998.

[8] J. P. Teixeira: Conversor Texto-Fala para o Português – Desenvolvimentos / Ferramentas. In: 1as jornadas do CEFAT, Bragança, Portugal (1998).

[9] Trancoso, M. Viana, F. Silva, G. Marques, L. Oliveira: Rule-Based vs. Neural Network Based Approaches to Letter-to-Phone Conversion for Portuguese Common and Proper Names. In: Proc. ICSLP'94, Yokohama, Japan (Sep. 1994).

[10] D. Caseiro, I. Trancoso, L. Oliveira, C. Viana: Grapheme-to-Phone using Finite-State Transducers. In: IEEE Workshop on Speech Synthesis Santa Monica, California (Sep. 2002).

[11] P. D. Gouveia, J. P. Teixeira, D. R. Freitas: Divisão Silábica Automática do Texto Escrito e Falado. In: V PROPOR – Processamento Computacional da Língua Portuguesa Escrita e Falada, S. Paulo, Brazil (Nov. 2000).

[12] J. P. Teixeira: A Prosody Model to TTS Systems. PhD dissertation, Faculty of Engineering of University of Porto, Portugal (2004).

[13] H. Meinedo, J. Neto, L. Almeida: Syllable onset detection applied to the Portuguese language. In: EUROSPEECH'99 - 6th European Conference on Speech Communication and Technology, Budapest, Hungary (Sep. 1999).

[14] C. J. Teixeira, I. M. Trancoso, A. J. Serralheiro: Accent Identification. In: ICSLP'96- 4th Int. Conf. on Spoken Language Processing, Philadelphia, USA (Oct. 1996).

[15] The International Phonetic Association, (http://www2.arts.gla.ac.uk/IPA/ipa.html)

[16] SAMPA-Speech Assessment Methods Phonetic Alphabet, (http://www.phon.ucl.ac.uk/home/sampa)

[17] "Código de Redacção Interinstitucional", (http://publications.eu.int/code/pt)

[18] http://publications.eu.int/code/pt/pt-4100100pt.htm#i137

[19] http://dms.irb.hr/tutorial/tut_mod_eval_1.php

MODELADO Y ESTIMACIÓN DE LA PROSODIA MEDIANTE RAZONAMIENTO BASADO EN CASOS

Ignasi Iriondo , J. Claudi Socoró, Lluís Formiga, Xavier Gonzalvo, Francesc Alías y Pere Miralles *

Enginyeria i Arquitectura La Salle. Universitat Ramon Llull.
Departamento de Comunicaciones y Teoría de la Senyal
Ps. Bonanova 8, 08022 Barcelona

{iriondo; jclaudi; gonzalvo; llformiga; falias; is08388}@salle.url.edu

RESUMEN

En este artículo se presenta la utilización del aprendizaje analógico, en particular el razonamiento basado en casos, como herramienta de generación automática de la prosodia a partir de texto, el cual ha sido etiquetado de forma automática con atributos prosódicos. Se trata de un método basado en corpus para el modelado cuantitativo de la prosodia y su estimación en un sistema de conversión del texto en habla. El principal objetivo es conseguir un método común para predecir los 3 rasgos prosódicos principales: la curva de frecuencia fundamental (F0), la duración segmental y la intensidad. Se ha llevado a cabo una evaluación objetiva y subjetiva para considerar su uso en el ámbito de la síntesis del habla expresiva.

1. INTRODUCCIÓN

La conversión texto en habla (CTH) tiene como finalidad la transformación automática de cualquier texto escrito en la correspondiente realización sonora. La investigación en este campo se centró inicialmente en conseguir el mayor grado de inteligibilidad posible. Posteriormente el objetivo ha sido conseguir una mayor naturalidad, es decir en la capacidad de emular la riqueza del habla humana que es intrínsecamente expresiva, ya que posee la capacidad de complementar la información verbal con una intención, actitud o estado emocional determinados. En este contexto, la mejora de la expresividad de los sistemas CTH se ha debido a avances en el modelado de la prosodia y la generación de la señal de voz de una alta calidad.

El estudio de la correlación entre el habla y la expresividad es complejo y se ha abordado desde diferentes enfoques (ver estudio comparativo en [1]). Algunos trabajos se centran en estudiar las variaciones en el timbre de voz, mientras que otros se basan en un uso para-lingüístico de la prosodia. Los estudios orientados a la síntesis del habla expresiva suelen abordar los dos enfoques, como es el caso del *Affect Editor* [2] o el trabajo de Montero et al. [3]. Además hay que tener en cuenta que en la producción de sonido humana se mezclan aspectos verbales y no verbales fruto de la evolución. La incorporación de este tipo de sonidos (*affect bursts*) se ha demostrado clave en la mejora de la expresividad del habla sintética [4].

La síntesis del habla expresiva a partir de texto requiere la generación automática de una prosodia adecuada al estilo/emoción deseado y un módulo de síntesis de voz capaz de generar habla de alta calidad con altas variaciones de los rasgos prosódicos y de los parámetros de cualidad de voz relacionados con el timbre. En la actualidad, la técnica predominante es la síntesis del habla basada en corpus o selección de unidades [5].

Este artículo presenta la utilización del Razonamiento basado en casos (CBR) aplicado a la generación automática de parámetros prosódicos (contorno de la frecuencia fundamental (F0), duración e intensidad de los fonemas) de habla expresiva en español. Por lo tanto, este trabajo está enfocado principalmente al modelado y estimación de la prosodia del habla expresiva. Con este propósito, se ha desarrollado un corpus de habla expresiva, el cual se utiliza también en el proceso de síntesis concatenativa del habla. Las medidas objetivas utilizadas para evaluar la precisión del modelado prosódico son el error cuadrático medio (RMSE) y el coeficiente de correlación (ρ). La evaluación subjetiva se ha llevado a cabo mediante un test perceptual utilizando la escala MOS. Las locuciones del test se han generado mediante la resíntesis de frases del corpus que no han formado parte del entrenamiento, a las que se les han modificado los parámetros prosódicos según la predicción obtenida automáticamente a partir de su texto.

Este artículo se organiza como sigue. La sección 2 explica el modelo prosódico, detallándose los atributos prosódicos extraídos del texto de entrada y los parámetros prosódicos a predecir. La sección 3 explica la adaptación de la técnica CBR al modelado prosódico, entrando en detalle de nuestra propuesta en la sección 4. En la sección 5 se muestran los resultados de los experimentos realizados. Y finalmente, la sección 6 proporciona las conclusiones y el trabajo futuro.

* Este trabajo ha sido parcialmente financiado por la Comisión Europea, proyecto SALERO FP6 IST-4-027122-IP.

2. MODELADO Y ESTIMACIÓN DE LA PROSODIA

Los parámetros que determinan la prosodia de un texto hablado son esencialmente la duración e intensidad segmental, el posicionamiento y duración de las pausas y el contorno de F0 [6]. En el ámbito de los sistemas CTH, la literatura en modelado prosódico es muy extensa. La curva de entonación es el parámetro prosódico con una mayor presencia, distinguiéndose entre métodos cuantitativos (como TILT [7], Fujisaki [8] o Bezier [9]) y métodos cualitativos (ToBI [10] o Intsint [11]). Para el modelado de la duración segmental se han utilizado métodos basados en reglas [12], o métodos estadísticos tales como redes neuronales [13] o árboles de clasificación y regresión (CART) [14]. El modelado de la intensidad es el menos presente en la literatura aunque hay algunos trabajos específicos en esta dirección tales como [15]. En nuestro trabajo previo [16], se confirmó la importancia de este parámetro en el modelado del habla para ciertas emociones.

En el habla natural, la duración de los sonidos depende del contexto en que se encuentran. La mayoría de estudios (por ejemplo [17] [18] [19]), utilizan el fonema como unidad básica para la duración, aunque haya aproximaciones basadas en difonemas o sílabas. Según estos estudios, los factores que influyen en la duración de los sonidos se deben básicamente a: *i)* la identidad del fonema y los de su contexto (habitualmente, el anterior y el posterior), *ii)* información sobre el acento, *iii)* información sobre la posición del fonema en la frase y en la sílaba. Cada estudio presenta su manera particular de codificar esta información.

La predicción de la curva de intensidad se suele llevar a cabo generalmente a nivel de fonema. Aunque muchos sistemas CTH no consideran este rasgo, los factores a tener en cuenta [6] están también relacionados con la identidad del fonema, el acento y la posición.

En [9] se realiza un estudio de diversos trabajos a cerca de las unidades de entonación y los factores que caracterizan cada una de estas unidades. Existen diferentes tipos de unidades utilizadas para modelar el contorno de entonación: las unidades inferiores a la sílaba y la sílaba (microentonación), el grupo acentual (GA) —relacionado con el ritmo del habla—, el grupo de entonación (GE) y otras unidades superiores (planificación del discurso). Dicho autor propone el uso del GA como unidad básica para el modelado de la entonación. Además, se concluye que algunos de los factores a considerar a nivel de grupo acentual para modelar la entonación estan relacionados con: *i)* el tipo de GE al que pertenece el GA, *ii)* la posición del GA dentro del GE, *iii)* la posición del acento, *iv)* la posición del GE dentro de la frase, y *v)* el número de sílabas del GA y del GE. La curva de entonación de cada GA se puede modelar con diferentes funciones (polinomios, Bezier [9], logarítmicas).

3. CBR APLICADO AL MODELADO DE LA PROSODIA

El aprendizaje artificial (ML) comprende un conjunto de técnicas que permiten reconocer una situación problemática y reaccionar utilizando la estrategia aprendida para un nuevo problema. La utilización de ML puede ser interesante en aquellos dominios en que la experiencia es escasa y la codificación del conocimiento que la describe es limitada o fragmentaria y por lo tanto incompleta. La predicción de los rasgos prosódicos a partir del texto es una tarea compleja en la cual intervienen muchos elementos (lingüísticos, fonéticos, pragmáticos). La utilización de técnicas de ML para dicha tarea puede deparar resultados válidos dentro del ámbito de los sistemas CTH. El proceso de ML se puede ver como la suma de dos fases: selección más adaptación. En una primera fase el sistema escoje (selecciona) las características más relevantes de un objeto, las compara con otras de conocidas a través de algun sistema de comparación, y cuando las diferencias son significativas, adapta el modelo de aquel objeto según el resultado de la comparación. Según sean estas dos fases los sistemas de ML se clasifican en Aprendizaje Analógico, Inductivo, Evolutivo y Conexionista. Las técnicas más utilizadas en el modelado prosódico pertenecen al aprendizaje inductivo, tales como árboles de decisión [14], y al aprendizaje conexionista, tales como Redes Neuronales [13].

En este trabajo, presentamos una aplicación del Razonamiento basado en casos (CBR) al modelado de la prosodia, ya que esta técnica de aprendizaje analógico permite un tratamiento sencillo de atributos de diferente naturaleza. El ciclo principal del CBR puede descomponerse en cuatro tareas (Ciclo 4R): recuperar los casos mas similares (*retrieve*), reutilizarlos para resolver el problema (*reuse*), revisar la solución propuesta (*revise*) y aprender de la experiencia (*retain*) [20] (figura 1). A continuación, se detalla la adaptación de estos pasos al modelado prosódico basado en corpus.

La inicialización del sistema no es propiamente una tarea del ciclo 4R del CBR pero será imprescindible para conseguir una memoria de casos que sea un equilibrio entre representatividad y compactación. En primer lugar hay que identificar los atributos que definen los casos para cada uno de las tres tareas del sistema: predicción de la duración y de la energía de los fonemas y de la curva de entonación de los GAs. Estos atributos y su naturaleza son los que se muestran en el apartado 4. Posteriormente, se genera el conjunto de entrenamiento mediante la unión de los rasgos prosódicos del corpus con los atributos prosódicos extraídos del análisis lingüístico del texto. La reducción de casos se consigue mediante el agrupamiento de casos representados por los mismos atributos. En el caso de reducir a un único caso cada grupo, el parámetro prosódico será la media de todos los casos del grupo. En cambio, si un conjunto de casos con los mismos atributos se divide en k subconjuntos, se hace necesario un proce-

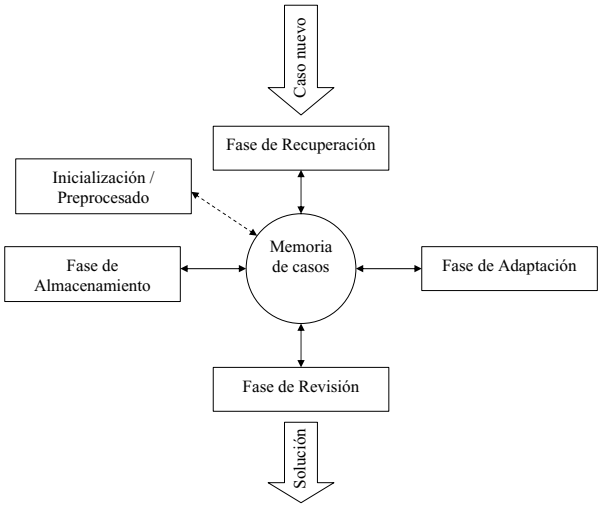


Figura 1. *Ciclo 4R del CBR.*

dimiento para seleccionar el mejor candidato en función del contexto.

El objetivo de la tarea de recuperación es mapear la solución desde la memoria de casos al nuevo problema. Se recupera el caso (o los *k* casos) más similar utilizando una métrica adecuada a los atributos que lo representan.

La tarea de reutilización trata de solventar un nuevo caso a partir de la información almacenada en la memoria de casos. En primer lugar se predice la duración de los fonemas, ya que la recuperación de la curva de F0 se hace sobre un eje temporal normalizado (ver figura 2). De esta forma, una vez conocida la duración de cada fonema se expande el eje temporal y se asocia el valor de F0 de cada fonema según el polinomio recuperado. En el caso de recuperar más de una muestra ($k > 1$) hay que realizar la adaptación para obtener únicamente una solución. Para el caso de la entonación, proponemos solventar este proceso decisional mediante la búsqueda de un camino óptimo que maximice la continuidad de F0 entre GAs.

En esta implementación, no se requiere revisión de la solución propuesta. El almacenamiento se realiza únicamente en la fase de inicialización. Por lo tanto el sistema no tiene la posibilidad de añadir nuevos casos en la fase de explotación.

4. ENFOQUE

La extracción de atributos prosódicos a partir del texto se realiza de forma automática mediante nuestra herramienta de análisis lingüístico que proporciona la transcripción fonética del texto y lo marca en grupos de entonación (GE), grupos acentuales (GA), palabras y sílabas. El GE se define como una estructura coherente de entonación que no incluye ninguna ruptura prosódica importante. Las rupturas prosódicas se producen debido a las pausas o inflexiones significativas del contorno de F0. Por el momento, únicamente consideramos las rupturas definidas por los signos de puntuación. El GA se define como una palabra acentuada precedida, si es el caso, por una o más palabras no acentuadas.

Para su implementación práctica, finalmente y despues de provar diferentes configuraciones, se han utilizado los atributos mostrados en la tabla 1 para la duración, intensidad y F0 respectivamente. Esta tabla muestra la etiqueta utilizada, una breve descripción y el tipo de atributo[1].

Para el modelado de la duración segmental y de la intensidad, hemos escogido el fonema como unidad acústica básica. La duración de un fonema depende principalmente de su identidad y del contexto donde se encuentra. Para la intensidad se han utilizado atributos parecidos (ver tabla 1).

Tabla 1. *Características prosódicas para duración, energía y F0.*

Etiqueta	Atributo	Tipo
F0	Fonema anterior	D
F1	Fonema actual	D
F2	Fonema siguiente	D
ACENTUADO	Fonema acentuado	B
GA-en-GE	Posición de GA en GE	D
FON-en-GE	Posición de FON en GE	D
DURACION	Duración del fonema en ms	N
Etiqueta	Atributo	Tipo
F1	Fonema actual	D
ACENTUADO	Fonema acentuado	B
GA-en-GE	Posición de GA en GE	D
FON-en-GA	Posición de FON en GA	D
FON-en-GE	Posición de FON en GE	D
INTENSIDAD	Intensidad *rms*	N
Etiqueta	Atributo	Tipo
TIPO-GE:	Tipo of GE	D
GA-en-GE	Posición de GA en GE	D
ACENTO	Posición de la sílaba tónica	D
GA-en-FRA	Posición del GA en la frase	D
NUM-SIL	Número de sílabas del GA	N
$a_0, a_1, \ldots a_n$	Coeficientes del polinomio	A

Para el modelado de la curva de F0 hemos elegido el GA como unidad básica siguiendo la propuesta de Escudero en [9]. El GA incorpora la influencia de la sílaba (cada GA está compuesto de una sílaba tónica y el resto átonas) y la estructura a nivel de GE se consigue mediante la concatenación de GAs. Por contra, este modelo carece de variaciones debidas a microentonación. Por el momento, sólo diferenciamos entre GEs enunciativos, interrogativos o exclamativos, que son fácilmente detectables a partir de los signos de puntuación. El atributo ACENTO indica la posición de la la sílaba tónica en el GA y el número de sílabas está relacionado con la longitud del GA (ver Tabla 1).

[1](D) Discreto, (B) Binario, (N) Numérico, (A) Array numérico

Para la curva de entonación de cada GA se ha utilizado una representación cuantitativa mediante los coeficientes de un polinomio aproximador de grado n (ecuación 1). Para encontrar los coeficientes del polinomio se parte de una colección de puntos (x_i, y_i) que representan el valor de la F0 media de cada fonema. Este valor de F0 media está referenciado al instante central del fonema. Mediante el método de mínimos cuadrados se calcula el polinomio aproximador de grado n que minimiza el error dado en la fórmula 2.

$$P(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \qquad (1)$$

$$E = \sum_{i=0}^{m} \left(P(x_i) - y_i\right)^2 \qquad (2)$$

El eje temporal se normaliza entre 0 y 1 para todos los GAs, de forma que el instante 0 representa el inicio del primer fonema del GA y el instante 1 el final del último fonema del GA. Se ha estudiado también la posibilidad de incluir en el cálculo de los coeficientes del polinomio información contextual, es decir incluir los puntos correspondientes a los dos fonemas adyacentes de los GAs vecinos (ver figura 2).

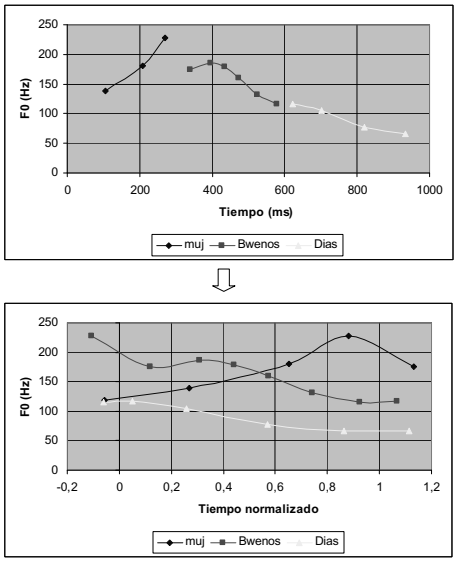


Figura 2. *Proceso de normalización del eje temporal de la curva de F0.*

5. EXPERIMENTOS Y RESULTADOS

5.1. Corpus

Para la realización de los experimentos hemos utilizado un corpus de habla desarrollado conjuntamente con el LAICOM del Departamento de Publicidad y Comunicación de la Universidad Autónoma de Barcelona. Un conjunto de textos extraídos de una base de datos de publicidad se han grabado por una locutora profesional en nuestro estudio de grabación. Los textos ya estaban clasificados por temáticas y se ha asociado un estilo expresivo a cada categoría. Los tres estilos iniciales son el neutro, el alegre y el sensual, que son los que se han realizado los experimentos. Recientemente, el corpus se ha ampliado con dos estilos más, agresivo y triste, que están en fase de segmentación y etiquetado.

5.2. Evaluación objetiva

El funcionamiento del sistema se ha evaluado mediante técnicas objetivas utilizando el error cuadrático medio (RMSE) y el coeficiente de correlación de Pearson (ρ). Los corpus de voz se han dividido en un 75 % para entrenar el sistema y un 25 % para evaluación. El RMSE mide la diferencia entre los valores observados y los predecidos en términos cuadráticos. El coeficiente de correlación mide el grado de dependencia lineal entre dos muestras de datos. Se calculan dichas medidas sobre las frases que forman el conjunto de test.

Los resultados para los tres estilos (neutro, alegre, sensual) se muestran en la Tabla 2. A nivel de entonación, el estilo sensual es el que presenta un mejor resultado debido a que es el estilo que presenta menos inflexiones. En cambio, presenta peores resultados en cuanto a la duración debido a que presenta segmentos de voz aspirada que tienden a alargarse. El estilo alegre es el que presenta mayores variaciones en la entonación, siendo el que obtiene un error mayor.

La comparación de resultados con otros trabajos se hace difícil al utilizar un corpus diferente. De todas formas, los resultados obtenidos son parecidos a los publicados por otros autores, la cual cosa nos permite considerar que es un buen punto de partida para el modelado de la prosodia. Existe un amplio margen para el ajuste y perfeccionamiento en ciertas tareas del modelado prosódico hecho hasta el momento, que seguramente permitirán mejorar los resultados finales.

5.3. Evaluación subjetiva

El experimento subjetivo se ha realizado mediante un test de percepción que pretende comparar la similitud entre la prosodia natural y la sintética. Los sujetos tienen que juzgar el grado de parecido (5= Muy alto, 4= Alto, 3= Cierto parecido, 2=Poco, 1=Nada) en la prosodia de dos versiones de la misma frase. Se indica a los usuarios que se fijen principalmente en la similitud de la prosodia. La frase con la prosodia natural se genera mediante resíntesis con los valores de F0 media, intensidad y duración etiquetados en el corpus. La otra frase se genera mediante resíntesis de la frase a partir de los valores de prosodia estimados. Para la modificación de la señal utilizamos nuestro sintetizador de voz basado en selección de unidades que realiza un ajuste de F0, duración y intensidad mediante TD-PSOLA [21]. Ambas versiones de ca-

Tabla 2. *Resultados del experimento objetivo*

	Entonación (F0)		Duración		Intensidad			
Estilo	RMSE (Hz)	ρ	RMSE (msec)	ρ	RMSE (rms)	ρ	Frases de Test	GA por frase
Neutro	29,668	0,631	21,688	0,746	0,021	0,853	169	5,41
Alegre	73,157	0,520	23,698	0,635	0,024	0,810	124	5,33
Sensual	23,534	0,438	28,936	0,655	0,029	0,673	101	4,76

da frase se presentaron via interfaz web a 10 sujetos para su evaluación. Las frases se podían escuchar las veces que fuera necesario.

En la tabla 3 se muestra el porcentaje de similitud que han obtenido los 3 estilos. Destaca la correlación entre estos resultados y el RMSE obtenido para la entonación. En la figura 3 se muestra la puntuación por cada frase y cada estilo.

Tabla 3. *Porcentaje de similitud entre prosodia natural y sintética para los tres estilos*

	Neutro	Sensual	Alegre
Muy Alto	5,00 %	42,31 %	5,56 %
Alto	50,72 %	30,00 %	44,44 %
Cierto parecido	30,71 %	22,31 %	29,86 %
Poco	13,57 %	5,38 %	17,36 %
Nada	0,00 %	0,00 %	2,78 %

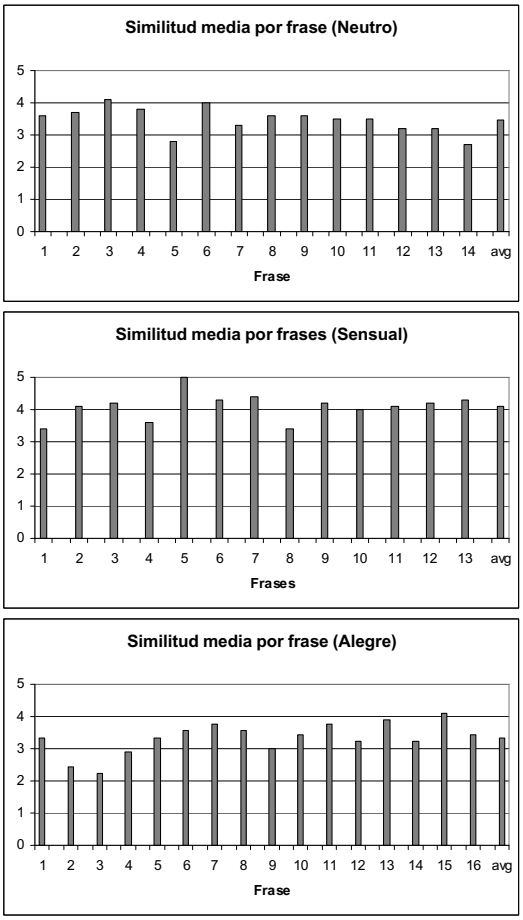


Figura 3. *Similitud media por frase y estilo (5=Muy Alta, 1=Nada).*

6. CONCLUSIÓN Y TRABAJO FUTURO

Este artículo presenta un nuevo enfoque para predecir la parámetros prosódicos desde texto en diferentes estilos expresivos. Se trata de la adaptación del CBR para el modelado cuantitativo de la prosodia (curva de frecuencia fundamental (F0), la duración segmental y la intensidad) para su uso en la conversión del texto en habla. Se ha llevado a cabo una evaluación objetiva y subjetiva para considerar su uso en el ámbito de la síntesis del habla expresiva. Con dicho fin se ha creado un corpus oral expresivo a partir de textos publicitarios en tres estilos (neutro, alegre y sensual) que se está ampliando los estilos agresivo y triste.

Como trabajo futuro, se pretende comparar esta técnica con otras técnicas de ML aplicadas sobre el mismo corpus, ampliar el estudio a los dos estilos nuevos y explorar el uso de nuevos atributos prosódicos y su relación con cada estilo.

7. BIBLIOGRAFÍA

[1] R.Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, y J. G. Taylor, "Emotion recognition in human computer interaction," *IEEE Signal Processing*, vol. 18, no. 1, pp. 33–80, January 2001.

[2] J. E. Cahn, "Generating expression in synthesized speech," M.S. thesis, Massachusetts Institute of Technology, 1989.

[3] Montero J.M., J. Gutiérrez Arriola, J. Colás, E. Enríquez, y J.M. Pardo, "Analysis and modelling of emotional speech in Spanish," in *Proceedings of 14th International Conference of Phonetic Sciences*, San Francisco, USA, 1999, pp. 957–960.

[4] M. Schröder, "Experimental study of affect bursts," *Speech Communication*, vol. 40, pp. 99–116, 2003.

[5] E. Eide, A. Aaron, y Cohen P. Donovan R. Hamza W. Mathes T. Picheny M. Polkosky M. Smith M. Viswanathan M Bakis, R., "Recent improvements

to the ibm trainable speech synthesis system," in *Proceedings of ICASSP'03*, Hong Kong, 2003.

[6] J. Llisterri, M. J. Machuca, C.de la Mota, M. Riera, y A. Ríos, *Entonación y tecnologías del habla*, Tecnologías del texto y del habla. Prieto, P. (Ed.) Teorías de la entonación. Ariel (Lingüística), Barcelona, 2003.

[7] Paul Taylor, "Analysis and Synthesis of Intonation using the Tilt Model," *Journal of Acoustical Society of America*, 2000.

[8] H. Fujisaki, S. Ohno, K.Ñakamura, M. Guirao, y J. Gurlekian, "Analysis of accent and intonation in Spanish based on a quantitative model," in *Proc. ICSLP*, 1994.

[9] D. Escudero, *Modelado Estadístico de Entonación con Funciones de Bézier: Aplicaciones a la Conversion Texto-Voz en Español*, Ph.D. thesis, Universidad de Valladolid, 2003.

[10] K. Silverman, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, y J. Hirschberg, "ToBI: A standard for labelling English prosody," in *Proceedings of ICSLP92*, 1992.

[11] D.J. Hirst, N. Ide, y Veronis J., "Coding fundamental frequency patterns for multilingual synthesis with INTSINT in the MULTEXT project," in *2nd ESCA/IEEE Workshop on Intonation*, 1994.

[12] D.H. Klatt, *Synthesis by rule of segmental durations in english sentences*, B. Lindblom and S. Öhman (Ed.), Frontiers of Speech Communication. New York: Academic, 1979.

[13] N.W. Campbell, "Analog I/O nets for syllable timing," *Speech Communication*, vol. 9, pp. 56–61, 1990.

[14] B. Möbius y J. van Santen, "Modelling segmental duration in German TTS synthesis," in *Proc. of ICSLP*, 1996.

[15] J. Trouvain, W. J. Barry, C.Ñielsen, y O. Andersen, "Implications of energy declinations for speech synthesis," in *3rd ESCA/ COCOSDA Workshop on Speech Synthesis, November*, Jenolan Caves, Australia, 1998, pp. 47–52.

[16] I. Iriondo, R. Guaus, A. Rodríguez, P. Lázaro, N. Montoya, J. Blanco, D. Bernadas, J. Oliver, D. Tena, y L. Longhi, "Validation of an acoustical modelling of emotional expression in Spanish using speech synthesis techniques," *Proc. of the I. W. on Speech and Emotion*, pp. 161–166, Sept. 2000.

[17] A. Febrer, J. Padrell, y A. Bonafonte, "Modeling Phone Duration: Application to Catalan TTS," in *3rd Int. Workshop on Speech Synthesis*, Jenolan Caves, Australia, 1998.

[18] Navas E., Hernáez I., y Sánchez J.M., "Modelo de duración para conversión texto a voz en euskera," *Procesamiento del Llenguaje Natural*, vol. 1, no. 3, 2002.

[19] J. P. Teixeira y D. Freitas, *Evaluation of a Segmental Durations Model for TTS*, Computational Processing of the Portuguese Language - 6th International Workshop, PROPOR 2003. N. Mamede, J. Baptista, I. Trancoso, M.G. Nunes (Eds), Edited by Springer, 2003.

[20] A. Aamodt y E. Plaza, "Case-based reasoning: foundational issues, methodological variations, and system approaches," *AI Commun.*, vol. 7, no. 1, pp. 39–59, 1994.

[21] E. Moulines y F. Charpentier, "Pitch-synchronous waveform processing techniques for TTS synthesis using diphones," *Speech Communication*, vol. 9, pp. 453–467, 1990.

SPOKEN DOCUMENT RETRIEVAL FOR A NATIONAL GALLERY OF THE SPOKEN WORD

John H. L. Hansen

Center for Robust Speech Systems (CRSS)
Department of Electrical Engineering
Erik Jonsson School of Engineering and Computer Science
University of Texas at Dallas, U.S.A.
http://crss.utdallas.edu/

RESUMEN

The problem of reliable speech recognition for information retrieval is a challenging problem when data is recorded across different media and equipment. In this talk, we address the problem of audio stream phrase recognition for a new National Gallery of the Spoken Word (NGSW]. This will be the first large-scale repository of its kind, consisting of speeches, news broadcasts, and recordings that are of significant historical content. An NSF initiative was recently established to provide better transition of library services to digital format. As part of this Phase-II Digital Libraries Initiative, researchers from Michigan State Univ. (MSU) and Center for Robust Speech Systems (Univ. of Texas, Dallas) have teamed to establish a fully searchable, online WWW database of spoken word collections that span the 20th Century. The database draws primarily from holdings of MSU's Vincent Voice Library which include +60,000 hours of recordings (from T.Edison's first cylinder recordings, to famous speeches such as man's first steps on the moon "One Small Step for Man", to American presidents over the past 100 years). In this partnership, MSU will house the NGSW collection, as well as digitize (with assistance from LDC), catalog, organize, and provide meta-tagging information. MSU is also responsible for a number of engineering challenges such as digital watermarking and effective compression strategies. The team at CRSS-UTD is responsible for developing the robust automatic speech recognition and segmentation for transcript generation, and for the proto-type online audio/metadata/transcript based user search engine (http://SpeechFind.utdallas.edu). Additional challenges for SpeechFind include dialect/accent tagging, speaker tracking, cross-language SDR applications, and environmental sniffing concepts.

Biografía: John H. L. Hansen

John Hansen received the PhD and MSEE from Georgia Institute of Technology, and BSEE from Rutgers University, all in Electrical Engineering. He is Department Chair and Professor in Dept. of Electrical Engineering, Erik Jonsson School of Engineering and Computer Science, and Professor in the School of Brain and Behavioral Sciences (Speech & Hearing) at the University of Texas at Dallas (UTD). He also holds the Distinguished Chair in Telecommunications Engineering at UTD. From 1999-2005, he was Professor in the Dept. of Speech, Language and Hearing Sciences (SLHS), and Dept. of Electrical and Computer Engineering (ECE) at the Univ. of Colorado. He also served as Dept. Chair of SLHS. He was co-founder of the Center for Spoken Language Research, and coordinator of the Robust Speech Processing Group at CSLR. He was a faculty member at Duke Univ., in the Dept. of Electrical Engineering and Dept. of Biomedical Engineering from 1988-1999. His research interests are focused on speech analysis and modeling, robust automatic recognition, and language technology. He previously served as the Technical Advisor to the U.S. Delegate for NATO, and served as general chairman/organizer of ICSLP-2002, Inter. Conf. on Spoken Language Processing, Denver, CO. He presently serves as IEEE Distinguished Lecturer (2005/06), and previously served as Associate Editor for IEEE Trans. Speech & Audio Processing (1993-1999), Associate Editor for IEEE Signal Processing Letters (1999-2000), Editorial Board for IEEE Signal Processing Magazine (2000-2002), IEEE Student Branch Coordinator (1988-99). He has supervised 33 PhD and MS thesis students over the past 17 years, and has authored/co-authored 222 journal papers, conference papers, textbooks in the field of speech processing and language technology.

A PROPOSAL OF METRICS FOR DETAILED EVALUATION IN PRONUNCIATION MODELING

R. Barra, J. Macías-Guarasa, F. Fernández, L.F. D'Haro, J.M. Montero and J. Ferreiros

Grupo de Tecnología del Habla. Depart. de Ingeniería Electrónica. Universidad Politécnica de Madrid
E.T.S.I. Telecomunicación. Ciudad Universitaria s/n, 28040-Madrid, Spain
{barra, macias, efhes, lfdharo, juancho, jfl}@die.upm.es

ABSTRACT

In the context of ASR systems it is of major importance to accurately model the allophonic variations to be faced in a real world task.

The evaluation of which pronunciation variants are actually improving the system performance is crucial, as it determines the acceptance of the pronunciation alternatives used. Traditional approaches use different criteria and, typically, evaluation only cares about the global impact of the augmented dictionaries in the WER, so that this leads to little further insight on till what extent the proposed variations are actually working or not. Our proposal in this paper is also evaluating the *effective improvement* due to every pronunciation variation used, defining specific improvement metrics on the utterance level. We will show how these metrics actually highlight the beneficial impact achieved by the application of phonological rules when dealing with certain pronunciations variants, while the differences observed in global WER are not statistically significant.

1. INTRODUCTION

In the literature there are plenty of references to the problem of introducing pronunciation variants in speech recognition systems (an excellent revision can be found in [1]). Explicit modeling of the pronunciation alternatives is able to achieve substantial improvements if the acoustic models closely match the transcriptions [2]. The most usual strategies to add pronunciation variants use either knowledge based approaches, applying phonological rules to the canonical dictionaries, or data-derived pronunciation variants. The research teams are extremely careful when adding variants and several approaches to generate and limit their number have been proposed, such as using a ML criterion [3], smoothing the automatically derived phonetic transcriptions [4] or measuring the occurrences of variants [5], to name a few.

All of them try to evaluate till what extent the added variants are actually achieving better results or not. Unfortunately, traditional evaluation takes only into account the effectiveness of a given method in improving the overall WER. Detailed error analysis or alternative evaluation metrics do not seem to be a priority in order to provide insight into the processes underlying pronunciation variation (due to a number of practical reasons) [1].

In this scenario, when the overall WER evaluation leads to results that are not statistically significant, it is very difficult to decide whether the given pronunciation variants are effective or not, so that they may be easily thrown away without actually knowing whether their impact is relevant for a given subset of the database (for example, we may think of a minority set of speakers coming from an specific dialectal area).

In this paper, we propose a set of metrics and a methodology for its application, that can give additional insight on the detailed improvements that may be achieved by the application of phonological rules, when the differences observed in global WER are not statistically significant.

2. EXPERIMENTAL SETUP

In this paper, we will work in both isolated (IWR) and continuous speech recognition (CSR) systems using the hypothesis-verification paradigm, where we will consider pronunciation alternatives at the segmental level (more specifically, within word variations generated using manually-derived rules).

2.1. Speech recognition systems

In the IWR task [7], the hypothesis module follows a bottom-up approach in which a phonetic string build up algorithm (using CI semicontinuos HMMs) is followed by a lexical access stage. The verification module is based on the Viterbi algorithm, using context-dependent HMMs. The latter receives a list of *preselected* words (sorted according to their likelihoods) generated by the hypothesis module and generates the final recognition result.

In the CSR system [8], the hypothesis module uses an integrated search approach combining CD continuous

HMMS and bigram LM. It generates a word graph to be further rescored by the verification module. In the current version, the verification module uses additional information stored in a trigram language model.

In both the IWR and CSR tasks, we decided to use canonical pronunciations while training the acoustic models, as we want to assume the worst-case scenario regarding the trained HMMs.

2.2. Databases and dictionaries

In our IWR experiments, we have used a subset of the VESTEL database [9], composed of 9790 utterances. VESTEL is a realistic speaker-independent speech corpus collected over commercial telephone lines. Cross-validation is applied by means of a *leave-10%-out* strategy, in order to increase the statistical significance of the results. The *canonical dictionary* is composed of 1952 words, with a single phonetic transcription per word.

In our CSR experiments, we have used a subset of the INVOCA database that was designed to support research and development in spontaneous speech recognition systems in air traffic control tasks. INVOCA contains spontaneous conversations between air traffic controllers and airplane pilots in the Madrid-Barajas (MAD) airport [8]. Our results will be based on the evaluation of the *clearances subset*, composed of 8.9 hours of recorded conversations (5011 utterances), using 8 hours (4588 utterances) for training and 0.9 hours (503 utterances) for testing, with a *canonical dictionary* composed of 994 words (single phonetic transcription per word). The word graph is generated by an *n*-best search strategy during decoding (*n* between 10 and 20).

2.3. Rule selection process

From internal studies carried out in our Group, we generated an exhaustive repertoire of Castilian Spanish pronunciation variants, in order to be included in speech recognition systems. In this work, we further reduced this repertoire, leading to a selection of up to 13 phonological rules. The main selection criterion was to keep the rules commonly accepted as 'typical' for the average Spanish speaker. Nevertheless, our target is proposing and evaluating a set of performance metrics, so that the actual set of rules used is not that important. Rules are applied to the *canonical* dictionary in order to generate the *modified* one.

3. EVALUATION STRATEGY

In our proposal, the evaluation of pronunciation alternatives takes two factors into account:

- The (traditional) overall impact in WER, measured on the whole test set. It corresponds to a global, average performance, which we do not want to increase. However, we could tolerate a certain minimum degradation (i.e., statistically not significant), if this degradation leads to a benefit in some aspect that we may consider relevant.
- The (proposed) *effective improvement* of the considered rules, measured on the subset of the whole test set for which there exist differences between using or not the pronunciation alternatives (we call this improvement "*effective"* in the sense that it shows the actual detailed impact of the rules). This effect could seem irrelevant due to its limited impact in the overall performance (as pronunciation variants rarely have a big impact in global WER), but it is justified as it allows real-world systems to correctly recognize a certain set of speakers or pronunciations that, otherwise, would be poorly handled by the speech recognizer: in publicly deployed speech recognition systems, it is of outmost importance to reduce the number of speakers for which the system would not work at all (and allowing for a statistically not significant degradation in WER).

Our idea for designing new evaluation metrics has a common ground both in the IWR and CSR tasks: generating a specific *per-utterance performance metric* and integrating all the individual calculations in several quality metrics.

3.1. Evaluating *effective* improvements in the IWR task

The general idea in the IWR case is calculating in which position within the (sorted) list of recognized words, the correct word was actually recognized. So, the basic *per-utterance performance metric* would be the "rank" of the given utterance 'R': R_c when using the canonical dictionary and R_m when using the modified one. After making this calculation for both the canonical and modified dictionaries, we can compare the results and decide, for every utterance, which dictionary got the best result (the one with the lower R). The higher the differences between R_c and R_m, the higher the impact of the phonological rule applied.

We have evaluated a full set of up to 19 evaluation metrics, for the sake of brevity, we will just show a sample subset of them here:

- Number of utterances for which there was no difference between both dictionaries ($R_c = R_m$)
- For the subset utterances in which using the modified dictionary lead to better results ($R_c > R_m$). We will refer to this subset as 'improve' set:
  - Number of utterances
  - Average relative difference between R_m and R_c (measured as a percentage relative to R_c)

- For the subset of utterances in which using the modified dictionary lead to worse results ($R_c < R_m$). We will refer to this subset as 'worsen' set:
  - Number of utterances
  - Average relative difference (%) between R_c and R_m

Each of the proposed metrics (the ones shown here and the rest belonging to the full set of 19) gives more insight on specific aspects of the impact of the added variants (overall impact, absolute or relative improvement gain, details on canonical or modified dictionaries improvements and preferences, etc.) that would be impossible to evaluate by just looking at global WER.

3.2. Evaluating *effective* improvements in the CSR task

In the CSR task, the basic *per-utterance performance metric* has a natural candidate, the *per-utterance WER*, as proposed in [10], given its considerable improvement in informativeness and adequacy for statistical testing. When talking about *per-utterance WER*, we will refer to WER_c when using the canonical dictionary and WER_m when using the modified one.

From the full list of 15 evaluation metrics we have considered in [7] we just show a sample subset of them here:

- Number of utterances for which there was no WER difference between both dictionaries ($WER_c = WER_m$)
- For the 'improve' set: the subset of utterances in which the WER using the modified dictionary is better than when using the canonical one ($WER_c > WER_m$):
  - Number of utterances
  - Absolute and relative improvement of the average WER
- For the 'worsen' set: the subset of utterances in which the WER using the modified dictionary is worse than when using the canonical one ($WER_c < WER_m$):
  - Number of utterances
  - Absolute and relative improvement of the average WER

As in the IWR case, those metrics give us valuable detailed information, complementary to the global WER evaluation.

4. EXPERIMENTAL RESULTS

In order to validate our proposal and show how these metrics can be actually used and interpreted, we will give details on general evaluation metrics (search space increase and overall impact in error rate) and the proposed ones. The experimental procedure includes, as an example, the evaluation of a sample individual rule and the evolution of the results when applying the full set of rules, incrementally selected by means of the application of a *greedy* search algorithm (in which the search criteria is based on maximizing the error rate improvement for every addition of a new rule).

4.1. Evaluation in the VESTEL IWR task

The relative increase in dictionary size (compared to the canonical one) rises over 130% if we apply the full set of rules simultaneously. Relative increases when applying single rules vary from 0.14% and 37.60%. As an example, when applying the *dfinal* rule, the relative increase is around 2%.

When evaluating the overall impact in performance using any set of rules, we found that, while slightly increasing the error rate, the differences were not statistically significant. For example, when using the 13 phonological rules, the relative increase in inclusion error rate is below 0.06% (which is negligible, especially if we take into account that the number of entries in the dictionary has increased 130%). When only applying, e.g., the *dfinal* rule, we found a minor improvement in inclusion error rate: 1.38% relative.

In table 1 we show, as an example, the results for the application of the *dfinal* rule in the VESTEL IWR task.

Table 1: Effective improvement/worsening results when applying the *dfinal* rule (IWR task)

Metric	Value
Number of utterances for which *canonical = modified* ($R_c = R_m$)	9393 (96.6%)
Number of utterances for which *modified* is worse ($R_c < R_m$)	263 (2.7%)
Number of utterances for which *modified* is better ($R_c > R_m$)	64 (0.7%)
Avg. rel. worsening (R_c-R_m)/ R_m when *modified* is worse	**8.36%**
Avg. rel. improvement (R_m-R_c)/ R_c when *modified* is better	**70.25%**

The first important observation is that the number of words adversely affected by the added variants is higher than the number of benefited ones (263 vs. 64). From these figures, it is clear that the application of that rule will never have a statistically significant impact in overall performance and, if any, it will probably be negative. However, if we then focus on the average differences between R's, we can see that the improvement obtained with the modified dictionary is significantly larger than the improvements obtained with the canonical dictionary: every adversely affected word, in average, loses a 8.36% in relative position within the list of recognized words, while the benefited ones gain 70.25%, also in average. These figures show that the effective negative impact of

the application of the rule is much smaller than its effective positive impact in the words for which the recognition results are actually improved, with similar quantitative results for other rules.

Additional evaluation metrics show that the negative impact takes place for words recognized in high (bad) positions within the preselection list ($R_c \geq 100$), while positive impact takes place in low (good) positions ($R_m \approx 25$). As we are using a hypothesis-verification strategy and given the preselection list size, this means that, due to the introduction of pronunciation variants, the correct word will probably be included in the preselection list, thus having an opportunity to be correctly recognized by the verification module.

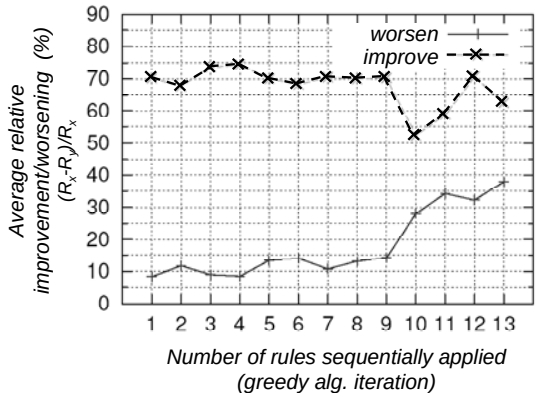


Figure 1: Average relative improvement/worsening in rank for the "worsen" and "improve" sets (IWR task)

Regarding the *greedy-based* incremental selection of optimal rules, Figure 1 shows the evolution of the average relative improvement in recognized position (rank) for the set of words benefited by the inclusion of pronunciation alternatives (*improve* in the figure) and the adversely affected ones (*worsen* in the figure), as a function of the number of rules incrementally applied (iterations of the *greedy* algorithm). We note that, for the first 9 best rules, the effective positive impact is much higher than the effective negative impact. This means that even though the number of utterances for which we can measure an improvement due to the use of pronunciation alternatives is low, the actual improvement obtained for such utterances is, again, much higher than the actual worsening due to the same alternatives. The main reason for such improvements is that the added pronunciation variants have a very strong positive effect in certain words that are poorly handled by the recognizer when using their canonical transcriptions, while leading to a minor impact in the rest of the words.

Plots such as the one shown in Figure 1 can give researchers a clue on how many rules should be used while achieving significant effective improvements due to the application of pronunciation alternatives (and provided that the overall impact in error rate is statistically not significant).

Table 2: Effective improvement/worsening results when applying the *dfinal* rule (CSR task)

Metric	Value
Number of utterances for which *canonical* = *modified* ($WER_c = WER_m$)	462 (91.9%)
Number of utterances for which *modified* is worse ($WER_c < WER_m$)	19 (3.8%)
Number of utterances for which *modified* is better ($WER_c > WER_m$)	22 (4.3%)
Avg. rel. WER worsening when *modified* is worse ($WER_c < WER_m$)	**31.46%**
Avg. rel. WER improvement when *modified* is better ($WER_c > WER_m$)	**53.33%**

4.2. Evaluation in the INVOCA CSR task

The relative increase in dictionary size if we apply the full set of rules rises over 173%. When applying single rules, relative increases vary from 0.56% to 32.24%. As an example, when applying the *dfinal* rule, the relative increase is 2.8%.

Again, when evaluating the overall impact measured as the increase in global WER, we obtained minor differences which were not statistically significant. For example, when using the 13 phonological rules, the relative increase in global WER is below 3% (negligible considering the 173% increment in number of entries). When only applying the, e.g., *dfinal* rule, the relative increase in global WER is below 0.3%. In Table 2 we show, as an example, the results for the application of the *dfinal* rule in the INVOCA CSR task.

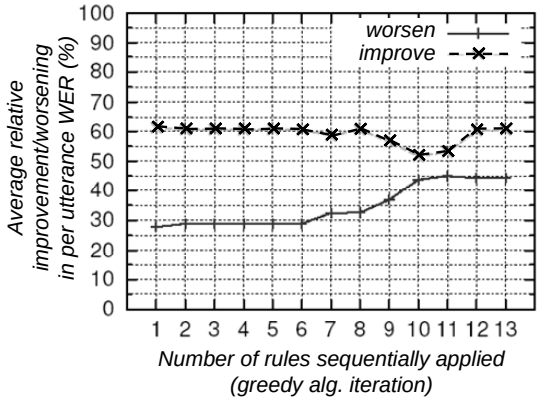


Figure 2: Avg. relative improvement/worsening in per-utterance WER (%) for the "worsen" and "improve" sets (CSR task)

In this case, the number of utterances adversely affected by the added variants is roughly the same than

the number of benefited ones (19 vs. 22), but, similarly to the IWR task, the average relative improvements due to pronunciation variants is higher than the performance loses (53.33% vs. 31.46%).

Figure 2 shows the evolution of the average relative improvement in WER (estimated from the word graph using an optimal search rescoring, and guided by the *greedy* algorithm searching for the optimal rules) for the *worsen* and *improve* set of utterances. Improvements due to pronunciation variants keep almost constant till the application of the 8th best rule and we could make similar considerations to the ones discussed in the IWR task: better inclusion results in the word graph lead to better results after rescoring.

4.3. Computational complexity considerations

As shown above, the increase in dictionary size can be high when using pronunciation variants. Our experiments show that, with the current systems using aggressive beam search techniques, the computational complexity increase due to increased dictionary size still allows for real time performance with the abovementioned results.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a proposal for the evaluation of pronunciation alternatives in speech recognition systems. It is aimed at complementing the traditional evaluation approach which only takes into account the impact in global error rate.

We have defined specific effective improvement metrics, especially adapted to get more insight on the actual impact of the pronunciation alternatives used, from different perspectives. All of the proposed metrics are based in the combination of a given *per-utterance quality metric* evaluated on the subsets for which differences between using or not pronunciation alternatives are found.

We have performed an experimental evaluation on both IWR and CSR tasks. Our results show that, even though the overall impact of the pronunciation alternatives is not statistically significant (even in dictionaries significantly bigger than the canonical ones), we can get effective benefits, important enough for certain users or pronunciations to have an opportunity to be correctly recognized. We show examples of the effect of including a single pronunciation rule and the incremental combination of all the selected rules by means of a *greedy* search algorithm. With the given metrics and the combination of evolution plots of these metrics, researchers can get additional information on till what extent individual rules are actually achieving effective improvements in the relevant database subsets.

To summarize, in our proposal the decision of including certain pronunciation variants is based on three considerations:

- The overall WER differences are not statistically significant
- The increase in computational load when using the variants is acceptable, given a certain hardware setup
- The relative improvements due to pronunciation variants is higher than the performance loses, according to the proposed metrics and plots

We are currently working in refining our methodology and specifically dealing with the evaluation of multiple pronunciation strategies using a data-driven approach.

6. ACKNOWLEDGEMENTS

This work has been partially supported under contracts TINA (UPM-CAM R05/10922), ROBINT (DPI2004-07908-C02) and EDECAN (TIN2005-08660-C04).

7. REFERENCES

[1] Strik, H. and Cucchiarini, C. “Modeling pronunciation variation for ASR: a survey of the literature”. Speech Communication, vol 29, p. 225-246. 1999.

[2] Saraçlar, M., Nock, H. and Khudanpur, S. “Pronunciation modeling by sharing Gaussian densities across phonetic models”. Computer Speech and Language, vol 14, p. 137-160. 2001.

[3] Holter. T. “Maximum Likelihood Modelling of Pronunciation in Automatic Speech Recognition”. PhD. Thesis. Norwegian University for S&T. 1998.

[4] Riley, M., Byrne, W et al. “Stochastic pronunciatio modeling from hand-labelled phonetic corpora”. Speech Communication, vol 29, p. 209-224. 1999.

[5] Kessens, J. and Wester, M. “Imrpving the performance of a Dutch CSR by modeling within-word and cross-word pronunciation variation”. Speech Communication, vol 29, p.193-207. 1999.

[6] Wester, M. "Pronunciation modeling for ASR – knowledge-based and data-derived methods". Computer Speech and Language, vol 17, p 69-85. 2003

[7] Macías-Guarasa, J., Ferreiros, J., Colás, J., Gallardo, A. and Pardo, J.M. “Improved Variable List Preselection List Length Estimation Using NNs in a Large Vocabulary Telephone Speech Recognition System”. ICLSP00, pp. 823-826. 2004

[8] Fernández, F., Córdoba, R., Ferreiros, J., Sama, V., D'Haro, L.F. and Macías-Guarasa, J. “Language Identification Techniques based on Full Recognition in an Air Traffic Control Task “ ICLSP04, pp. 1565-1568. 2004

[9] Tapias, D., Acero, A., Esteve, J., and Torrecilla, J.C. "The VESTEL Telephone Speech Database". ICSLP94, pp. 1811-1814. 1994.

[10] H. Strik, C. Cucchiarini, J.M. Kessens. “Comparing the performance of two CSRs: How to determine the significance level of the differences”. Eurospeech 2001, Vol. 3, pp. 2091-2094.

BILINGUAL SPEECH RECOGNITION IN TWO PHONETICALLY SIMILAR LANGUAGES

Vicente Alabau[*]*, Carlos D. Martínez*[*]

[*]Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València
Camí de Vera, s/n. 46071 València, Spain
{valabau,cmartine}@dsic.upv.es

ABSTRACT

As Speech Recognition Systems improve, they become suitable for facing new problems. Multilingual speech recognition is one of such problems. In the present work, the case of the *Comunitat Valenciana* multilingual environment is studied. The official languages in the *Comunitat Valenciana* (Spanish and Valencian) share most of their acoustic units, and their vocabularies and syntax are quite similar. They have influenced each other for many years. A small corpus on an Information System task was developed for experimentation purposes. This choice will make it possible to develop a working prototype in the future, and it is simple enough to build semi-automatic language models. The design of the acoustic corpus is discussed, showing that all combinations of accents have been studied (native, non-native speakers, male, female, etc.). In addition, some experiments have been conducted with this corpus that show promising results for a Spanish-Valencian multilingual speech recognizer.

1. INTRODUCTION

The quality of Automatic Speech Recognition (ASR) has improved greatly in recent years [1, 2, 3]. Some commercial products have appeared for real-world tasks, such as speech transcription systems in restricted domains and automatic call centres. However, some problems arise in these real-world tasks: recognition performance is low under adverse circumstances, and the models are very noise sensitive [4, 5].

In this paper, we design and acquire a corpus to research one of these problems: multilingual interoperability. The problem of multilingual interoperability presents several issues related to the components of a classical ASR system, like acoustic or language models.

With respect to acoustic models, ASR systems are very language-dependent, because the phone sets are different in each language. Moreover, coarticulation effects of the same phonemes may differ in each language, and even the articulation of a phoneme may have its own singularities. Some work, for example the introduction of contextual acoustic models (triphones), has already been done to find more robust acoustical units under these conditions [6].

Language models are also very language-dependent, because of their vocabulary and gramatical issues. Furthermore, vocabulary transcription is dialect-dependent as well. For example, Spanish utterances from South America and Spain differ in a noticeable way.

Language determination is also an important issue. In some tasks, the speaker's language is unknown. Thus, the system has to find the best way to determine which language it is. Moreover, when the system has to answer the speaker, the identification of the language is needed in order to be able to answer in the same language.

In multilingual environments, other difficulties are added to speech recognition, even in monolingual ASR. Languages are usually influenced by other languages that are present in the environment and by the speaker's mother tongue (e.g., the perception distortion of a non-native Dutch speaker is equivalent to a reduction of the signal-to-noise ratio of 3-4 dB for non-native Dutch speakers [7]). This interference is demonstrated by mispronunciation and the use of syntactical structures and vocabulary from the mother tongue. For anyone who has studied foreign languages, it is easy to understand that phonemes that are not present in the mother tongue are hard to pronounce. It is even possible to identify the nationality of some people by their accent. Some syntactical and vocabulary mistakes are produced by the lack of knowledge of the foreign language.

In this work, the case of the *Comunitat Valenciana* is studied. In *Comunitat Valenciana*, two official languages coexist: Spanish and Valencian. Valencian is the name for the Catalan language dialect that is spoken in the *Comunitat Valenciana*. Catalan is one of the most widely spoken minor languages in Europe. About 6.5 million people speak it actively (on a daily basis), and about 12 million people are potential speakers (they know the language but use Spanish on a daily basis). Furthermore,

Work supported by the "Agència Valenciana de Ciència i Tecnologia" under grant GRUPOS03/031, the Spanish project TIC2003-08681-C02-02 and the "Programa d'Incentiu a la Investigació 2004 UPV".

the Catalan government is making an important effort to promote the use of the Catalan language in all spheres. Therefore, there is great interest in the speech recognition technologies for Catalan.

As official languages, every citizen has the right to know and use both Spanish and Valencian in the *Comunitat Valenciana*. However, the repression of the use and learning of Valencian in the Franco period (1939-1977) (also called Catalan Negationism) and other historical reasons, have caused that currently only 85% of the population of *Comunitat Valenciana* understand Valencian, and only 48% are active speakers[8]. This has also caused the Valencian phone set to be reduced by the extensive use of Spanish, which is true even for Valencian native speakers (a situation which has not occurred in other Catalan dialects). Thus, nowadays the Valencian phone set differs very little from the Spanish phone set.

In the following sections, we describe the design of a multilingual corpus for Spanish and Valencian, and we summarize the most common multilingual approaches presented in the literature. We also present preliminar results on this corpus that show the performance of each aproach. Conclusions and future work are presented in the last section.

2. MULTILINGUAL CORPUS DESIGN

As stated above, the Valencian dialect has special phonetic features with respect to standard Catalan. Thus, although there are a few speech recognition resources for the Catalan language there was no resource for Valencian, and a Valencian language corpus had to be acquired. For this reason, we had to acquire a specific Valencian speech corpus and a similar Spanish one. Although Spanish speech corpora are available [9], it was important to have Spanish and Valencian corpora with the same features to be able to compare them more faithfully.

Thus, we decided to acquire our own multilingual corpus specifically for experimentation purposes (i.e., not for real system development). We planned to acquire a small, simple corpus and decided to design a set of 120 medium-length sentences (60 for each language) for 20 speakers, which corresponds to approximately 1 hour of speech per language (actually, the length of the recorded signal is about 2 hours). This amount of speech signal should be enough for the experimental purposes that the corpus is going to be used for.

We chose an Information System task to design the corpus. This was done because this task is complex enough for demostration purposes, and it is simple enough to semi-automatically generate the task sentences. As there are few syntactic differences between Spanish and Valencian (especially for this task), the semi-automatic sentences could be easily translated. Dictionary translation for single words and some minor modifications were sufficient to accomplish the translation task.

The goal of this Information System was to provide information about the staff of a department by phone [10]. The possible information items the system could be asked for included timetables, office hours, phone numbers, e-mail addresses, or office locations. Some example sentences are shown in Figure 1.

Spanish

- Por favor, quiero saber el e-mail de Álvaro Rodríguez, adiós.
- Buenas noches, quería la extensión de la señorita Silvia Abrahao, muchas gracias.
- Buenos días, ¿cuál es el horario de consultas del doctor Vicente?, gracias.

Valencian

- Per favor, vull saber l'e-mail d'Álvaro Rodríguez, adeu.
- Bona nit, volia saber l'extensió de la senyoreta Silvia Abrahao, moltes gràcies.
- Bon dia, quin és l'horari de consultes del doctor Vicente?, gràcies.

English

- Please, I want to know the e-mail of Álvaro Rodríguez, goodbye.
- Good evening, I wanted to know the extension of Miss Silvia Abrahao, thank you very much.
- Good morning, what are the office hours of the Dr. Vicente?, thanks.

Figure 1. This is a selection of sentences that are representative of the corpus. The English sentences are provided for a better understanding of the examples.

This task was tested in a previous work [10] with acoustic models that were designed for other tasks. This work showed promising results in bilingual Valencian-Spanish ASR and has encouraged us to continue research in this field.

3. LANGUAGE MODELLING

Language modelling is crucial in an ASR system. Language models define which kind of sentences are allowed in the system. Therefore, any sentence said by a speaker will not be recognized correctly if it does not belong to the language model. Indeed, this sentence will be recognized as the one that is closest to one that exists in the language model.

The language model of this corpus was designed to suit our experimentation needs. That is, it should be

block	greeting
Spanish	por favor, buenas noches, buenos días
Valencian	per favor, bona nit, bon dia
English	please, good evening, good morning
block	question
Spanish	quiero saber, quería saber, cuál es
Valencian	vull saber, volia saber, quin és
English	i want to know, i wanted to know, what is
block	information
Spanish	el e-mail, la extensión, el horario de consultas
Valencian	l'e-mail, l'extensió, l'horari de consultes
English	the e-mail, the extension, the office hours
block	title
Spanish	señorita, doctor
Valencian	señoreta, doctor
English	Miss, Dr.
block	person
All languages	Álvaro Rodríguez, Àlvar Rodríguez, Álvaro, Àlvar, Rodríguez, Silvia Abrahao, Silvia, Abrahao,Vicente, Vicent
block	farewell
Spanish	gracias, muchas gracias, adiós
Valencian	gràcies, moltes gràcies, adeu
English	thanks, thank you very much, goodbye

Table 1. This table shows examples of phrases belonging to the blocks for Valencian and Spanish. English phrases are provided for a better understanding of the examples.

able to model Valencian and Spanish separatedly, but it should also be able to model a mixture of both languages. The latter is due to the fact that non-native speakers may use words of their native language when the correct word is unknown. This fact is known as barbarism. The modelling of barbarism is not only relevant to multilanguage environments but also to communities with a large number of immigrants.

In order to provide barbasism tolerance to some extent, the sentences were divided into six blocks, each of which represents a concept in the sentence. As we will see below, this allowed us to construct an automaton that could switch between languages (block-combined language models). The blocks were: greeting, question, information, title, person, and farewell. A set of frequently used phrases was used to build an acceptor automaton for each block. An acceptor automaton accepts only a set of given sentences, in this case, the phrases of the block. Samples of these phrases for the sentences in Figure 1 are shown in Table 1.

Finally, these block-oriented automata were used to build the final automata. Two methods were applied in this task:

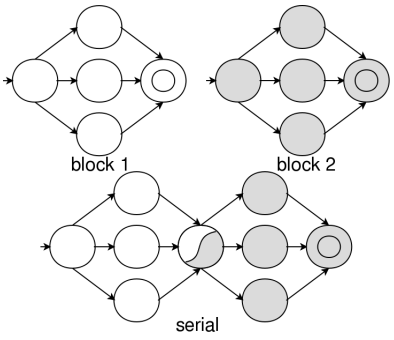


Figure 2. Illustration of the serialization process.

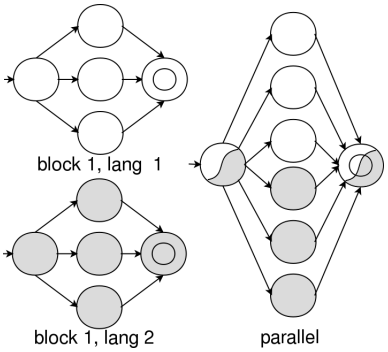


Figure 3. Illustration of the parallelization process.

- Separate language models: an automaton was build for each language. It was made by joining the block-oriented automata in a series. For every two consecutive automata, the final states of the first automaton were merged with the initial states of the second one. Figure 2 shows an example of the serialization process.
- Block-combined language models: a single automaton was built by joining two automata. The automata were joined in parallel on a block-basis manner. Thus, the initial states (and the final states) of each language were merged for each block. Figure 3 shows an example of the parallelization process. Afterwards, the joint blocks were also joined in series. Figure 4 shows an example of the parallelization process for the joint automata.

The automaton corresponding to the block 'person' was, in both cases, the list of all the people in the two languages. This reflects the natural tendency of speakers to call people the way they are used to doing so. Moreover, the names and surnames were allowed separately as well.

4. CORPUS ACQUISITION

The corpus should have about 1 hour per language in order to make a quick acquisition and to be long enough

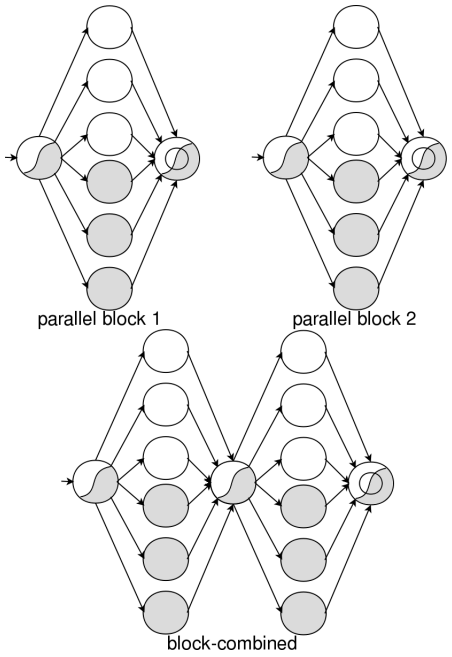


Figure 4. Illustration of the combined parallelization and serialization process.

to train reliable acoustic models in future experiments. Assuming that the average length of an utterance is 3 seconds, we decided to design a set of 120 sentences (60 for each language) for 20 speakers. This provides approximately 1 hour of speech signal per language.

The separate language models were used to generate the corpus. However, a human reviewer was needed to correct the syntactic inconsistencies introduced by the block-oriented automata development, such as gender and number agreement.

The corpus acquisition was developed on the telephone line. Half of the volunteer speakers were native Spanish speakers and the other half were native Valencian speakers. Both languages were acquired from all the participants; thus, non-native speech was recorded for both languages. Male and female speakers were equally distributed in these groups.

In the final design of the corpus, there were five groups of people with four people per group. Each group contained people of all types (men/women, Spanish/Valencian). With this distribution, we ensured a balanced distribution of native and non-native utterances, along with male and female utterances, for both languages.

The Spanish phone set was formed by 26 phonemes in the phonetical scheme that we used. Transcriptions were automatically performed following the rules described in [11] for the SAMPA phonetic alphabet [12]. However, Valencian pronounciation does not follow clear, simple rules as Spanish does. No studies have been done to help us transcribe the sentences automatically. Therefore, the Valencian transcriptions were performed manually for each word of the vocabulary, including all the known phonetic variations. The Valencian phone set we used differs by only one phoneme from the Spanish set. The Spanish phoneme $/c/$ (as in *zapato* $/capato/$) is not present in Valencian, but $/\int/$ (as in *roig* $/ro\int/$) is. The remaining phonemes are shared between the two languages.

Each acquisition session lasted an average of 50 minutes. Although literal reading was compulsory, the speakers were allowed to pronounce Valencian and Spanish as they normally do.

Nearly 2 hours of speech signal were actually acquired (including the silences) for each language. The signal was recorded with a GSM encoding at 8000 Hertz using a 3COM U.S. Robotics modem [13]. Although the GSM encoding signal provides worse quality than a-law or mu-law encoding, the fact is that the GSM encoding is currently being widely used in mobile telephony. As the mobile phone market is rising sharply, most of the potential users of these systems will use mobile phones. Therefore, the use of lossless encoding schemes would not improve the signal quality for real world applications.

The ambient noise was the typical noise found in a computer laboratory with the occasional mobile phone intefering with the phone line. Silences at the beginning and end of the speech signal were not removed.

5. EXPERIMENTS

In this section, preliminar experiments on the adquired corpus are presented. The aim of these experiments was not to measure the recognition accuracy of the models but to observe the impact of multilingual modelling versus monolingual modelling. Not surprisingly, the results presented in this work will not be very accurate if it is taken into account the small corpus provided. Nevertheless, the expriments will show the trend of the applied techniques.

Three evaluation measures have been used in the experiments. These measures have been selected in order to show the most interesting points of this work:

- *Word Error Rate (WER).* The WER is a measure of the ASR quality given a reference sentence. It computes the edition distance between the recognized sentence and the reference sentence.

- *Semantic Word Error Rate (SemWER).* The SemWER is a measure of the recognition quality for the semantic sentence. The semantic sentences are obtained as a subproduct of the recognition proccess and consist of a series of sorted semantic fields. In this work, the semantic fields are the required information (timetables, office hours, phone numbers, e-mail addresses, and office locations) and names. SemWER is computed as the WER for these semantic sentences.

		Spanish	Valencian
Training	Sentences	240	240
	Running words	2887	2692
	Length	1h 33m	1h 29m
	Vocabulary	131	131
	Perplexity	3.32	3.70
Test	Sentences	60	60
	Running words	705	681
	Length	23m	21m
	OOVs	0	2
	Perplexity	5.80	6.14

Table 2. Corpus statistics. OOV (Out Of Vocabulary) words are that ones which have been observed in the test corpus but not in the training one.

Acoustic	Spanish	Valencian
Spanish	15.4 / 12.1	–
Valencian	–	19.5 / 13.5
Shared	14.0 / 10.5	18.9 / 14.0

Table 3. Monolingual WER / SemWER.

- *Language Identification Rate (LIR).* This rate measures the language identification performance of the models. It is computed as the percentatge of sentences whose language is identified correctly.

The Table 2 summarizes the statistics of the corpus. It should be noted that, although the perplexity of the language is very low, the size of the speech corpus is small as well. Therefore, ASR parameters obtained were not as accurate as might be desired, which causes the high error rates presented in the results.

The acoustic models were obtained using the HTK toolkit. The HMMs followed a 3-state left-to-right topology without skips. A 64 Gaussian mixture was used in each state. Furthermore, due to the small amount of data provided, only context independent phonemes were trained.

Table 3 shows monolingual ASR performance as a baseline for the experiments. It can be seen that Valencian performs worse than Spanish. It may be due to the higher variability in Valencian pronunciations, and the quality of the automatic phoneme transcriptior used in the training proccess, which was not as accurate as the Spanish one.

The WER and SemWER for various acoustic and language models is presented in the Table 4. The separate acoustic models behave especially bad for Valencian. As it has been explained before, the Valencian acoustic models have worse parameter estimation than the Spanish ones. However, the second method proposed achieves almost the same performance as the monolingual ones. As shared acoustic models have double training data, the parameters are estimated more precisely.

It might seem weird that block-combined model achieve better SemWER having such WER for Spanish. This fact can be explained by the similarity of both languages. A Spanish utterance may be easily confused with a Valencian one and viceversa by the ASR, especially in block-combined models, in which confusion may come at block levels. This would lead to WER errors as vocabularies of both languages are different. However, the semantic of the recognition would not be affected.

Models	Spanish	Valencian	Average
Sep-Sep	16.0 / 13.6	26.1 / 20.7	21.1 / 17.5
Sha-Sep	15.9 / 11.5	21.0 / 16.3	18.5 / 13.9
Sha-Blo	20.2 / 10.6	21.4 / 16.3	20.8 / 13.5

Table 4. Multilingual WER / SemWER. The column Models contains the models that were used to obtain the rates, where Sep-Sep stands for separate acoustic and language models, Sha-Sep for shared acoustic models and separate language models, and Sha-Blo for shared acoustic models and block-combined models.

Models	Spanish	Valencian	Average
Sep-Sep	99.6	85.8	92.7
Sha-Sep	97.5	99.6	98.6
Sha-Blo	91.2	96.7	94.0

Table 5. Language identification rate. The column Models contains the models that were used to obtain the rates, where Sep-Sep stands for separate acoustic and language models, Sha-Sep for shared acoustic models and separate language models, and Sha-Blo for shared acoustic models and block-combined models.

Looking carefully at the Table 5, it may be noticed that there is a high correlation between the language identification rate of each language, and its correspondent word error rate. In systems where LIR is above 99%, the ASR multilingual performance is almost equivalent to the monolingual system. It is important to observe that LIR rates very well even with these two languages so phonetically and gramatically similar.

6. CONCLUSIONS AND FUTURE WORK

Specifically, experiments were focused on two goals. The first one was to evaluate the corpus in a coupled multilingual speech recognizer which was expected to perform similarly or better than a separate recognizer. The second goal was to obtain a good ratio in speaker-language identification. Both goals were succesfully achieved, even with such a small corpus.

In average, the best results were achieved using shared acoustic models and separate language models. It is obvious that with such similarity in phonetics, shared models behave better because more training data is being used for the same models. However, it should be

noted that this conclusion cannot be extrapolated for languages with more different phonetics [14]. Following the same reasoning, block-combined models would mix up word from different languages, because for many of them they are pronounced almost in the same way. Consequently, separate language models rate higher. Other works sustain this observation for a wider range of languages [15].

The purpose of this paper was to acquire a corpus to assess the viability of this research line. The acquisition of an acoustic corpus is a tedious task, and therefore, we decided to acquire a minimal corpus which may be a drawback for larger experiments. However, huge Spanish acoustic resources are widely available in the community. The Valencian acoustic signal of this corpus could be used for adaptation purposes, e.g., to adapt good Spanish acoustic models to the Valencian dialect by means of speaker adaptation techniques [16]. Previous work supports this approximation [17].

Finally, further experiments are planned for a larger corpus which is already being acquisited for all the Catalan dialects [18].

Acknowledgements

The authors wish to thank the volunteer speakers who participated in the multilingual corpus acquisition.

7. BIBLIOGRAPHY

[1] A. L. Gorin, G. Ricardi, y J. H. Wright, "How may i help you?," *Speech Communication*, vol. 35, pp. 113–127, 1997.

[2] J. Chu-Carroll y R. Carpenter, "Vector-based natural language call-routing.," *Computational Linguistics*, vol. 25(3), pp. 361–388, 1999.

[3] R. Billi, F. Canavesio, y C. Rullent, "Automation of telecom italia directory assistance service: Field trials results," in *Proc. IVTTA 1998*, Turin, 1998, pp. 11–16.

[4] S. Furui, "Toward robust speech recognition under adverse conditions," in *Proceedings ESCA Workshop on Speech Processing in Adverse Conditions*, Nov. 1992, pp. 31–41.

[5] H. G. Huang, "Speech recognition in adverse environments.," *Computer Speech and Language*, vol. 5, pp. 275–294, 1991.

[6] R. Eklund y A. Lindström, "Xenophones: An investigation of phone set expansion in swedish and implications for speech recognition and speech systhesis.," *Speech Communication*, vol. 35, pp. 81–102, 2001.

[7] Sander J. van Wijngaarden, "Intelligibility of native and non-native dutch speech.," *Speech Communication*, vol. 35, pp. 103–113, 2001.

[8] Institut Valencià d'Estadística, Comunitat Valenciana, *Població en vivendes familiars de 3 i més anys, segons el coneixement del valencià, l'edat i el sexe.*, 2001.

[9] J.E. Diaz-Verdejo, A.M. Peinado, A.J. Rubio, E. Segarra, N. Prieto, y F. Casacuberta, "Albayzin: a task-oriented spanish speech corpus," in *Proceedings of LREC*, 1998, vol. 1, pp. 497–501.

[10] F. Mas, J. Vicedo, y C.D. Martínez-Hinarejos, "Development of a voice-based telephone information system," in *Actas de las III Jornadas de Tecnologías del Habla*, Valencia, Spain, Nov. 2004.

[11] A. Quilis, *Tratado de fonología y fonética españolas.*, Madrid (Gredos), second edition, 1999.

[12] UCL, *SAMPA computer readable phonetic alphabet*, 1993.

[13] USRobotics, *3Com U.S. Robotics 56K Message Modem Users Guide and Reference*, 1993.

[14] J. Köhler, "Multilingual phone models for vocabulary-independent speech recognition tasks," *Speech Communication*, vol. 35, pp. 21–30, 2001.

[15] U. Uebler, "Multilingual speech recognition in seven languages," *Speech Communication*, vol. 35, pp. 53–69, 2001.

[16] C.J. Leggeter y P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models.," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[17] T. Schultz y A. Waibel, "Language-independent and language-adaptative acoustic modeling for speech recognition," *Speech Communication*, vol. 35, pp. 31–51, 2001.

[18] A. Moreno, A. Febrer, y L. Márquez, "Generation of Language Resources for the Development of Speech Technologies in Catalan," in *Proc. of LREC'06*, 2006.

CUANTIFICACIÓN DE LA CALIDAD DE LA VOZ PARA SU EVALUACIÓN CLÍNICA POR MEDIO DEL ANÁLISIS ACÚSTICO

Jesús B. Alonso [1], Miguel A. Ferrer [1], José de León [2], Carlos M. Travieso [1]

[1] Dpto. de Señales y Comunicaciones, Universidad de Las Palmas de Gran Canaria
Campus de Tafira, 35017 Las Palmas de Gran Canaria, SPAIN
email: jalonso@dsc.ulpgc.es

[2] Servicio de Otorrinolaringología, General Hospital de Gran Canaria "Dr.Negrín"
Las Palmas de Gran Canaria, SPAIN

RESUMEN

En lo que respecta a la evaluación del sistema fonador, las técnicas basadas en métodos de inspección visual directa son las técnicas que permiten obtener la mejor valoración del estado del sistema fonador de un locutor, sin embargo, estas técnicas presenta ciertos inconvenientes como el hecho de tratarse de técnicas invasivas, con la correspondiente incomodidad para el locutor que se preste a la evaluación, o el hecho de que este tipo de pruebas han de ser realizadas por profesionales del ámbito médico especializado.

En este trabajo se propone un procedimiento para evaluar la calidad de la voz de un locutor a partir de una grabación de audio. Este procedimiento, consiste en evaluar cuatro características de la señal de voz: la estabilidad, la riqueza espectral, la presencia de ruido y su comportamiento no lineal. Además, se han identificado las medidas de calidad de la voz que mejor cuantifican a cada uno de estas características de la voz. Finalmente, el trabajo concluye presentando una metodología del análisis acústico para la evaluación clínica de la voz.

1. INTRODUCCIÓN

Medir la calidad de la voz de un locutor a partir de una grabación consiste en cuantificar diferentes características acústicas de la voz, las cuales permitan compararla respecto a ciertos patrones de referencia. En la bibliografía actual no existe una única medida que documente todas las características acústicas de la voz.

Una medida de la calidad acústica de la voz basada en una valoración auditiva es intrínsecamente difícil de establecer como referencia comparativa entre distintas voces y distintos evaluadores. En la bibliografía actual se han realizado varios intentos de objetivar la medida de la calidad de voz por medio de medidas clínicas multidimensionales basadas en métodos auditivos. Existen ejemplos bien conocidos como son: la escala GRABS de Japón [1] y su formulario extendido, desarrollado y aplicado en Europa [2], o el utilizado en Suecia donde se ha trabajado en otro juego de descriptores clínicos de la voz referentes a la percepción [3], o como en el que se introduce un juego de características fonéticas las cuales intentan agregar información de la excitación del tracto vocal en la medida de la calidad de la voz [4]. Estos procedimientos de medida de la calidad de la voz pretenden alcanzar una medida objetiva a partir de una valoración subjetiva.

Existen diferentes trabajos en los que se proponen medidas objetivas de la calidad de la voz obtenidas a partir de una grabación de audio [6][7][8][9][10[11][12]. Una característica común de la mayoría de estos trabajos es la de utilizar, para el cálculo de medidas de la calidad de la voz, una grabación de un sonido sonoro sostenido, principalmente una vocal. Esto es debido a que durante la fonación de este tipo de sonidos el sistema de producción de voz pone en funcionamiento gran parte de sus mecanismos (flujo glótico de aire constante, vibración de las cuerdas vocales de forma continuada, …) permitiendo captar cualquier anomalía de estos mecanismos. En estos trabajos, los distintos autores presentan diferentes conjuntos de medidas con la finalidad de cuantificar objetivamente la calidad de voz. Todos ellos tienen en común la necesidad de realizar diferentes medidas de la voz con la finalidad de recoger en el proceso de medida los distintos aspectos de las características acústicas de la voz.

2. LA CALIDAD DE LA VOZ DESDE DIFERENTES DOMINIOS

Una grabación de voz aporta diferentes características respecto a la calidad de la voz del locutor. Esta señal de voz registrada es posible representarla en diferentes dominios, lo cual resulta interesante ya que cada dominio en el que se puede representar la señal de voz,

manifiesta de forma preferencial alguna de las características de la voz del locutor. Los principales dominios estudiados en el procesado de señales de voz están centrados en los siguientes dominios:

- *Dominio Temporal*
- *Dominio Espectral*
- *Dominio Cepstral*
- *Dominio del Modelo Inverso*

Sin duda alguna gran cantidad de los trabajos sobre procesado digital de señales de voz existentes en la bibliografía actual centran sus estudios en estos dominios. Sin embargo, nuevas líneas de trabajo presentan nuevos dominios derivados de los anteriores.

De igual forma resulta de interés estudiar nuevos dominios que cada vez toman mayor importancia en el procesado de señales: los *dominios que manifiestan el comportamiento no lineal de la señal de voz*.

A continuación se describen las manifestaciones más relevantes de cada uno de los diferentes dominios.

2.1. Dominio Temporal

La envolvente de señal de voz de un locutor con voz de alta calidad resulta más regular que la envolvente de la señal de voz del locutor con voz de baja calidad. Este hecho resulta más significativo en intervalos de tiempo muy pequeños. Los fenómenos más característicos que permiten diferenciar entre voz de alta calidad y voz de baja calidad son:

- La energía de la señal contenida en un intervalo de tiempo pequeño, varía mucho de un intervalo al siguiente en las señales de voz de baja calidad respecto a las señales de voz de alta calidad.
- Otra característica que presenta las voces de baja calidad es la existencia de intervalos carentes de periodicidad durante fonaciones sonoras sostenidas.

2.2. Dominio Espectral

Las muestras de voz de un sonido sonoro sostenido producidas por un locutor con voz de baja calidad presentan las siguientes características diferenciales frente a las voces de alta calidad:

- Menor regularidad de la envolvente del espectro, principalmente en las bajas frecuencias.
- Mayor porcentaje de energía en las bajas frecuencias respecto a la energía total.
- Presencia de bloques de energía en las altas frecuencias, los cuales son debido a la presencia de ruido glótico.
- Gran variación del espectro de una trama con respecto a las tramas contiguas.

Una característica de las voces de alta calidad es la energía del espectro concentrada alrededor de determinados formantes, principalmente el primer y el tercer formante, mientras que las voces de baja calidad se caracterizan por presentar gran componente de ruido alrededor de los formantes.

Las voces de alta calidad presentan gran cantidad de riqueza espectral. Por el contrario, las voces de baja calidad presentan muy poca componente armónica, estando ésta concentrada en las componentes de muy baja frecuencia. En cualquier caso, la cantidad de riqueza espectral es una peculiaridad de la voz de un locutor, sin embargo, la variación de la riqueza espectral a lo largo del tiempo durante la fonación de un sonido sonoro sostenido si es indicativo de una voz de baja calidad.

Otra característica típica de las voces de baja calidad es la de presentar, durante la fonación de un sonido sonoro sostenido, variaciones en el ritmo de vibraciones de las cuerdas vocales, lo cual es equivalente a decir que existe variaciones de la frecuencia de *pitch*.

2.3. Dominio Cepstral

Por medio de una representación cepstral es posible identificar características que permiten valorar la calidad de la voz (la envolvente del espectro, la riqueza espectral, identificar las componentes armónicas y las componentes de ruido, etc). En el caso de una fonación de un sonido sonoro sostenido, realizando una representación cepstral de diferentes tramas con duración equivalente a tres periodos de *pitch*, pueden identificarse diferentes características.

La riqueza espectral de una muestra de voz puede cuantificarse por medio de la amplitud y anchura de la componente cepstral correspondiente al *pitch*. La existencia de un pico de una amplitud considerable indica la presencia notable de energía en dicha componente armónica, siendo ésta una característica de las voces de alta calidad. También, una anchura reducida del pico cepstral, correspondiente al *pitch*, indica la alta estabilidad de la frecuencia de *pitch* para tres periodos consecutivos, también siendo ésta una característica de las voces de alta calidad. Características como la amplitud y la estrechez del pico cepstral correspondiente al segundo armónico son, al igual que ocurre con el pico del *pitch*, aspectos que diferencian a las muestras de voz de alta y baja calidad. La presencia de un pico cepstral correspondiente al primer armónico el cual resulte más estrecho que el pico ***cepstral*** correspondiente al segundo armónico resulta una característica propia de las voces de alta calidad.

La presencia de ruido glótico en la señal de voz puede estimarse por medio de relaciones entre diferentes regiones de los cepstrum: la componente armónica (las componentes cepstrales correspondientes al *pitch* y sus armónicos), y la componente de ruido (el resto de las componentes cepstrales).

2.4. Dominio del Modelo Inverso

En este dominio se estima, a partir de la señal de voz, la forma de onda del pulso de aire que se produce debido a las cuerdas vocales durante la fonación de un sonido

sonoro, es decir, la forma de onda de la señal residual (forma de onda del flujo glótico). Esta estimación de la forma de onda se obtiene mediante un filtro inverso sobre la señal de voz, donde se elimina el efecto del tracto vocal y el efecto de la radiación en los labios.

Las diferentes manifestaciones de la calidad de la voz en la señal glótica se cuantifican por medio de los valores de amplitudes, de los instantes de inicio de la apertura, de máxima apertura de las cuerdas vocales, de inicio del cierre, etc, y diferentes relaciones entre distintos instantes del ciclo glótico: *open quotient, speed quotient, closing quotient*, etc.

2.5. Dominio No Lineal

Los principales sistemas comerciales que permiten evaluar objetivamente la calidad de la voz de un locutor a partir de una grabación (Dr Speech *(Tiger Elemetric)*,SSVA *(System for Sigle Voice Analysis)*, MDVP *(Multi-Dimensional Voice Program)* ,EVA *(Evaluation Vocal Assistee)*, CSL *(Computerized Speech Laboratory)* PRAAT, VISHACSRE *(Computerized Speech Research Environment)*, MEDIVOZ, etc) no evaluan las características no lineales presentes en la señal de voz.

El modelo más popular de caracterización del sistema de producción de voz es un sistema tiempo-variante, basado en teorías acústicas lineales, consistente en un modelo de fuentes y filtros (*source/filter model*). La existencia de variaciones en la amplitud espectral de la señal de voz, en la frecuencia fundamental, para una excitación estable, son indicios para presuponer un comportamiento no lineal.

Ha sido identificada [13] la presencia de una frecuencia fundamental (fs) y de un subarmónico (fs/2) en la señal de voz, donde el efecto de los subarmónicos se manifiesta o bien en una modulación de amplitud o bien en una modulación de frecuencia, existiendo la posibilidad de que se produzcan los dos fenómenos. Estudios realizados [14] indican que un 31% de las muestras de voz de locutores patológicos presentan subarmónicos. Si embargo, también se han identificado la existencia de subarmónicos en las voces de alta calidad [15], donde el fenómeno se estima que lo presenta el 10.5% de la población, siendo esta presencia de subarmónicos no necesariamente indicativa anormalidades.

Existen dos posibles teorías para justificar la presencia de subarmónicos:

- La teoría de Titze [16]: la producción de los subarmónicos se debe a asimetrías mecánicas o geométricas entre las cuerdas vocales.
- La teoría de Svec [17]: la frecuencia subarmónica se debe a la combinación de dos modos vibracionales (bifonación: la presencia de dos frecuencias principales) cuyas frecuencias tienen la relación 3:2.

Sin embargo, ambas teorías son las mismas según [18], donde los autores indican que la presencia de bifonación puede ser inducida por asimetría entre cuerdas derecha-izquierda o por desincronización en la vibración anterior-posterior, siendo estas asimetrías o desincronizaciones en las fases de la onda mucosa el origen de los subarmónicos [19], lo que se asocia a significativas diferencias en las masas o propiedades viscoelásticas entre las cuerdas vocales, siendo este comportamiento susceptible de ser modelado por modelos de fonación no lineales. Esta idea es compartida por [20], que propone la necesidad del uso de modelos no lineales para explicar el efecto de la viscosidad de la mucosa de las cuerdas vocales en aspectos como su vibración (la mucosa en la superficie de las cuerdas vocales genera tensión superficial y causa adhesión).

Desde otra perspectiva, el modelo tradicional del tracto vocal asume que la propagación de la onda sonora se realiza por medio de una propagación plana. Sin embargo, las medidas de presión sonora y las medidas de variación de volumen se adaptan de forma más realista a un modelo no lineal de fluido de dinámicos, debiéndose esto a que las cavidades existentes entre las cuerdas vocales y las falsas cuerdas vocales pueden provocar un torbellino, incluso pudiéndose producir torbellinos periódicos, el cual excita al tracto vocal durante la fase de cierre de las cuerdas vocales.

A partir de estudios empíricos, se ha estudiado la dimensión fractal, por parte de diferentes autores, coincidiendo entre ellos en la baja dimensionalidad de la señal de voz de alta calidad. En [5] se afirma que la cantidad de alinealidades en el sistema vocal es una importante determinación de la presencia de fonación anormal, también indica que la dimensionalidad del espacio de fases, utilizado para caracterizar el *attractor*, puede estar relacionada con la cantidad de masa de las cuerdas puestas en juego.

3. FENÓMENOS FÍSICOS IMPLICADOS EN LA CALIDAD DE LA VOZ

En el apartado anterior se ha realizado un recorrido por los principales dominios de representación en los que se puede estudiar la señal de voz, describiendo las diferentes manifestaciones, propias de cada dominio, que permiten valorar la calidad de la voz. Cada una de las diferentes manifestaciones mencionadas caracteriza un fenómeno físico que interviene en la generación de la voz y sin embargo el mismo fenómeno físico puede manifestarse en diferentes dominios de represtación de la señal de voz. En este trabajo se ha identificado el conjunto de fenómenos físicos que son necesarios describir para realizar una correcta documentación de la calidad de voz de un locutor. Los cuatro fenómenos físicos identificados son los siguientes:

- **Estabilidad de la voz**: es la capacidad de un locutor de producir, durante la producción de un sonido sonoro sostenido, un flujo de aire con una intensidad constante para excitar las cuerdas vocales. Este fenómeno físico se cuantifica a partir de medidas de estabilidad de la señal de voz.

- **Riqueza espectral**: se define como la capacidad de generar, durante la fonación de un sonido sostenido, un movimiento periódico de las cuerdas vocales y de producir una excitación sonora del tracto vocal con gran cantidad de componentes espectrales. Este fenómeno físico se cuantifica por medio del cálculo de la estabilidad de la frecuencia de *pitch* y por la presencia de armónicos con energía elevada en las diferentes bandas de frecuencias.
- **Presencia de ruido**: se define como la presencia de ruido glótico en la voz, durante la fonación de un sonido sonoro sostenido, debido a carencias en la fase de cierre de las cuerdas vocales. Este fenómeno físico se cuantifica por medio de la presencia de ruido no estacionario en la voz.
- **Comportamiento no lineal**: este fenómeno físico es ocasionado por un funcionamiento anómalo de las cuerdas vocales, debido a irregularidades en las masas que interviene en la fase de cierre de cuerdas, la existencia de asimetrías en el movimiento de dichas cuerdas y factores relacionados con la mucosa que recubre a las cuerdas. Estos fenómenos son cuantificables por medio de la identificación de comportamientos no lineales en la señal de voz.

Una voz de calidad anormal presenta al menos uno de los valores correspondientes a la cuantificación de los cuatro fenómenos físicos fuera de los rangos de normalidad. Este procedimiento de cuantificación de la calidad de la voz permite identificar calidades de voz anómalas de diferente origen, es decir, la voz de un locutor con baja calidad debida a un estado incipiente de una patología orgánica presentará probablemente la alteración de la "Estabilidad de la voz" mientras que otro locutor con un estado incipiente de una patología funcional presentará valores anormales de "Presencia de ruido". A medida que la calidad de la voz de un locutor empeora, aumentará el número de fenómenos físicos y la desviación de los valores de la cuantificación de dichos fenómenos respecto a los valores de normalidad.

Estos cuatro diferentes fenómenos físicos son cuantificables en diferentes dominios de representación de la señal de voz, existiendo diferentes medidas objetivas de la calidad de la voz que cuantifican con mayor o menor precisión un mismo fenómeno físico.

4. ESTRATEGIA PARA SELECCIONAR LAS MEDIDIDAS DE CALIDAD DE LA VOZ

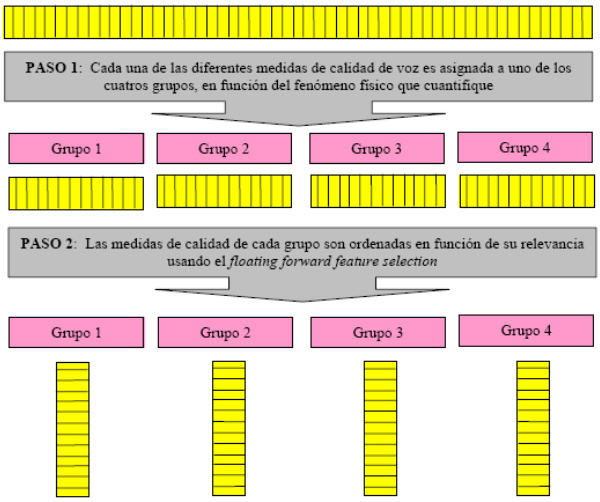
Hasta ahora en la bibliografía actual nos encontramos diferentes escenarios en el estudio de las medidas de calidad de la voz. Por un lado, nos encontramos los resultados obtenidos a partir de bases de datos de voces de diferentes calidades donde el número de muestras es reducido, en los que se presentan los resultados obtenidos para el estudio de una determinada medida o de un subconjunto de medidas. Por otro lado existen estudios donde, a partir de una base de datos de voces de diferentes calidades con un número significativo de muestras, se realiza el estudio de conjuntos de medias e incluso se aplican técnicas de selección de características con el fin de cuantificar la relevancia de las diferentes medidas de calidad de la voz. Existen diferentes aspectos en la metodología de estos estudios que no permiten realizar una generalización de los resultados obtenidos.

Debido principalmente a la inexistencia de una gran base de datos, resulta necesario demostrar la independencia de los resultados de la cuantificación de la relevancia de las medidas de calidad de la voz con la base de datos.

Otro aspecto a considerar, es la gran cantidad de estudios en los que no se tiene en cuenta en el proceso de selección de medidas de la calidad de la voz todo el conocimiento existente en el ámbito clínico respecto a la naturaleza de los trastornos del sistema fonador que producen una alteración de la calidad de la voz. En este sentido, apreciamos la necesidad de disponer de medidas de calidad de la voz que cuantifiquen cada uno de los cuatro fenómenos físicos identificados con el fin de poder documentar la gran variedad de tipos de alteraciones de la calidad de la voz existentes, ya que cualquier voz de baja calidad presenta al menos la alteración de una de las cuatro fenómenos físicos identificados.

En este trabajo se propone clasificar las diferentes medidas de calidad de voz en los cuatro grupos atendiendo al fenómeno físico que cuantifica. Una vez identificadas las diferentes medidas de calidad de voz pertenecientes a cada una de los diferentes grupos de características es necesario estudiar la relevancia de las medidas para cada grupo. El método de selección de características que se propone, para estudiar la relevancia de las medidas de cada grupo, es el algoritmo *floating forward feature selection*, ya que permite obtener ordenadas las medidas de la calidad de la voz en función de su relevancia y tiene un coste computacional razonable.

En concreto, el método propuesto consiste en asignar las diferentes medidas de calidad de la voz en uno de los cuatro grupos de medidas que hemos identificado. Posteriormente, las medidas de cada grupo se ordenan en función de su relevancia por medio del método *floating forward feature selection*. Este método proporciona, para cada grupo de medidas, un conjunto óptimo de características, y es por ello que el proceso de evaluación de cada uno de los diferentes grupos de medidas se repite k iteraciones, y posteriormente se ordenan las características en función del número de ocasiones en las que para las diferentes k iteraciones estuvieron seleccionadas dentro del subconjunto óptimo.



Para la evaluación del sistema propuesto para identificar al subconjunto óptimo de medidas de calidad de la voz, se han utilizado dos bases de datos totalmente independientes. Con una de ellas, se realizó la selección de las medidas y su posterior evaluación, y con la otra base de datos sólo se realizó la evaluación, con el fin comparar los resultados obtenidos en diferentes sistemas de clasificación.

5. DATOS

Con el fin de evaluar el sistema de selección de medidas de calidad de la voz, se utilizó una colección de voces de locutores los cuales presentan diferentes patologías en el sistema fonador y voces de locutores sanos. Dentro de las muestras de voces de locutores patológicos se dispuso de muestras con distintos grados de afección consiguiendo disponer de un variado rango de calidades de voz. Para el estudio se dispuso de dos bases de datos.

5.1. Base de Datos 1

La grabación de voz ha sido realizada en un centro hospitalario, en una habitación en la que se ha tenido en cuenta los niveles de contaminación acústica, intentándose obtener un compromiso entre muy bajos niveles de ruido, y niveles de ruido realistas, ya que si no fuese así resultaría imposible aplicar a las condiciones normales de trabajo cualquier resultado obtenido a partir de la base de datos.

El contenido de las grabaciones corresponde a la fonación de la vocal "a" castellana de forma sostenida y no susurrada. Este formato consiste en audio digital, con una frecuencia de muestreo de 22050 KHz y con una resolución de 16 bits por muestra. Se utiliza una tarjeta de sonido convencional y un micrófono común, con ancho de banda lineal superior a 11 Khz.

En la siguiente tabla se presenta los datos pormenorizados por sexos y grados de patologías.

Grupo de Muestras de Voces		Hombres	Mujeres	Total	
Muestras de voces sanas		43	44	87	
Muestras de voces patológicas	Disfonía leve	7	19	26	93
	Disfonía moderada	22	14	36	
	Disfonía severa	18	13	31	

5.2. Base de Datos 2

Esta base de datos es la *Voice Disorders Database Model* 4337 desarrollada por el *Massachussets Eye and Ear Infirmary Voice and Speech Lab*.

Contiene 1400 muestras de voz de aproximadamente 700 sujetos de habla inglesa. Consiste en la fonación sostenida de la vocal "ah", de pacientes con una amplia variedad de patologías del sistema fonador, al igual que 53 muestras de voces sanas. Para su elaboración se usó un micrófono capacitivo en una cámara acústicamente aislada. La señal fue registrada con un grabador DAT con una frecuencia de muestreo de 44.1 Khz, utilizándose un preamplificador calibrado.

6. RESULTADOS

Se han identificado las principales medidas de calidad de la voz presentes en la bibliografía actual, posteriormente han sido clasificadas en uno de los cuatro grupos de medidas, y tras aplicar la metodología propuesta, las medidas de cada uno de los cuatro grupos han sido evaluadas por separado [6], utilizando un diferentes sistemas de clasificación (redes neuronales probabilísticas (PNN), redes neuronales *feedforward* (NN) con una capa oculta, clasificadores basados en funciones de base radial (RBF), máquinas de vectores soporte (SVM) y clasificadores basados en los K vecinos más cercanos (KNN)). La evaluación se realizó de forma incremental, añadiendo de forma individual medidas de calidad de la voz en orden, en función de su relevancia. Se observó que la tasa de éxito en los diferentes clasificadores no se incrementaba respecto a los resultados obtenidos utilizando la medida de calidad con mayor relevancia, a medida que se añadían más medidas de calidad de la voz en el conjunto de características de entrada de los diferentes clasificadores.

Las medidas que mayor relevancia presentaron para los diferentes grupos [6] fueron:

Fenómeno Físico	**Medida de Calidad de la Voz**
Estabilidad de la voz	Amplitude Perturbation Quotient (APQ)
Riqueza Espectral	Media Valor Cepstral del primer pico de los cepstrum derivados
Presencia de Ruido	Media Relación Espectral Subarmónico - armónico de al señal residual
Comportamiento no Lineal	Media Área inferior del biespectro integrado

Los resultados obtenidos para los diferentes clasificadores son los siguientes:

Clasificador	Tasa de Éxito Base de datos 1	Tasa de Éxito Base de datos 2
PNN	80.95%	78.21%
NN	88.21%	84,36%
RBF	86.94%	84.53%
SVM	**84.34%**	**85.31%**
KNN	84.66%	85.38%

Por lo tanto, además de resultar suficiente, con una medida de calidad de la voz para cada grupo de medidas, se observa que los resultados obtenidos con la base de datos 1 (utilizada para la selección) y los resultados obtenidos con la base de datos 2 (utilizada sólo para la evaluación) resultan muy similares.

Se ha estudiado además los rangos de normalidad de las diferentes medidas, estimándose en un intervalo de confianza del 95% y asociando el valor de la medida al fenómeno físico que cuantifica, obteniendo los siguientes rangos:

Estabilidad de la Voz	0.000……0.136
Riqueza Espectral	0.919……1.759
Presencia de Ruido	0.189……0.561
Comportamiento no Lineal	0.530……0.754

7. CONCLUSIONES

En este trabajo se han identificado los diferentes fenómenos físicos que caracterizan la calidad de la voz y que necesitan ser cuantificados para una correcta documentación de la calidad de la voz, introduciendo de forma novedosa la cuantificación del comportamiento no lineal de la señal de voz.

Además, se propone una estrategia de selección de medidas, que combina la experiencia clínica con los métodos ciegos de selección de características, y que identifican las medidas de calidad de la voz que mejor permiten documentar la calidad de la voz de un locutor.

Finalmente, se propone un protocolo de evaluación de la calidad de la voz por medio de 4 medidas objetivas obtenidas a partir de una grabación de voz, que permiten asociar el valor de la medida con la cuantificación de un determinado comportamiento fisiológico del sistema fonador.

8. BIBLIOGRAFÍA

[1] Hirano, M., Clinical Examination of Voice. New York, Springer-Verlag, 1981

[2] Dejonckere, P. H. Remacle, M. Fresnel-Elbaz, E. Woisard, V. Crevier-Buchman, L. Millet, B., "Differentiated perceptual evaluation of pathological voice quality: reliability and correlations with acoustic measurements". Revue de Laryngologie Otologie Rhinologie, vol. 117, no.2, pp. 219-224, (Bordeaux) 1996.

[3] Hammarberg, B. y Gauffin, J., Perceptual and acoustic characteristics of quality differences in pathological voices as related to physiological aspects,. in O. Fujimura & M.Hirano (eds.), Vocal Fold Physiology, pp 283-303, 1995

[4] Laver, J., The Gift of Speech, Edinburgh University Press,1991

[5] Robert F. Orlikoff, R. J. Baken, "Curing Diagnosis: Improving the Taxonomy of Phonatory Dysfunction", Sixth Conference on Advances in Quantitative Laryngology. Hamburg, Germany, April 2003

[6] Jesús B. Alonso, Metodología del Análisis Acústico para la Evaluación Clínica de la Voz, Tesis Doctoral, Universidad de Las Palmas de Gran Canaria, 2006

[7] Boyanov, B; Hadjitodorov, S. "Acoustic analysis of pathological voices. A voice analysis system for the screening of laryngeal diseases". IEEE Engineering in Medicine and Biology Magazine, vol.16,no.4, pp.74 – 82, 1997.

[8] Hansen, J.H.L.; Gavidia-Ceballos, L.; Kaiser, J.F. "A nonlinear operator-based speech feature analysis method with application to vocal fold pathology assessment". IEEE Transactions on Biomedical Engineering, vol.45, no. 3, pp.300-313, March 1998.

[9] Stefan Hadjitodorov, Petar Mitev, "A computer system for acoustic analysis of pathological voices and laryngeal diseases screening", Medical Engineering Physics, vol .24, pp. 419-429, 2002.

[10] Michaelis D.; Frohlich M.; Strube H. W. "Selection and combination of acoustic features for the description of pathologic voices". Acoustical Society of America. vol.103, no.3, pp.1628-1640, 1998.

[11] Boyanov B., Doskov D., Mitev P., Hadjitodorov S., Teston B. , "New cepstral parameters for description of pathologic voice", Comptes Rendus de L'Academie Bulgare des Sciences (Ann. of Bulgarian Academy of Sciences), vol.53, no.3, pp.41-44, 2000

[12] Godino-Llorente, J.I.; Aguilera-Navarro, S.; Gomez-Vilda, P. "Non supervised neural net applied to the detection of voice impairment". Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '00. vol.6, pp.3594-3597, 2000.

[13] Xuejing Sun and Yi Xu, "Perceived Pitch of Synthesized Voice with Alternate Cycles", Journal of Voice, vol.16, no. 4, pp. 443-459, 1995

[14] Nuñez Batalla F, Suarez Nieto, C., Muñoz Pinto, C., Baragaño Rio, L., Álvarez Zapico, M.J., Martínez Ferreras, A., "Estudio Espectrográfico de la disfonía: Subarmónicos", Acta Otorrinolaring. Esp., vol.52,no. 1, pp.52-56, 2000.

[15] C. Michael Haben, Karen Kost, George Papagiannis, "Lateral Phase Mucosal Asymmetries in the Clinical Voice Laboratory", Journal of Voice, vol.17,no. 1, pp.3-11,2003.

[16] TItze, IR., Principles of Voice Production, Englewood Cliffs, NJ: Prentice-Hall, Inc, 1994 .

[17] Svec JG, Schutte HK, Miller DG." A Subharmonic vibratory pattern in normal vocal folds", Journal of Speech and Hearing Research, vol.39, no.1, pp.135-143,1996.

[18] Jürgen Neubauer, Patrick Eysholdt, Ulrich Eysholdt, Hanspeter Herzel, "Spatio-temporal analysis of irregular vocal fold oscillations: Biphonation due to desynchronization of spatial models", J. Acoustical Society of America, vol.110, no. 6, pp.3179-3192, December 2001.

[19] C. Michael Haben, Karen Kost, George Papagiannis, "Lateral Phase Mucosal Asymmetries in the Clinical Voice Laboratory", Journal of Voice, vol.17,no. 1, pp.3-11,2003.

[20] Stephane Ayache, Maurice Ouaknine, Philippe Dejonkere, Pierre Prindere, Antoine Giovanni, "Experimental Study of the Effects od Surface Mucus Viscosity on the Glottic Cycle", Journal of Voice, vol.18, no.1, 2004.

ESTIMACIÓN DE MATRICES DE TRANSFORMACIÓN MEDIANTE FUNCIONES DE COSTE BASADAS EN EL MÍNIMO ERROR DE BAYES Y SU APLICACIÓN A GMM

J. L. Navarro-Mesa[†], F. D. Lorenzo-García[*], A.G. Ravelo-García[†], S. I. Martín-González[†], P. J. Quintana-Morales[†], E. Hernández-Pérez[†]
Departamento de Señales y Comunicaciones[†]. Departamento de Ingeniería Telemática[*]
*Universidad de Las Palmas de Gran Canaria. Spain. †jnavarro@dsc.ulpgc.es, *fdlorenzo@dit.ulpgc.es*

ABSTRACT

Presentamos dos nuevos métodos para mejorar los resultados de clasificación en problemas de hipótesis binarias donde las clases son modeladas mediante mezclas de Gausianas. Partimos de una función de coste basada en el error de Bayes y a partir de la cota del error dada por Chernoff proponemos otra basada en la distancia del mismo nombre. A partir de ambas funciones se estima una matriz de transformación a un nuevo espacio de características que minimice el error de clasificación entre las clases. La estimación de las matrices se hace por el método de paso ascendente y se da la formulación resultante. Para ello, obtenemos las derivadas de las funciones y las aplicamos a una versión simplificada donde se selecciona una componente de cada mezcla participar en la estimación. Se estudia la inicialización del método y se propone el método de inicialización que mejores resultados nos ha proporcionado en los experimentos. Este método de inicialización está basado en una maximización de la divergencia entre las clases. Los experimentos se han realizado sobre una base de datos de voz con y sin patología y muestra que nuestra aproximación representa una mejora en los resultados de clasificación sobre otros métodos clásicos de diseño de matrices de transformación.

1. INTRODUCCIÓN

El problema de la discriminación entre clases es bien conocido y ha sido estudiado en múltiples trabajos. El principal problema es que los resultados de clasificación se degradan significativamente cuando las clases son altamente confundibles y, en consecuencia, el error es alto. Nuestro principal objetivo es abordar este problema en función de conseguir una baja tasa de error en un test de hipótesis binario. En la literatura se han propuesto varios métodos discriminantes para dar solución al problema de la confusión entre clases. Algunos métodos usan criterios optimizados basados en información mutua [1] o en error mínimo de clasificación [2] para calcular el modelo de parámetros tal que la separación entre las clases sea maximizada. En [3] se usa una medida ponderada de la divergencia como criterio para encontrar una matriz de transformación que mapee las características originales a un subespacio más discriminante. Esta y otras aproximaciones [4,3,6] usan proyecciones lineales que mapean las características del espacio original a un subespacio transformado maximizando o minimizando un criterio convenientemente formulado, p.e., maximizar la separación entre clases.

En este artículo partimos de la idea de que las propiedades estadísticas de las clases están recogidas mediante un modelo de mezcla de gaussianas. Estos modelos describirán las clases en un espacio de características original y mediante una matriz de transformación pasaremos a un nuevo espacio en el cual se cumplan unos objetivos expresados en unas funciones de coste. Una vez obtenida la matriz se transforman todos los vectores y se calculan los modelos de cada clase. Es en este espacio donde se realizará finalmente la clasificación.

Nuestro objetivo fundamental es minimizar el error de clasificación bayesiano. Para ello, partimos de una formulación clásica del mismo dando una primera función de coste y mostramos los elementos matemáticos principales. Dada la complejidad de las expresiones que se obtienen elegimos el método de paso descendente para estimar una matriz de transformación óptima. Una vez hecho esto aprovecharemos la desigualdad que relaciona el error de clasificación con la cota de Chernoff y, posteriormente, de ésta pasar a la distancia del mismo nombre [7]. Tomando como nuevo punto de referencia dicha distancia se llega a una nueva función de coste cuyo objetivo principal es el mismo, minimizar el error de clasificación, si bien en este caso se hará una maximización de dicha función. Las expresiones a las que se llegan también son complejas y utilizamos el método de paso ascendente para estimar una matriz de transformación óptima. Una idea clave de nuestro trabajo es que si bien con la primera función de coste pretendemos minimizar el error de clasificación a partir de su definición con la segunda minimizamos una cota del mismo.

Una vez definidas las funciones de coste y el método de estimación de las matrices centramos nuestro trabajo en los experimentos, particularmente, clasificación de habla con y sin patología. Los vectores en el espacio original representan características de tipo energía logarítmica obtenida a la salida de un banco de filtros (Mel-warped log-

Filterbank Energies, MFE). Entonces, el objetivo de los experimentos es comparar los métodos de estimación de las matrices de transformación a través de la tasa de error. A efectos de realizar comparaciones, como matriz de transformación de referencia elegimos el basado en la transformada de coseno discreta (MFCC) por ser un clásico en la literatura sobre reconocimiento del habla dados los buenos resultados que se obtienen.

2. DEFINICIÓN DE UNA FUNCIÓN DE COSTE BASADA EN EL ERROR DE BAYES

En general, cualquier regla de decisión que utilicemos para obtener una clasificación no va a dar una clasificación perfecta. Para evaluar el rendimiento de una regla de decisión debemos calcular la probabilidad del error, es decir, la probabilidad de que una muestra sea asignada a la clase equivocada. El error condicional dado una muestra 'x' es aquel en que la probabilidad a posteriori de la clase *i-ésima* dada 'x', $q_1(x)$ o $q_2(x)$, sea el menor. Es decir,

$$r(x) = \min[q_1(x), q_2(x)] \tag{2.1}$$

El error total, llamado error de Bayes, se obtiene calculando la siguiente expresión $E\{r(x)\}$.

$$\begin{aligned} \varepsilon = E\{r(x)\} &= \int r(x)\,p(x)dx \\ &= \int \min[P_1 p_1(x), P_2 p_2(x)]dx \\ &= P_1 \int_{L_2} p_1(x)dx + P_2 \int_{L_1} p_2(x)dx \end{aligned} \tag{2.2}$$

donde $p_i(x) = p(x/w_i)$ $\{i=1,2\}$ y $P_i = P(w_i)$ son la densidad de probabilidad condicional y la probabilidad a priori de la clase *'i'*, respectivamente. La ecuación (2.2) muestra varias formas de expresar el error de Bayes, ε. La primera línea es la definición del ε. La segunda línea se obtiene insertando la ecuación (2.1) en la primera línea y aplicando el teorema de Bayes. Las regiones integrales L_1 y L_2 de la tercera línea son las regiones donde 'x' se clasifica como ω_1 y ω_2 por esta regla de decisión, y se llaman las regiones ω_1 y regiones ω_2. En L_1 $P_1p_1(x) > P_2p_2(x)$ y, por tanto, $r(x) = P_2p_2(x) / p(x)$. Asimismo, $r(x) = P_1p_1(x) / p(x)$ en L_2 porque $P_1p_1(x) < P_2p_2(x)$. Desde un punto de teórico práctico nos centraremos en la segunda línea mientras que desde un punto de vista práctico los haremos sobre la tres pues nos dice que hemos de evaluar el error ε sobre los ejemplos mal clasificados.

Ahora definimos $Y=\{(x^1,y^1),\ldots,(x^N,y^N)\}$ como un conjunto finito de instantes de entrenamiento, donde a cada instante x^t corresponde a la etiqueta $y^t=\{1,2\}$. Definimos A $(k \times m)$ como una matriz lineal que mapea las observaciones originales x^t a uno transformado como $v^t = A^T x^t$, donde x^t es un vector k-dimensional, v^t es un vector m-dimensional y $m \leq k$. La naturaleza de (2.2, línea 2) no es apropiada para obtener una matriz de transformación *'A'* que maximice la separación entre las clases. Como alternativa proponemos una función de coste basada en el error de Bayes (FEB, Función de Error de Bayes) como se muestra a continuación:

$$D = \sum_{t=1}^{N} \tanh\left\{ s \cdot \max_{\alpha, A}\left[-Ln(P_1 p_1(v^t)), -Ln(P_2 p_2(v^t)) \right] \right\} \tag{2.3}$$

donde la integral ha sido sustituida por un sumatorio porque el conjunto de entrenamiento es finito, se ha introducido la tangente hiperbólica por conveniencia y $s>0$ es un factor constante que controla el rango dinámico del argumento. Observar que en (2.3) utilizamos el máximo de $-Ln(\cdot)$, que es equivalente a buscar $min[P_1p_1(x),P_2p_2(x)]$ en (2.2). La función resultante es cóncava creciente y facilita la búsqueda de un mínimo error entre clases mediante búsqueda de un máximo de (2.3). Esta búsqueda se hará con un método de paso ascendente que será presentado en la sección 4.

3. FUNCIÓN DE COSTE BASADA EN LA DISTANCIA DE CHERNOFF

La distancia de Chernoff es una medida de similitud entre dos funciones de densidad de probabilidad (pdf). Por ejemplo, cada pdf podría definir la probabilidad de pertenecer a una clase dada, y por tanto significa cuan similar o diferente son las dos clases. La distancia se puede definir como sigue [5]:

$$\overline{D} = \max_{0 \leq \alpha \leq 1}\left\{ -Ln\left(\int P_1^{\alpha} p_1^{\alpha}(x) P_2^{1-\alpha} p_2^{1-\alpha}(x)dx \right) \right\} \tag{3.1}$$

Obviando la maximización en (3.1) con respecto a α, y sin pérdida de generalidad, se asume que es una constante de ahora en adelante y es igual a ½ en los experimentos. Para nuestro propósito, el aspecto importante es que cuanto mayor sea la distancia $\overline{D}$ entre las dos distribuciones será la probabilidad de error de clasificación entre las clases. De hecho, a menudo la expresión (3.1) se usa para obtener una frontera superior en la probabilidad de error de clasificación tal que a mayor distancia, menor probabilidad.

La naturaleza convexa de (3.1) debido a la aplicación del $-Ln[.]$ sobre la integral no es apropiada para obtener la matriz de transformación *'A'*, por lo que usaremos la siguiente función de coste [7] (FEC, Función de Error de Chernoff):

$$D = \max_{\alpha, A} \tanh\left\{ -s \cdot Ln\left[\sum_{t=1}^{N} P_1^{\alpha} p_1^{\alpha}(v^t) P_2^{1-\alpha} p_2^{1-\alpha}(v^t) \right] \right\} \tag{3.2}$$

donde el sumatorio, la tangente hiperbólica y $s>0$ tienen el mismo significado de la sección anterior. Téngase en cuenta que ahora cuando las pdf de las dos clases tiendan a solaparse, $\overline{D}$ y D tienden a cero pero cuando el solapamiento disminuye entonces D tiende a uno. La

función resultante es cóncava creciente y facilita la búsqueda de mínimo solapamiento (confusión) entre clases mediante búsqueda de un máximo. Esta búsqueda se hará con un método de paso ascendente que será presentado en la siguiente sección.

4. MAXIMIZACIÓN DE LAS FUNCIONES DE COSTE

Vamos a particularizar para el caso en el que cada probabilidad $p_j(x)$ de la clase $j=\{1,2\}$ se representa por una mezcla de M Gausianas con medias μ_j^i, matrices de covarianzas Σ_j^i, factores de peso ω_j^i y $\{i=1,\ldots,M\}$. El objetivo es obtener la matriz de transformación A tal que la función de coste se minimiza o maximiza según proceda ya que está asociada a un error de clasificación mínimo. Empezamos por realizar la derivada parcial de (2.3) con respecto de A. Aplicando la regla de la cadena obtenemos

$$\frac{\partial D}{\partial A} = s\sum_{t=1}^{N} 1-\tanh^2\left[-s\cdot Ln\left(p(v^t)\right)\right]\left[\frac{\partial}{\partial A}\left(-Ln\left(p(v^t)\right)\right)\right] \quad (4.1)$$

donde $p(v^t)$ es la función de densidad de probabilidad mayor obtenida de la ecuación (2.3) dado v^t.

Para la función de coste basada en la distancia de Chernoff realizamos los mismos pasos. Aplicando la regla de la cadena a (3.2) con respecto de A obtenemos

$$\frac{\partial D}{\partial A} = s\left[1-\tanh^2\left(-s\cdot Ln\left(\sum_{t=1}^{N} p_1^{\alpha}(v^t)p_2^{1-\alpha}(v^t)\right)\right)\right]\left[\frac{\partial}{\partial A}\left(-Ln\left(\sum_{t=1}^{N} p_1^{\alpha}(v^t)p_2^{1-\alpha}(v^t)\right)\right)\right] \quad (4.2)$$

Ahora vamos a desarrollar en más detalle la derivada parcial en el lado derecho de las ecuaciones (4.1) y (4.2). Tomando de nuevo la regla de la cadena y derivando el logaritmo de esta forma $\partial Ln[g(x)]/\partial x = g'(x)\,\partial Ln[g(x)]/\partial[g(x)]$, el segundo factor entre corchetes del lado derecho de la ecuación (4.1) se puede igualar a

$$\frac{\partial}{\partial A}\left(-Ln\left(p(v^t)\right)\right) = \frac{\partial/\partial A(p(v^t))}{p(v^t)} \quad (4.3)$$

Y el lado derecho de la ecuación (4.2) se puede igual a

$$\frac{\partial}{\partial A}\left(-Ln\left(\sum_{t=1}^{N} p_1^{\alpha}(v^t)p_2^{1-\alpha}(v^t)\right)\right) = \sum_{t=1}^{N}\left[\frac{\partial/\partial A(p_1^{\alpha}(v^t)p_2^{1-\alpha}(v^t))}{\sum_{t=1}^{N} p_1^{\alpha}(v^t)p_2^{1-\alpha}(v^t)}\right] \quad (4.4)$$

donde el denominador en parte derecha de las ecuaciones (4.3) y (4.4) actúa como peso normalizador de la contribución de la derivada para cada vector de entrenamiento x^t. Debido a que estas derivadas tienen muchos términos y algunos de ellos son numéricamente insignificantes tomaremos la simplificación de tomar en cuenta solo los componentes más importantes de cada mezcla. Así, sólo se considera una mezcla por clase en (4.3) y (4.4).

La expresión de la componente *'i'* de la clase *'j'* en el espacio transformado se puede expresar como:

$$p_j^i(x^t, A) = \frac{1}{(2\pi)^{n/2}\left|A^T\Sigma_j^i A\right|^{1/2}}\exp\left\{-\frac{1}{2}\left(x^t-\mu_j^i\right)^T A\left(A^T\Sigma_j^i A\right)^{-1}A^T\left(x^t-\mu_j^i\right)\right\} \quad (4.5)$$

Por ejemplo, en el caso particular de la clase $j=1$, la derivada parcial con respecto a la matriz de transformación 'A' de un componente dado *'i'* es:

$$\frac{\partial p_1^{i\,\alpha}(A)}{\partial A} = p_1^{i\,\alpha}(A)\left[\alpha\cdot B_1^i A-\alpha\cdot\Sigma_1^i A\left(A^T\Sigma_1^i A\right)^{-1}\left(A^T B_1^i A\right)+\Sigma_1^i A\left(A^T\Sigma_1^i A\right)^{-1}\right] \quad (4.6)$$

donde $B_j^i = A^T\left(x^t-\mu_j^i\right)\left(x^t-\mu_j^i\right)^T A$. Note que (4.6) se calcula para cada vector de entrenamiento x^t y, por tanto, cada uno de ellos influirá directamente en el cálculo de A.

La matriz de transformación se podría obtener resolviendo la ecuación de igualar a cero la ecuación (4.1) o (4.2), lo cual es difícil. En cambio, aplicaremos un método de paso ascendente para encontrar la matriz A, p.e., dada una matriz inicial A^0 y una matriz actualizada $A^{(l)}$ de la siguiente manera:

$$A^{(l+1)} = A^{(l)} + \gamma_l \left.\frac{\partial D}{\partial A}\right|_{A=A^{(l)}} \quad (4.7)$$

para $\{l=1,2,\ldots, N_l\}$ donde *'l'* es el índice de iteración, N_l es el número de iteraciones, $\gamma_l = \gamma_0\,(1-l/N_l)$ es el tamaño del paso que depende del paso y γ_0 es la inicialización de una constante que ha sido puesta a 0'1 en los experimentos.

Ahora podemos tomar algunas consideraciones. El factor $(1-\tanh^2[.])$ en (4.1) y (4.2) juega un papel interesante porque es un reflejo de la forma en que la función de coste progresa hacia un máximo cuando se lleva a cabo una nueva iteración. Así, este factor tiende a cero conforme se avanza hacia la convergencia. El producto entre el factor de suavizado *'s'* y el tamaño del paso γ_l actúa como prevención de un progreso rápido hacia un máximo.

La forma en que se inicializa la matriz A^0 está abierta. En los métodos iterativos la inicialización es muy importante porque afecta a la evolución hacia un máximo global o local. En este artículo exponemos aquella inicialización de la matriz de transformación que mejores resultados nos ha dado [6]. La matriz A que maximiza la divergencia entre las clases se puede construir seleccionando los m autovectores de la matriz $\Sigma_2^{-1}\Sigma_1$ correspondiente a los m mayores autovalores, donde Σ_i es la matriz de covarianza de las Gausianas más cercanas entre sí.

4. EXPERIMENTOS Y RESULTADOS

Para los experimentos de clasificación utilizamos habla con y sin patología. Usamos la base de datos Disordered Voice Database Model 4337 compuesta de 54 locutores sin patología y 608 locutores con patología. Las grabaciones consisten en la vocal sostenida /ah/. La mitad de la base de datos se usó para el entrenamiento y la otra mitad para el test. Las características originales MFE fueron obtenidas aplicando m=20 filtros triangulares dando lugar a una dimensión de 20 en el espacio original. La dimensionalidad del espacio transformado es k=10. La forma de onda está muestreada a 25 KHz, se toman tramas de 30 msec. Con un solape de 20 msec. entre bloques adyacentes. Cada trama se pasa a través de un filtro pre-énfasis con un factor de 0'95 y se aplica una ventana Hamming. Entonces, se hace una FFT de 2048 puntos para producir 1024 puntos del espectro de energía. A este se le aplican los *'m'* filtros triangulares para obtener la energía logarítmica. La totalidad del conjunto de vectores de entrenamiento se usa para representar cada clase en el espacio original mediante una mezcla de Gausianas con M=7 componentes. A efectos de comparación hemos realizado experimentos en el espacio original (MFE) y en el espacio transformado con MFCC, FEB y FEC con la inicialización introducidas en la sección 4. Hemos encontrado que s=1/32 es una buena elección como factor de control en (2.3). En la tabla 1 mostramos los resultados de clasificación para los cuatro métodos mencionados en la sección anterior. Como puede verse, ambas versiones del método que proponemos mejora a MFE y MFCC.

Method	Scores
MFE	*86.10%*
MFCC	*83.69%*
FEB	*93.35%*
FEC	*95.77%*

Tabla 1. Resultados de clasificación para diferentes métodos

Es interesante observar que la transformación MFCC no mejora los resultados de clasificación en comparación con MFE como se podría esperar de la literatura. Los resultados obtenidos a partir de nuestras transformaciones son realmente mejores.

Metodo	Patología	NP	HP
MFE	NP	*74.07%*	*25.93%*
MFE	*HP*	*12.83%*	*87.17%*
MFCC	NP	*77.78%*	*22.22%*
MFCC	*HP*	*15.79%*	*84.21%*
FEB	NP	*62.96%*	*37.04%*
FEB	*HP*	*3.95%*	*96.05%*
FEC	NP	*77.78%*	*22.22%*
FEC	*HP*	*2.63%*	*97.37%*

Tabla 2. Matrices confusión para diferentes métodos

Ya que la cantidad de ejemplos para el habla no patológica es pequeña, en la tabla 2 vemos la matriz de confusión de los diferentes métodos entre habla patológica (HP) y no patológica (NP).

Podemos ver que el método de Chernoff ofrece los menores resultados de confusión incluso para la clase NS que tiene menos cantidad de ejemplos comparados con la clase PS.

5. CONCLUSIONES

Hemos formulado dos funciones de coste basada en el error de Bayes y hemos dado la formula para una maximización iterativa. La versión simplificada presentada aquí tiene menos demanda computacional que el original mientras que da unos resultados de clasificación buenos. Obviando simplificaciones, desde el punto de vista teórico, en este artículo abrimos el ámbito de las posibilidades de obtener matrices de transformación, pero además todos los parámetros que caracterizan las mezclas de componente de cada clase. Este y otros temas relacionados se tratarán en el trabajo futuro en test de hipótesis binaria y M-aria con un énfasis especial en extender la formulación a modelos ocultos de Markov.

6. REFERENCIAS

[1] L.R. Bahl, P.F. Brown, P.V. Souza, & R.L. Mercer, "Maximum mutual information estimation of hidden Markov models for speech recognition". Proc ICASSP'86, pp 49-52, 1986.

[2] P.C. Chang & B-H. Juang, "Discriminative training of dynamic programming based speech recognizers". IEEE Trans. Speech Audio Processing, Volume 1, number 2, pp 135-143, 1993.

[3] P.C. Loizou, & A.S. Spanias, "Improved speech recognition using a subspace projection approach". IEEE Transactions on Speech and Audio Processing, Volume: 7, Issue: 3, pp 343–345, 1999.

[4] R. Haeb-Umbach & H. Ney, "Linear discriminant analysis for improved large vocabulary continuous speech recognition". Proc ICASSP'92, pp 13-16, 1992.

[5] Fukunaga, K, "*Introduction to Statistical Pattern Recognition*". Academic Press, Inc. 2nd Edition, 1990.

[6] P.C. Loizou, & A.S. Spanias, "High-performance alphabet recognition". IEEE Transactions on Speech and Audio Processing, Volume: 4, pp 430-445, 1996.

[7] F.D. Lorenzo, A.G. Ravelo, J.L. Navarro, S.I. Martín, P.J. Quintana, E. Hernández. "A Chernoff-based Approach to the Estimation of Transformation Matrices for Binary Hypothesis Testing". IEEE-ICASSP, pp V-753 – V-756

ESTIMACIÓN DE PROBABILIDADES A POSTERIORI EN SVMS MULTICLASE PARA RECONOCIMIENTO DE HABLA CONTINUA

Rubén Solera Ureña, Fernando Pérez-Cruz, Fernando Díaz de María

Universidad Carlos III de Madrid
Departamento de Teoría de la Señal y Comunicaciones
Avda. de la Universidad, 30, 28911-Leganés (Madrid), ESPAÑA

RESUMEN

El presente artículo aborda el problema de la estimación de probabilidades en el marco del reconocimiento automático de habla mediante Máquinas de Vectores Soporte (SVMs). Aunque las SVMs están pensadas para tareas de clasificación, en ciertas aplicaciones, especialmente problemas multiclase, es conveniente conocer el nivel de incertidumbre asociado a la decisión del clasificador o la probabilidad a posteriori de cada una de las etiquetas posibles. Durante los últimos años se han desarrollado diversos métodos para estimar dichas probabilidades a partir de las salidas blandas de las SVMs; sin embargo, la base teórica subyacente no está suficientemente justificada. En este artículo se revisan las debilidades de algunos de estos métodos y se presentan diversas alternativas basadas en un uso más directo de las salidas de las SVMs.

1. INTRODUCCIÓN

Las Máquinas de Vectores Soporte [1] constituyen el estado del arte en tareas de clasificación no lineal. Sus ventajas frente a otros clasificadores tradicionales han llamado la atención de numerosos investigadores en diversas áreas, entre ellas la de reconocimiento automático de habla. Fundamentalmente, hay tres razones que han conducido a plantear el uso de las SVMs como posible alternativa a los Modelos Ocultos de Markov (HMMs) en reconocimiento de voz: a) el modelado acústico es fundamentalmente un problema de clasificación de patrones, mientras que los HMMs son modelos generativos y, por tanto, lo que buscan es estimar las funciones de densidad de probabilidad correspondientes a cada uno de los modelos; b) la función de coste definida por las SVMs establece un compromiso entre la minimización del riesgo empírico y del riesgo estructural, lo que produce máquinas con una mayor capacidad de generalización frente a otros clasificadores tradicionales; c) el entrenamiento de las SVMs busca la maximización del margen, definido como la distancia de las muestras de entrenamiento a la frontera de decisión, lo que a priori les hace más robustas frente a condiciones ruidosas, uno de los principales problemas en reconocimiento de habla.

No obstante, la aplicación de las SVMs en reconocimiento de habla no es inmediata, debido principalmente a que estas máquinas trabajan con vectores de entrada de dimensión fija. Por el contrario, los métodos más comunes de parametrización de la voz producen secuencias de parámetros de longitud variable, dependiendo de la duración de la locución. En los últimos años han aparecido diversas formas de abordar este problema. El sistema de reconocimiento de habla continua mediante SVMs que se presenta en [2] lo evita trabajando trama a trama. El reconocedor propuesto emplea una SVM multiclase para determinar a qué clase pertenece el segmento de voz considerado y, a continuación, se usa el algoritmo de Viterbi para obtener una secuencia de palabras a partir de las decisiones acústicas que proporciona la SVM.

Como es sabido, los HMMs contribuyen al proceso de reconocimiento de la secuencia de parámetros espectrales de entrada aportando la verosimilitud de que la muestra actual haya sido generada por cada uno de los modelos que constituyen el sistema de reconocimiento de voz (palabras, fonemas, trifonemas...). Las SVMs, en cambio, proporcionan la etiqueta de la clase asignada al vector de entrada. Para aquellas aplicaciones en las que el clasificador sólamente se encarga de una parte de la decisión global, se han propuesto diversas formas para estimar probabilidades *multiclase* a partir de las salidas blandas de las SVMs [3, 4]. Su planteamiento depende principalmente de la estrategia multiclase que se adopte en la SVM (*1-contra-1* ó *1-contra-el resto*), las cuales se detallarán en la siguiente sección. Una de las más usadas, empleada en [2], se basa en el cálculo de la probabilidad de Platt [5] para cada SVM binaria (en la aproximación *1-contra-1*) y el empleo de una variante del método de Refregier-Vallet para obtener las probabilidades *multiclase* [3].

No obstante, el fundamento teórico de los métodos citados no está suficientemente justificado, tal y como muestran los experimentos realizados. Su principal debilidad consiste en la hipótesis de gaussianidad en las funciones de densidad de probabilidad condicional de la salida de las SVMs binarias, dada una cierta clase ($p(f|y=+1)$ y $p(f|y=-1)$, donde f denota la salida blanda de la SVM binaria para la muestra actual y $+1$ y -1 son las etiquetas asociadas a las dos clases consideradas). En este artículo se exploran diversas alternativas basadas en el uso directo de las salidas blandas de las SVMs, evitando en lo posible

hipótesis como la señalada anteriormente.

El artículo está estructurado de la siguiente forma: la sección 2 presenta los conceptos básicos de las Máquinas de Vectores Soporte y repasa las formas más comunes de estimar probabilidades *multiclase* a partir de las salidas de las SVMs. En la sección 3 se presentan nuestras propuestas para estimar las probabilidades a posteriori a partir de la salida blanda de las SVMs. En la sección 4 se describe a grandes rasgos el sistema de reconocimiento de habla empleado en los experimentos ([2]) y se muestran los resultados obtenidos sobre una tarea de reconocimiento de dígitos conectados con el fin de evaluar las prestaciones de los métodos propuestos.

2. MÁQUINAS DE VECTORES SOPORTE

2.1. Fundamentos de las SVMs

Una SVM es un clasificador binario que asigna una etiqueta $y \in \{+1, -1\}$ al vector de entrada $\mathbf{x}$ conforme al signo de la siguiente expresión:

$$f(\mathbf{x}) = \mathbf{w}^T \cdot \phi(\mathbf{x}) + b, \tag{1}$$

donde $\phi(\mathbf{x}) : \Re^d \mapsto \Re^H$ ($d << H$) es una transformación del espacio de entrada a un espacio *de características* de mayor dimensión (incluso infinita), en el que se supone que las clases son linealmente separables. El vector $\mathbf{w}$ define el hiperplano de separación en dicho espacio y b representa el sesgo respecto al origen de coordenadas.

La razón que hace a las SVMs más robustas que otros clasificadores es su criterio de entrenamiento, consistente en un compromiso entre la minimización del riesgo empírico y del riesgo estructural. Éste último evita un posible sobreajuste de la máquina al conjunto de entrenamiento. La solución viene dada por el siguiente problema de minimización cuadrática:

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i, \\ \text{sujeto a} \quad & y_i(\mathbf{w}^T \cdot \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0;\ i = 1, \cdots, n, \end{aligned} \tag{2}$$

donde $\mathbf{x}_i \in \Re^d$ ($i = 1, \ldots, n$) son las muestras de entrenamiento con etiquetas $y_i \in \{+1, -1\}$. Las variables ξ_i miden el error de entrenamiento de cada muestra y C es un factor de ponderación entre el riesgo empírico y el riesgo estructural. Para más detalles sobre la SVM se puede consultar [6, 1, 7].

El problema de programación cuadrática en (2) se suele resolver empleando el dual de Wolfe [8], en el que los multiplicadores de Lagrange α_i se calculan maximizando:

$$\begin{aligned} \max_{\alpha_i} \quad & \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j) \\ \text{sujeto a} \quad & \sum_{i=1}^{n}\alpha_i y_i = 0,\ 0 \leq \alpha_i \leq C \end{aligned} \tag{3}$$

Este problema es cuadrático y convexo, por lo que la convergencia al mínimo global está asegurada. Una vez resuelto, el vector de pesos $\mathbf{w}$ se puede expresar de la siguiente forma:

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \phi(\mathbf{x}_i). \tag{4}$$

Sólo aquellas muestras cuyo multiplicador asociado α_i sea distinto de 0 contibuyen a la definición de la frontera de decisión, razón por la que reciben el nombre de *vectores soporte*.

Normalmente, la función $\phi(\mathbf{x})$ no se conoce de forma explícita o es imposible de evaluar. No obstante, esto no plantea ninguna dificultad, ya que si nos fijamos en las expresiones (1) y (3) veremos que únicamente se precisa calcular los productos escalares $\phi^T(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$, los cuales, empleando lo que se ha denominado *truco del kernel* [1], se pueden evaluar mediante la función de *kernel* $K(\mathbf{x}_i, \mathbf{x}_j)$. De esta forma, la salida blanda de la SVM adopta la siguiente expresión:

$$f(\mathbf{x}) = \sum_{i=1}^{n}\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b. \tag{5}$$

Los *kernels* más comunes son el lineal (K_L):

$$K_L(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \cdot \mathbf{x}_j \tag{6}$$

y el gaussiano (K_{RBF}):

$$K_{RBF}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right). \tag{7}$$

Al comienzo de este apartado se indicó que una SVM es, en principio, un clasificador binario. Existen algunas aproximaciones al problema multiclase que reformulan las expresiones anteriores para considerar todas las clases a la vez [9, 10]. Su elevado coste computacional ha llevado, no obstante, a emplear combinaciones de SVMs binarias para abordar el caso multiclase. Existen básicamente dos versiones. La primera, denominada *1-contra-el resto*, consiste en comparar cada clase con todas las demás, mientras que en la segunda versión cada clase se compara con las restantes de forma separada (*1-contra-1*). Aunque en este último caso el número de SVMs binarias es mayor ($\frac{k(k-1)}{2}$ frente a k SVMs, siendo k el número de clases), en [2] se adopta esta solución dado que el menor número de vectores de entrenamiento en cada SVM conduce a un menor coste computacional con prestaciones similares [11]. A continuación se aplica un proceso de votación entre todas las SVMs binarias para decidir la etiqueta correspondiente a la muestra de entrada.

2.2. Estimación de probabilidades en SVMs

Dependiendo de la solución multiclase que se adopte, las SVMs binarias pueden comparar dos clases entre sí o

bien una de ellas con todas las demás. No obstante, por lo general el proceso de obtención de probabilidades a posteriori a partir de las salidas de las SVMs consta en ambos casos de dos pasos: 1) obtener la probabilidad de que la muestra pertenezca a cada clase en todas las SVMs binarias, y 2) transformar estas probabilidades *binarias* a probabilidades *multiclase*. En la aproximación *1-contra-el resto*, este último paso puede reducirse a una simple normalización para que la suma de las probabilidades a posteriori sea uno, ya que el número de SVMs binarias coincide con el número de clases.

La forma más comúnmente empleada para transformar la salida de una SVM en probabilidades *binarias* consiste en el uso de una función sigmoide. Su origen está en [12], donde se propone ajustar mediante sendas gaussianas las funciones de densidad de probabilidad condicional de la salida de la SVM ($p(f|y=+1)$ y $p(f|y=-1)$. Aplicando la regla de Bayes:

$$P\left(y=1|f\right)=\frac{p\left(f|y=1\right)P\left(y=1\right)}{\sum\limits_{i=\pm1}p\left(f|y=i\right)P\left(y=i\right)} \tag{8}$$

y sustituyendo dichas expresiones para las funciones de densidad de probabilidad condicional se llega a:

$$P\left(y=1|f\right)=\frac{1}{1+exp\left(af^2+bf+c\right)}. \tag{9}$$

Para simplificar esta función y evitar que no sea monótona, en [12] se asume que las gaussianas mencionadas están centradas en los márgenes (±1) y tienen la misma varianza, la cual debe estimarse. En este caso, la expresión de la probabilidad a posteriori se simplifica en una sigmoide, cuya pendiente en la zona lineal se calcula a partir de la varianza de las gaussianas. El sesgo se calcula de forma que $P\left(y=1|f\right)=0{,}5$ en $f=0$.

Basándose en este trabajo y asumiendo que en la zona comprendida entre los márgenes las funciones de densidad de probabilidad condicional son aproximadamente exponenciales, Platt propone un modelo paramétrico para la probabilidad *binaria* a posteriori [5]:

$$P\left(y=1|f\right)=\frac{1}{1+\exp\left(Af+B\right)}. \tag{10}$$

La diferencia respecto al trabajo anterior consiste en que los parámetros A y B se estiman de manera discriminativa maximizando la verosimilitud.

Esta expresión proporciona directamente probabilidades *multiclase* en el caso *1-contra-el resto*. Si se tienen k clases distintas, la probabilidad a posteriori de la clase *i-ésima* se puede obtener como:

$$P\left(y=i|\mathbf{x}\right)=\frac{1}{1+exp\left(A_if_i\left(\mathbf{x}\right)+B_i\right)}, \tag{11}$$

siendo $f_i\left(\mathbf{x}\right)$ la salida de la SVM binaria que clasifica la clase i contra el resto. En este caso no se garantiza que la suma de las probabilidades sea uno; una forma de obtener probabilidades a posteriori normalizadas consiste en usar la función *softmax*:

$$P\left(y=i|\mathbf{x}\right)=\frac{exp\left(\gamma f_i\left(\mathbf{x}\right)\right)}{\sum\limits_{j=1}^{k}exp\left(\gamma f_j\left(\mathbf{x}\right)\right)}, \tag{12}$$

donde el parámetro γ se estima maximizando la verosimilitud.

En el caso *1-contra-1*, en primer lugar se debe calcular la probabilidad de Platt para la muestra de entrada en cada SVM binaria $\left(i,j\right),\forall i,j\in\{1,...,k\}$. La probabilidad de Platt de que **x** pertenezca a la clase i en la SVM binaria (i,j) se calcula como:

$$r_{ij}=P\left(y=i|\text{y=i ó j},\mathbf{x}\right)=\frac{1}{1+exp\left(A_{ij}f_{ij}(\mathbf{x})+B_{ij}\right)},$$

$$r_{ji}=P\left(y=j|\text{y=i ó j},\mathbf{x}\right)=1-r_{ij}(\mathbf{x}), \tag{13}$$

siendo $f_{ij}\left(\mathbf{x}\right)$ la salida de la SVM binaria (i,j).

A continuación, se emplea una modificación del método de Refregier-Vallet para transformar estas probabilidades *binarias* $r_{ij}\ \forall\ i,j$ en probabilidades *multiclase* $P(y=i|\mathbf{x})\ \forall\ i$. En [13] se propone resolver un sistema lineal formado por $k-1$ ecuaciones del tipo:

$$r_{ji}P\left(y=i|\mathbf{x}\right)=r_{ij}P\left(y=j|\mathbf{x}\right), \tag{14}$$

junto con otra que fuerce que la suma de las probabilidades sea uno. En [3] se señala que la solución que se obtiene depende en gran medida de las ecuaciones seleccionadas, por lo que se propone como alternativa el siguiente problema de minimización, que considera todas las ecuaciones posibles:

$$\min_{\mathbf{p}}\quad\frac{1}{2}\sum_{i=1}^{k}\sum_{j:j\neq i}^{k}\left(r_{ji}P\left(y=i|\mathbf{x}\right)-r_{ij}P\left(y=j|\mathbf{x}\right)\right)^2,$$

$$\text{sujeto a}\quad\sum_{i=1}^{k}P\left(y=i|\mathbf{x}\right)=1,$$

$$P\left(y=i|\mathbf{x}\right)\geq0\ \forall i, \tag{15}$$

con $\mathbf{P}(\mathbf{x})=\left[P\left(y=1|\mathbf{x}\right),...,P\left(y=k|\mathbf{x}\right)\right]$.

Finalmente, en [14] se propone una forma alternativa para obtener las probabilidades a posteriori *multiclase* cuando la probabilidad a priori es la misma para todas las clases:

$$P(y=i|\mathbf{x})=\frac{\prod\limits_{j:j\neq i}r_{ij}}{\sum\limits_{m=1}^{k}\prod\limits_{j:j\neq m}r_{mj}} \tag{16}$$

3. NUEVA ESTIMACIÓN DE PROBABILIDADES NO PARAMÉTRICAS

Tanto el trabajo presentado en [12] como el método de Platt asumen un comportamiento exponencial, ya

sea globalmente o de forma local entre ambos márgenes, en las funciones de densidad de probabilidad condicional de la salida de las SVMs binarias ($p(f|y=+1)$ y $p(f|y=-1)$). Sin embargo, normalmente no se cumple esta condición, por lo que la justificación teórica del uso de una sigmoide para obtener las probabilidades *binarias* no está suficientemente fundamentada. Otro problema, de carácter práctico, es que los parámetros de la sigmoide se estiman maximizando la verosimilitud de las muestras de entrenamiento. El procedimiento habitual para encontrar los mejores parámetros consiste en un proceso de validación cruzada en cada SVM binaria, por lo que el tiempo de entrenamiento de la SVM multiclase se multiplica aproximadamente por el número de grupos empleados en la validación. Véase que el gran tamaño de las bases de datos empleadas en habla y la elevada superposición de las clases consideradas (que produce máquinas con un gran número de vectores soporte) hacen del entrenamiento de la SVM multiclase un proceso costoso computacionalmente. El método de Platt produce un aumento considerable de este coste, a la vez que carece de una justificación teórica sólida. No obstante, hay que señalar que la función sigmoide resulta una manera intuitiva y directa de transformar la salida de la SVM, definida $\forall\ \Re$, en un valor asimilable a una probabilidad $\in [0, ..., 1]$.

Respecto al método de Refregier-Vallet modificado empleado en [3] para obtener las probabilidades *multiclase*, hay que señalar que su formulación produce que la probabilidad a posteriori de una clase dependa de la de todas las demás y, a través de ellas, de todas las probabilidades *binarias*. Es decir, en el cálculo de la probabilidad $P(y=i|\mathbf{x})$ intervienen las probabilidades $r_{jm}(\mathbf{x})$, $\forall\ j,m \in \{0, ..., k\}$. Esto significa que en el cálculo de la probabilidad a posteriori de una clase se están teniendo en cuenta decisiones de máquinas binarias en las que no participa, lo que a priori no resulta lógico.

Por todas estas razones, se han estudiado diversos métodos alternativos centrados en torno a tres objetivos principales: a) obviar en la medida de lo posible la hipótesis de gaussianidad en las funciones de densidad de probabilidad condicional de la salida de las SVMs; b) emplear únicamente la salida de las SVMs binarias en las que participa la clase considerada; c) reducir el coste computacional que introduce el proceso de estimación de probabilidades, tanto en el entrenamiento como en el test.

En reconocimiento de habla, el algoritmo de Viterbi suele trabajar con la suma de los logaritmos decimales de las probabilidades y verosimilitudes para evitar problemas numéricos. Así, la log-verosimilitud que proporcionan los Modelos Ocultos de Markov podría sustituirse por cualquier otra medida indicativa de la pertenencia de la muestra a cada uno de los modelos. Una medida de este tipo puede ser la distancia a la frontera de decisión (margen): a medida que crece podemos estar razonablemente más seguros de que la muestra pertenece a la clase considerada. Esta idea es la que se usa en las dos primeras alternativas que proponemos: emplear el margen en el Viterbi como medida directa de la pertenencia de la muestra a cada clase o modelo. El problema que aparece cuando usamos la versión multiclase *1-contra-1* es que no disponemos de un solo margen por clase, sino de los $k-1$ correspondientes a las SVMs binarias en las que está presenta la clase. La solución que proponemos es calcular un margen *multiclase* o *acumulado* como la suma de los márgenes en las SVMs binarias en las que participa la clase (método *Directo 1* $[P_{D1}]$).

$$P_{D1}(y=i|\mathbf{x}) = \sum_{j:j\neq i} f_{ij}(\mathbf{x}). \tag{17}$$

Cuando el número de clases es elevado, el valor absoluto de la suma de los márgenes puede dispararse, por lo que también se considera el uso del valor medio del margen (método *Directo 2* $[P_{D2}]$).

$$P_{D2}(y=i|\mathbf{x}) = \frac{P_{D1}(y=i|\mathbf{x})}{k}. \tag{18}$$

Como se muestra en el apartado 4.2, estos dos métodos no proporcionan buenos resultados, quizás debido a que el Viterbi empleado está optimizado para su uso habitual en reconocimiento de habla. Si la escala de los logaritmos de las verosimilitudes difiere en gran medida de la de los márgenes *acumulados*, podría suceder que se desajustase la ponderación entre el modelado acústico y el modelado del lenguaje, produciendo peores resultados. Para obtener valores asimilables a probabilidades en el rango $[0, ..., 1]$ se decidió usar la función *softmax* sobre las sumas de los márgenes (método *softmax* $[P_S]$):

$$P_S(y=i|\mathbf{x}) = \frac{exp\left(\gamma P_{D1}(y=i|\mathbf{x})\right)}{\sum_{j=1}^{k} exp\left(\gamma P_{D1}(y=j|\mathbf{x})\right)}, \tag{19}$$

siendo γ una constante normalizadora. Se puede observar que esta solución es la versión *1-contra-1* de la expresión (12).

La última de nuestras propuestas consiste en calcular las probabilidades de Platt para cada SVM y aplicar a continuación el método de Refregier-Vallet modificado, pero usando en este caso la misma función sigmoide en todas las SVMs binarias (método de *Platt modificado* $[r_{ij_M}]$):

$$r_{ij_M}(\mathbf{x}) = \frac{1}{1+exp\left(Af_{ij}(\mathbf{x})\right)}. \tag{20}$$

Esta expresión es similar a la (13), pero ahora el parámetro A es una constante negativa que no depende la SVM binaria y $B=0$. Este valor para el sesgo tiene sentido, ya que cualquier otro implica un desplazamiento de la frontera de decisión de la SVM binaria. Tanto en este método como en el anterior se emplea el logaritmo de la probabilidad estimada, mientras que en los métodos *directos* se usa la suma de los márgenes.

4. EXPERIMENTOS

4.1. Descripción del sistema

El sistema de reconocimiento de habla continua presentado en [2] se basa en el uso de una SVM multiclase que clasifica cada segmento de voz; como se ha explicado, se trata de un clasificador blando, ya que en realidad proporciona una estimación de la probabilidad a posteriori de cada clase. A continuación, un Viterbi se encarga de obtener una secuencia de palabras a partir de las decisiones acústicas de la SVM.

Por simplicidad, se aborda una tarea de reconocimiento de dígitos conectados en castellano; no obstante, el sistema es fácilmente extensible al reconocimiento de habla continua. Como unidades básicas se consideran tres clases por fonema, correspondientes a la parte inicial, la parte estable y la parte final del mismo. Se ha comprobado que esta forma de definir las clases (en vez de una sola por fonema) mejora las prestaciones del sistema, ya que se tiene en cuenta en mayor medida la variabilidad temporal del habla. Para los dígitos en castellano se definen 17 fonemas más el silencio, por lo que en total se tendrán $18 \cdot 3 = 54$ clases.

El software empleado para el entrenamiento de las SVMs, tanto en el sistema de referencia como en los experimentos que presentamos en el apartado 4.2, es *LIBSVM* [15]. Este programa implementa la solución multiclase *1-contra-1*, lo que en este caso implica entrenar y usar $\frac{54 \cdot 53}{2} = 1431$ SVMs binarias. Se usa un *kernel* gaussiano RBF, con parámetros C y σ obtenidos mediante validación cruzada. La estimación de las probabilidades a posteriori se basa en el cálculo de las probabilidades de Platt y el método de Refregier-Vallet modificado. Para la implementación del algortimo de Viterbi se ha empleado el software HTK [16].

La base de datos usada es SpeechDat [17]. Dispone de 4000 locutores, de los cuales se han empleado 3496 para entrenamiento (con un total de 71000 ficheros) y 350 para test. Para limitar la duración de los experimentos, en la fase de test se han utilizado únicamente 700 de los ficheros con dígitos conectados, disponiéndose de un total de 6546 palabras.

La parametrización de la voz consiste en 12 coeficientes MFCCs y la energía, y sus correspondientes diferencias primeras y segundas. En total, se obtienen 39 parámetros por muestra, con un periodo de muestreo de 10 ms. En experimentos anteriores se ha comprobado que la normalización de los datos es fundamental cuando se trabaja con Máquinas de Vectores Soporte, por lo que se ha realizado una normalización en media y varianza.

4.2. Resultados

La tabla 1 muestra los resultados obtenidos para cada uno de los modelos que se han estudiado: cálculo de las probabilidades de Platt seguido del método de Refregier-Vallet modificado, métodos directo 1 y 2, función *softmax* sobre los márgenes acumulados, y método de Platt modificado, para distintos valores de la pendiente de la sigmoide en la zona lineal. Se indican los casos en los que se utiliza el logaritmo decimal de la probabilidad estimada. Para el método *softmax* se han probado diversos valores de γ, obteniéndose el mejor resultado para un valor de $\gamma = 0{,}02$.

Método	**Tasa de reconocimiento de palabras (%)**
Platt+R-V (log)	$\mathbf{95{,}05}$ $[94{,}52 - 95{,}58]$
Directo 1	$90{,}65$ $[89{,}94 - 91{,}36]$
Directo 2	$91{,}60$ $[90{,}93 - 92{,}27]$
Softmax ($\gamma = 0{,}02$) (log)	$91{,}63$ $[90{,}96 - 92{,}30]$
Platt modificado+R-V ($A = -0{,}5$) (log)	$93{,}84$ $[93{,}25 - 94{,}43]$
Platt modificado+R-V ($A = -1{,}0$) (log)	$\mathbf{95{,}04}$ $[94{,}51 - 95{,}57]$
Platt modificado+R-V ($A = -1{,}5$) (log)	$94{,}88$ $[94{,}35 - 95{,}41]$
Platt modificado+R-V ($A = -2{,}0$) (log)	$94{,}74$ $[94{,}20 - 95{,}28]$

Tabla 1.
Tasas de reconocimiento de palabras (%) para los diversos métodos estudiados.

Las tasas de reconocimiento de palabras se presentan junto con sus correspondientes intervalos de confianza al $95\,\%$. Para calcular dichos intervalos se ha usado la siguiente expresión ([18], páginas 407-408):

$$\frac{\Delta}{2} = 1{,}96\sqrt{\frac{p\,(100-p)}{n}}. \qquad (21)$$

El parámetro p denota el porcentaje de palabras reconocidas correctamente en el experimento y n es el número de palabras presentes en el conjunto de test (6546). De esta forma, las tasas de reconocimiento que se presentan en la tabla se encuentran en el intervalo $\left[p - \frac{\Delta}{2}, p + \frac{\Delta}{2}\right]$ con una confianza del $95\,\%$.

Como se puede ver, todos los métodos que se basan en el cálculo del margen *acumulado* a partir de las salidas de las SVMs binarias obtienen peores resultados que aquellos que emplean la probabilidad de Platt. La razón más probable de este comportamiento se señaló en la sección 3: el cambio en la escala de los valores que aporta la etapa de modelado acústico en el Viterbi puede estar produciendo un desajuste en la ponderación del modelado del lenguaje que influya en los resultados finales.

Por otra parte, se observa que la modificación del método de Platt que se plantea en este artículo, consistente en usar una única sigmoide para transformar la salida de todas las SVMs binarias en las correspondientes probabilidades *binarias*, no produce diferencias significativas en la tasa de reconocimiento de palabras frente al método de Platt presentado en [5], para un rango de

$A \in [-2, ..., -1]$. Esto significa que no se obtiene ninguna ventaja real al estimar los parámetros A y B de forma individual en cada una de las 1431 SVMs binarias y, por tanto, este proceso puede eliminarse de la etapa de entrenamiento de la SVM multiclase, con lo que se consigue una importante reducción del tiempo de entrenamiento.

Finalmente, a tenor de los resultados presentados en la tabla 1, el método de Refregier-Vallet modificado [3] se presenta como una forma sensata de transformar las probabilidades *binarias* $r_{ij}(\mathbf{x})$ en probabilidades *multiclase* $P(y = i|\mathbf{x})$.

5. CONCLUSIONES

Ciertas aplicaciones requieren conocer el nivel de incertidumbre asociado a la decisión de un clasificador o la probabilidad a posteriori de cada una de las clases posibles. En el caso de las Máquinas de Vectores Soporte, la obtención de dichas probabilidades a partir de sus salidas no es obvia. La forma más habitual consiste en calcular la probabilidad de Platt y, en el caso multiclase, utilizar una modificación del método de Refregier-Vallet para obtener dichas probabilidades a posteriori. No obstante, ambos métodos presentan una serie de limitaciones tanto en su base teórica como en su aplicación práctica.

En este artículo se presentan diversas alternativas a estos dos métodos, en el marco del reconocimiento de habla continua mediante SVMs. Se ha comprobado experimentalmente que el método de Platt se puede simplificar, utilizando una única función sigmoide para obtener las probabilidades *binarias* a partir de las salidas de las SVMs binarias. Así mismo, se ha mostrado que los resultados obtenidos son poco sensibles a la pendiente de la sigmoide en la zona lineal, en un intervalo $[-2, -1]$. La consecuencia práctica es que el tiempo empleado en el entrenamiento de la máquina se reduce de una forma muy importante (entre 4 y 5 veces), factor de gran importancia dado el tamaño de las máquinas y de las bases de datos de voz consideradas.

6. BIBLIOGRAFÍA

[1] B. Schölkopf y A. Smola, *Learning with kernels*, MIT Press, Cambridge, MA, 2002.

[2] J. Padrell-Sendra, D. Martín-Iglesias, y F. Díaz de María, "Support Vector Machines for Continuous Speech Recognition," *Proc. EUSIPCO, Florence, Italy*, 2006.

[3] T. F. Wu, C. J. Lin, y R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Proc. NIPS, Vancouver, Canada*, 2003.

[4] J. Milgram, M. Cheriet, y R. Sabourin, "Estimating accurate multi-class probabilities with support vector machines," *IEEE International Joint Conference on Neural Networks*, vol. 3, pp. 1906–1911, 2005.

[5] J. C. Platt, *Probabilities for SV Machines, in Advances in Large Margin Classifiers*, MIT Press, Cambridge, MA, 1999.

[6] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.

[7] F. Pérez-Cruz y O. Bousquet, "Kernel Methods and Their Potential Use in Signal Processing," *Signal Processing Magazine*, vol. 21, no. 3, pp. 57–65, 2004.

[8] J. Nocedal y S. J. Wright, *Numerical Optimization*, Springer, New York, USA, 1999.

[9] K. Crammer y Y. Singer, "On the algorithmic implementation of multiclass kernelbased vector machines," *Journal of Machine Learning Research*, vol. 2, no. 5, pp. 265–292, 2001.

[10] J. Weston y C. Watkins, "Multi-Class Support Vector Machines," Tech. Rep., Department of Computer Science, Royal Holloway, University of London, Egham, UK, 1998.

[11] C. W. Hsu y C. J. Lin, "A Comparison of Methods for Multi-class Support Vector Machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[12] T. Hastie y T. Tibshirani, "Classification by pairwise coupling," *Proc. NIPS, Denver, USA*, 1997.

[13] P. Refregier y F. Vallet, "Probabilistic approach for multiclass classification with neural netwoks," *Proc. International Conference on Artificial Netwoks*, pp. 1003–1007, 1991.

[14] T. Hamamura, H. Mizutani, y B. Irie, "A multiclass classification method based on multiple pairwise classifiers," *Proc. 7th International Conference on Document Analysis and Recognition, Edinburgh, UK*, vol. 2, pp. 809–813, 2003.

[15] C. C. Chang y C. J. Lin, "LIBSVM: A Library for Support Vector Machines," *Software disponible en http://www.csie.ntu.edu.tw/cjlin/∼libsvm*, 2001.

[16] S. Young et al., "HTK-Hidden Markov Model Toolkit (versión 3.2)," *Software disponible en http://htk.eng.cam.ac.uk/*, 2002.

[17] A. Moreno, "SpeechDat Spanish Database for Fixed Telephone Network," Tech. Rep., Technical University of Catalonia, 1997.

[18] N. A. Weiss y M. J. Hasset, *Introductory Statistics*, Addison-Wesley, Reading, MA, 3rd edition, 1991.

ESTUDIO DE HEURÍSTICAS PARA LA IMPLEMENTACIÓN DE A* EN CTH BASADOS EN SELECCIÓN DE UNIDADES

Lluís Formiga y Francesc Alías

Departamento de Comunicaciones y Teoría de la Señal
Enginyeria i Arquitectura La Salle. Universitat Ramon Llull
Pg. Bonanova 8. 08022 Barcelona
$\{llformiga, falias\}$ @$salle.URL.edu$

RESUMEN

The Unit Selection based Text to Speech Systems (US-TTS) need to perform an optimal search of units in a speech-corpus, hence in order to obtain a high-quality synthesis. This search, until nowadays, has been carried out by a Viterbi algorithm. Our work replaces the formerly used algorithm for the A* algorithm to enhance its computational efficiency. With that goal, a review of previous work that intend this substitution is detailed. Afterwards, a benchmark is defined to score its efficiency and results are analyzed to validate, in the last step, its theorical argumentation.

1. INTRODUCCIÓN

Este trabajo se emmarca dentro del ámbito de los conversores de texto en habla basados en selección de unidades (CTH-SU) a partir de un corpus de voz con más de una realización por unidad fonética. La selección de unidades es un proceso que ejerce de cuello de botella en el contexto de la CTH-SU [1, 2, 3]. El proceso de la selección de unidades parte de una especificación fonética y prosódica de la unidad deseada indicada por el módulo de procesamiento del lenguaje natural del CTH-SU (PLN). Típicamente, a cada fonema se le asocia una prosodia, caracterizada por su frecuencia fundamental y duración entre otros. La selección se efectúa en una gran base de datos de unidades candidatas mediante unas funciones de coste [1] ponderadas adequadamente [4, 5, 6]. El proceso más utilizado para encontrar la mejor secuencia de unidades del corpus se realiza mediante una búsqueda *naive* de Viterbi que evalua los costes de unidad. La señal de voz que conforma dichas unidades es concatenada y ajustada para satisfacer las especificaciones originales.

En [3] y [7] se menciona y argumenta el incanzable aumento complejidad computacional para la minimización de la función de coste. Se detallan heurísticas (conocimiento experto) en varios puntos para mejorar la eficacia: proporcionar un tratamiento especial para las pausas, o predisponer la concatenación de semifonemas específicos son algunos ejemplos.

Durante los últimos años se han dedicado esfuerzos importantes a investigar cómo reducir la complejidad computacional y la obtención de una aproximación de síntesis en tiempo real [3, 8]. Parte de la investigación [9] se ha concentrado en disminuir el espacio de búsqueda, clusterizandolo mediante árboles de decisión. Otra línea de investigación [10] se basa en optimizar el método de búsqueda mediante técnicas basadas en poda o con *caché*, que se basan en evitar el cálculo de caminos que producen un coste no aceptable (vía estimación o comparativa de cálculo hecho).

En este trabajo se pretende dar una retrospectiva del estado de la cuestión de los resultados obtenidos hasta el momento con el objetivo de minimizar la complejidad de los costes de computación. Se propone reemplazar el algoritmo de Viterbi por el A* justificando teóricamente su aplicación con trabajo previo; se acompaña de un estudio detallado de las heurísticas con las que debe trabajar éste para mejorar la eficiencia de la búsqueda. En la sección 2 se detalla el proceso de la selección de unidades. En la sección 3 se introduce el algoritmo A*, se compara con el algoritmo de Viterbi clásico y se justifica su utilización. En la sección 4.1 se detalla la implementación y se exponen las distintas heurísticas. La sección 5 detalla el conjunto de pruebas realizadas y los resultados obtenidos; éstos son discutidos en la sección 6. Finalmente se detallan las conclusiones y las líneas de futuro de este trabajo.

2. PROCESO DE SELECCIÓN DE UNIDADES

A continuación se detallan las características fundamentales del módulo de selección de unidades del sistema de CTH utilizado en este trabajo de investigación.

2.1. Ajuste de los datos por fonema a nivel de unidad

La unidad de trabajo de predicción a partir del texto es la prosodia por fonema, pero en nuestro caso se establece por difonemas puesto que trabajamos con selección de

Partially supported by the European Community under the Information Society Technologies (IST) priority of the 6th framework programme for R&D. This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of its content.

unidades. Es necesario adaptar la prosodia del fonema a nivel de difonema. Considerando que las unidades (tanto difonemas como trifonemas) están formadas por semifonemas (ver figura 1), el primer paso consiste en dividir los fonemas en semifonemas y en establecer los semifonemas como unidad mínima de representación del modelado prosódico. El parámetro prosódico más sencillo de adaptar es la duración: se distribuye equitativamente la duración del fonema entre los dos semifonemas.

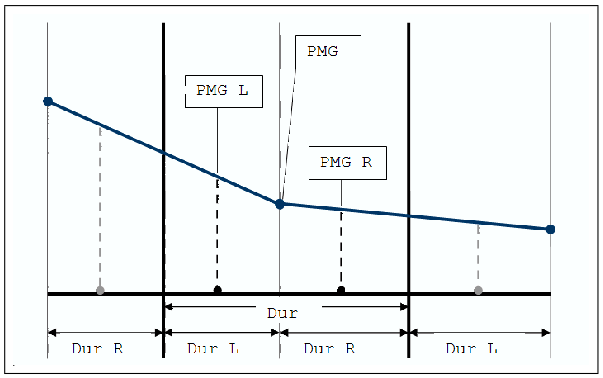


Figura 1. Recta de interpolación de la prosodia para pasar de fonema a semifonema

Siguiendo con los requisitos, transformar el *pitch* y la energía a semifonema es un proceso más complejo. Se consideran los requisitos presentados en el fonema como valores en el punto central (estable) del difonema. Cuando se divide el fonema en semifonemas los requisitos se conservan pero en el extremo de cada uno de los semifonemas: el extremo derecho en el semifonema izquierdo y extremo izquierdo en el semifonema derecho. Pero las características deseadas son las del punto central del semifonema, no las de los extremos. Para obtener estos datos se observará la tendencia contextual de la unidad teniendo en cuenta las prosodias del fonema anterior y del posterior. De esta forma se limitan los extremos de la variación prosódica más allá de la unidad. La recta de tendencia determinará el valor en los puntos requeridos. En el algoritmo 1 se dan más detalles del proceso.

Si se dispone de la prosodia para semifonemas trabajar por selección de unidades es immediato: todas las unidades están formadas por semifonemas y se pueden aplicar las mismas funciones de interpolación para determinar la prosodia de cada unidad.

2.2. Selección de Unidades

Para determinar qué unidades del corpus se seleccionan, se usa un algoritmo de programación dinámica que escoge la mejor combinación de unidades basándose en minimizar una función de coste de selección. Esta función de coste está desglosada en un coste de unidad (*target*) C^t (ver ecuación 2), que evalúa la diferencia entre la unidad candidata y la unidad objetivo, y en un coste de concatenación C^c equación 3, que evalúa la bondad de la unión entre las unidades [1].

Algorithm 1 Algoritmo de conversión de fonemas a semifonemas

procedure $splitFon(i)$

1: $Dur_L^i \Leftarrow Dur^i/2$
2: $Pitch_L^i \Leftarrow Pitch^{i-1} + (\frac{Dur_L^i}{2} * \frac{Pitch^i - Pitch^{i-1}}{Dur_R^{i-1} + Dur_L^i})$
3: $Energy_L^i \Leftarrow$
$Energy^{i-1} + (\frac{Dur_L^i}{2} * \frac{Energy^i - Energy^{i-1}}{Dur_R^{i-1} + Dur_L^i})$
4:
5: $Dur_R^i \Leftarrow Dur^i/2$
6: $Pitch_R^i \Leftarrow Pitch^i + (\frac{Dur_R^i}{2} * \frac{Pitch^{i+1} - Pitch^i}{Dur_R^i + Dur_L^{i+1}})$
7: $Energy_R^i \Leftarrow$
$Energy^i + (\frac{Dur_R^i}{2} * \frac{Energy^{i+1} - Energy^i}{Dur_R^i + Dur_L^{i+1}})$

$$C(t_1^n, u_1^n) = \sum_{i=1}^{n} C^t(t_i, u_i) + \sum_{i=2}^{n} C^c(u_{i-1}, u_i) \quad (1)$$

$$C^t(t_i, u_i) = \sum_{j=1}^{p} w_j^t C_j^u(t_i, u_i) \quad (2)$$

$$C^c(u_{i-1}, u_i) = \sum_{j=1}^{q} w_j^c C_j^c(u_{i-1}, u_i) \quad (3)$$

Donde t_1^n representa la secuencia de unidades objetivo $\{t_1, t_2, \ldots, t_n\}$ y u_1^n representa la secuencia de unidades candidatas $\{u_1, u_2, \ldots, u_n\}$.

$$C_j^t(t_i, u_i) = \frac{\left|P_j(t_i) - \overline{P_j}(u_i)\right|}{\sigma_{P_j}} \quad (4)$$

$$C_j^c(u_{i-1}, u_i) = \frac{\left|P_j^R(u_{i-1}) - P_j^L(u_i)\right|}{\sigma_{P_j}} \quad (5)$$

Donde $\overline{P_j}(\cdot)$ indica el valor medio del subcoste para la unidad analizada.

La selección aplicando Viterbi al proceso se realiza en dos fases. En la primera fase se crea una matriz de distancias entre todas las unidades candidatas y se asigna el coste asociado a cada una de las realizaciones. Una vez se ha obtenido la matriz de distancias se acumulan los costes de cada camino posible y se escoge el camino con el menor coste acumulado, obteniendo ya las unidades seleccionadas. La ponderación de los pesos de la selección (w_j^i) de las unidades de síntesis se puede realizar con procesos automáticos como MLR[1] o GA[4] o mediante procesos interactivos como en [11] y en [12].

La implementación del algoritmo de Viterbi aplicada a grandes bases de datos dificulta la tarea de realizar síntesis de calidad en tiempo real. En [10] se detallan los factores clave para deteminar la velocidad de síntesis para

sistemas basados en selección de unidades: la búsqueda que debe ser preprocesada y la eficiencia del coste computacional de la red de búsqueda.

En este artículo se propone reemplazar el algoritmo de programación dinámica de Viterbi, que devuelve los mejores caminos que presentan una distancia mínima, por un algoritmo que sólo encuentra la mejor distancia (A*).

3. ALGORITMO A* VS. VITERBI

3.1. Trabajo relacionado

Un primer enfoque de trabajo con los dos algoritmos para selección de unidades es empleado por [8] al usar transductores de estados finitos para hacer una búsqueda probabilística por Viterbi primero, y seleccionar la mejor sequencia de unidades con un A*.

Principalmente, el marco de investigación donde se han comparado estos algoritmos es en procesado del lenguaje natural (PLN) para el estudio de gramáticas (gramáticas probabilisticas incontextuales - *PCFG*). En [13] se detalla la comparativa entre A* y Viterbi concluyéndose que se obtiene la mejor solución en ambos casos y que en el peor caso de ordenamiento de heurísticas tienen el mismo coste computacional. Se destaca como ventaja la sencillez de construir el modelo A* reduciendo la complejidad computacional por transición.

En [7] se expone una comparativa entre los dos algoritmos trabajando en la decodificación dinámica posterior a la clasificación acústica para el reconocimiento del habla (ASR).

Siguiendo el razonamiento de [14] se justifica la utilización de Viterbi en selección de unidades por la tendencia y buenos resultados en ASR.

En base a estos trabajos consideramos utilizar A* en lugar de Viterbi por un cambio de paradigma de búsqueda en el problema. En la selección de unidades el hecho de obtener la función de coste es computacionalmente compleja y, a la vez, resulta factible estimar heurísticas por la similitud de los costes y el conocimiento previo de sus estadísticas.

3.2. El algoritmo de Viterbi

El algoritmo de Viterbi se divide en la parte de *forward* y una parte *backward*. Las principales aplicaciones del algoritmo de Viterbi son en problemas de estimación y detección en comunicaciones digitales y procesado de la señal.

En procesado de la señal en reconocimiento del habla se modela la señal de voz con modelos ocultos de Markov (HMM). Dado un estado inicial y un estado final definidos, devuelve el orden de los caminos que tienen menor probabilidad de error en los estados probabilísticos intermedios que los conforman[15]. En la fase de *forward* se acumula el error probabilístico para cada camino posible (sucesión de posibles estados con sus transiciones), sobreviviendo solo los caminos con menor error para cada estado. Una vez llegado al estado final, se procede en un proceso *backward* para reconstruir el camino de menor error hasta el estado inicial.

3.3. El algoritmo A*

A* es un algoritmo clásico de búsqueda *best-first search* estructurado en árbol, para la resolución de problemas en grafos. Fue formulado por Hart, Nilsson y Raphael (1968), revisado en 1969 y resumido por Nilsson en 1971 [16]. A diferencia del paradigma de la programación dinámica, el A* se basa en la decodificación de una pila de estados de búsqueda, donde el modelo de búsqueda es un árbol con las hipótesis parciales representadas por las distintas ramas hasta las hojas. Un concepto nuevo en el paradigma A* es la búsqueda heurística, ya que las heurísticas son un factor clave del éxito del algoritmo A*. Unas heurísticas pobres determinan un comportamiento similar a Viterbi o *backtracking* en terminos de coste computacional (y con más consumo de memoria).

A diferencia del algoritmo de Greedy, considerado como algoritmo heurístico puro, A* mantiene las propiedades de realizar una búsqueda óptima y completa [16], pero en cambio, prioriza el orden de búsqueda en función de heurísticas inherentes del problema. De este modo, se reduce el coste computacional de un algoritmo exaustivo de búsqueda como el *backtracking* aunque la reducción tiene relación con la calidad de las heurísticas. La principal ventaja del algoritmo A* es la computación del coste de un vértice y viene determinado por dos factores: sólo se computan los costes según la recomendación de la función heurística, evitando el cálculo de la totalidad de ellos, y no es necesario computarlos más de una vez. Por consiguiente es recomendable para problemas donde la evaluación de un estado es computacionalmente complejo, por ejemplo la función de coste en selección de unidades. En cambio, el principal inconveniente del A* es la memoria de la pila de búsqueda: para limitar el tamaño de la pila de memoria con problemas complejos existen las variantes IDA* (Iterative Deeping A*) y SMA* (Simplified Memory-bounded A*) [16] aunque no garantizan la mejor solución al limitar la búsqueda.

4. IMPLEMENTACIÓN

4.1. Consideraciones Iniciales

El algoritmo principal que rige el A* se detalla en el algoritmo 2. La filosofía del algoritmo es memorizar estados (nodos) de la búsqueda en cada iteración, asociándoles un coste a cada uno. Para proceder así almacena una lista de nodos para visitar y una lista de nodos visitados. A medida que va progresando la búsqueda actualiza los costes de los nodos ya visitados anteriormente. Dicho coste se calcula mediante la ecuación 6 donde el coste $f(n)$ del nodo n es el coste $g(n)$ acumulado hasta el nodo por el camino que se está iterando, más la estimación del

Algorithm 2 Algoritmo de búsqueda A*

procedure doAStarSearch($start$,$goal$)

```
openList ⇐ {start}
closedList ⇐ ∅
while openList ≠ ∅ do
  cur ⇐ first(openList)
  if cur = goal then
    return cur
  else
    for all succ_i ∈ cur ⇒ {succ_1, ..., succ_n} do
      cost = computeCost(cur, succ_i)
      cost = cost + goalHeuristic(succ_i)
      if succ_i ∈ openList then
        if cost < cost(openList[succ_i]) then
          openList[succ_i] = {cur, cost}
        end if
      else
        if succ_i ∈ closedList then
          if cost < cost(closedList[succ_i]) then
            closedList[succ_i] = ∅
          end if
        end if
        openList[succ_i] = {cur, cost}
      end if
    end for
    add(closedList, cur)
  end if
  sort(openList)
end while
```

coste $h'(n)$ de dicho nodo hasta el nodo objetivo.

$$f(n) = g(n) + h'(n) \tag{6}$$

En el caso de la selección de unidades el nodo inicial es una sequencia vacía de unidades seleccionadas y el estado objetivo es la secuencia con todas las unidades seleccionadas. El coste $g(n) = C(t_1^n, u_1^n)$ se calcula como se define en la ecuación 1. Para mejorar la eficiencia del algoritmo se implementan optimizaciones basadas en memoria caché para los costes de *target* y concatenación tal como se propone en [3].

4.2. Estudio de las Heurísticas

Las heurísticas son un factor clave para la eficacia del algoritmo. Para encontrar una solución rápidamente se necesita una heurística que no sobreestime el coste de pasos hacia la meta.

El factor efectivo de branqueo (b*) es un método propuesto [16] para evaluar la calidad de una heurística. Su fórmula se define en la ecuación 7, donde N es el número de nodos expandidos para la búsqueda y d es la profundidad de la solución en el árbol.

$$N = 1 + b^* + (b^*)^2 + \cdots + (b^*)^d \tag{7}$$

Por ejemplo, si para seleccionar 12 unidades (la d de la ecuación - pues la profundidad de la solución en el arbol es el número de unidades) se exploran 320 nodos (N), entonces $b^* = 1{,}47$. Este factor se suele mantener constante aunque aumenten las unidades a seleccionar. Una buena heurística debe tener un factor cercano a 1 [16], permitiendo así soluciones eficaces en problemas grandes. Al probar varias heurísticas para el mismo problema pero con distintas d, si una heurística h_1 es en todos los casos mejor que una heurística h_2, entonces se puede aplicar el concepto de dominancia pues h_1 domina a h_2 ($h_1 \geq h_2$): eso conlleva que la heurística h_1, en media, produzca menos nodos que h_2. En el mismo trabajo se concluye que siempre es "mejor utilizar una función heurística con valores altos mientras no sobreestime la d". Otra conclusion del trabajo es la definición de múltiples heurísticas para finalmente seleccionar el valor máximo (*worst-case*) proporcionado en cada una de ellas. Finalmente, se recomienda trabajar con datos estadísticos si hay experiencia previa en la exploración de nodos por dicho problema.

5. PRUEBAS Y RESULTADOS

Para implementar las heurísticas se define un banco de pruebas compuesto de 3 frases, 48 configuraciones de pesos diferentes y un barrido de 4 heurísticas globales para todo el corpus. Seguidamente se procede a explicarlo en detalle:

1. *Frases*: Se han seleccionado para sintetizar 3 frases distintas ("mantenen discretes","reunit amb els consellers","tenen amb els"). Dichas frase son fruto de un proceso de selección previo para determinar aquellas frases que presentan mayor variabilidad.

2. *Pesos*: Se han iniciado unos 144 conjuntos de pesos (48 para cada frase) que aportan un buen barrido de combinaciones para así proporcionar distintos costes en cada estado.

3. *Heurísticas* Las heurísticas se han implementado a nivel global por lo tanto aplica la misma en cada estado de la búsqueda. El resumen de las heurísticas efectuadas se presenta en las ecuaciones 8910111213 donde N_u es el número de unidades totales por seleccionar y N_{us} el numero de unidades ya seleccionados en ese estado.. La estimación de profundidad se realiza en función de las unidades que quedan por seleccionar. Como los costes normalizados se mueven en el intervalo [0,1] se prueban distintus valores en su intervalo de más frecuencia.

Para analizar los resultados se hace en función de tres parametros: se analiza el factor de branqueo eficaz (b*) ya que según [17] es el método ideal para evaluar una heurística independientemente del tamaño del problema. Se analizan, a la vez, los parámetros clásicos de tiempo de ejecución y número de nodos explorados. Aun así, el

Heurística	Frase	Pasos de búsqueda	b*	Tiempo Ejecución
VITERBI	Tenen amb els	$59774 \pm 0,00$	$2,39 \pm 0,00$	$29,65 \pm 0,56$
VITERBI	Mantenen discretes	$45843 \pm 0,00$	$1,79 \pm 0,00$	$23,06 \pm 0,56$
VITERBI	Reunit amb els consellers	$45508 \pm 0,00$	$1,59 \pm 0,00$	$21,98 \pm 0,67$
0,666	Tenen amb els	$640,12 \pm 262,85$	$1,55 \pm 0,11$	$26,60 \pm 11,71$
0,666	Mantenen discretes	$787,10 \pm 588,69$	$1,31 \pm 0,12$	$27,50 \pm 20,38$
0,666	Reunit amb els consellers	$890,68 \pm 519,99$	$1,25 \pm 0,10$	$20,14 \pm 10,69$
0,75	Tenen amb els	$640,12 \pm 262,85$	$1,55 \pm 0,11$	$26,95 \pm 11,69$
0,75	Mantenen discretes	$787,10 \pm 588,69$	$1,32 \pm 0,12$	$28,02 \pm 20,71$
0,75	Reunit amb els consellers	$890,68 \pm 519,99$	$1,25 \pm 0,10$	$20,52 \pm 10,83$
0,85	Tenen amb els	$427,66 \pm 481,60$	$1,39 \pm 0,22$	$16,31 \pm 19,17$
0,85	Mantenen discretes	$691,12 \pm 823,86$	$1,24 \pm 0,17$	$24,14 \pm 28,62$
0,85	Reunit amb els consellers	$473,54 \pm 772,12$	$1,14 \pm 0,12$	$10,79 \pm 16,56$
0,92	Tenen amb els	$427,66 \pm 481,60$	$1,39 \pm 0,22$	$16,25 \pm 19,03$
0,92	Mantenen discretes	$691,12 \pm 823,86$	$1,24 \pm 0,17$	$23,97 \pm 28,21$
0,92	Reunit amb els consellers	$473,54 \pm 772,12$	$1,14 \pm 0,12$	$10,75 \pm 16,32$
1	Tenen amb els	12 ± 0	$0,98 \pm 0$	$1,39 \pm 0,49$
1	Mantenen discretes	17 ± 0	$0,99 \pm 0$	$1,54 \pm 0,50$
1	Reunit amb els consellers	21 ± 0	$0,99 \pm 0$	$1,75 \pm 0,43$

Tabla 1. Resultados obtenidos según los parámetros de evaluación (pasos de búsqueda, b*, tiempo ejecución) para todo el *benchmark* de pruebas

tiempo de ejecución no es una medida determinante debido a que puede variar dependiendo de la gestión de procesos del sistema operativo. Los resultados obtenidos se muestran en la tabla 1 y en las gráficas comparativas de tiempo 2, de factor de branqueo 3 y de nodos visitados 4 (observar que se ha trazado con escala logarítmica).

$$d_{left} = N_u - N_{us} \quad (8)$$

$$h'(n) = d_{left} * 0{,}666 \quad (9)$$

$$h'(n) = d_{left} * 0{,}750 \quad (10)$$

$$h'(n) = d_{left} * 0{,}850 \quad (11)$$

$$h'(n) = d_{left} * 0{,}920 \quad (12)$$

$$h'(n) = d_{left} * 1{,}000 \quad (13)$$

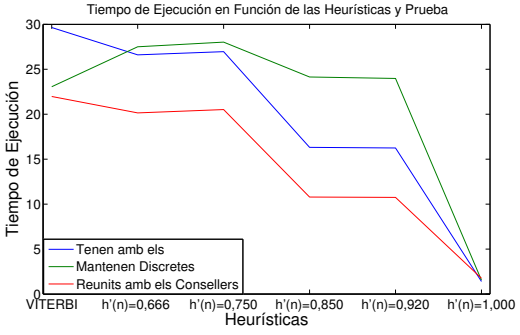


Figura 2. Tiempo de ejecución detallado para cada heurística y prueba

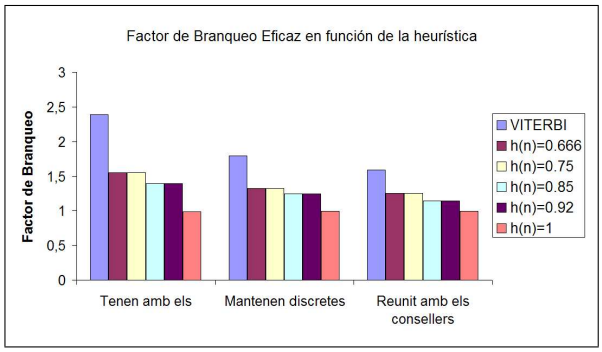


Figura 3. Factor de branqueo eficaz en función de las pruebas y heurísticas

6. DISCUSIÓN

Una vez planteados los resultados se puede observar que el A*, si se tiene en cuenta la exploración de nodos y el factor de branqueo eficaz, es drásticamente mejor que el algoritmo de Viterbi. También se observa, como ya se ha mencionado en trabajos anteriores, que el tiempo de ejecución en sus peores heurísticas (ver ecuaciones 9 y 10), se comporta de forma equivalente al algoritmo de Viterbi; en cambio, una buena heurística (ver ecuacion 13) puede acelerar el proceso. A la vez, los resultados validan la teoría que la heurística más pesimista (ver ecuacion 13) es la que se comporta mejor siempre y cuando no sobreestime la profundidad del árbol de búsqueda.

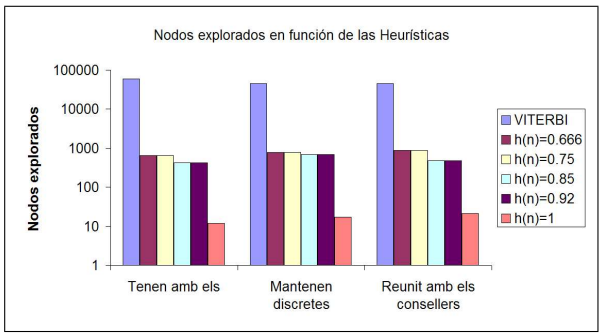


Figura 4. Nodos explorados en función de las pruebas y heurísticas

7. CONCLUSIONES Y LÍNEAS DE FUTURO

En este trabajo se aplica un nuevo algoritmo para la selección de unidades, justificando su elección en los trabajos anteriores y en los resultados obtenidos en nuestro caso. Se abre un nuevo frente de investigación sobre heurísticas de este algoritmo, puesto que este trabajo no deja de ser un primer enfoque.

En un trabajo futuro se plantea efectuar la selección A* con clusterización del espacio de búsqueda con árboles de decisión, y aplicar el trabajo hecho con estados finitos de transducción. Aún así, los resultados obtenidos son prometedores y representan un avance cualitativo y cuantitativo en el estudio de la eficacia de la selección de unidades en CTH-SU en tiempo real.

8. BIBLIOGRAFÍA

[1] A. Hunt y A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proceedings of ICASSP*, Atlanta, USA, 1996, vol. 1, pp. 373–376.

[2] A. W. Black y P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," in *Proceedings of EuroSpeech*, Rodas, Grecia, 1997, pp. 601–604.

[3] M. Beutnagel, M. Mohri, y M. Riley, "Rapid unit selection from a large speech corpus for concatenative speech synthesis," in *Proceedings of EuroSpeech*, Budapest, Hungría, 1999, vol. 2, pp. 607–610.

[4] F. Alías y X. Llorà, "Evolutionary weight tuning based on diphone pairs for unit selection speech synthesis," in *Proceedings of the 8th European Conference on Speech Communication and Technology (EuroSpeech)*, 2003.

[5] F. Alías, X. Llorà, I. Iriondo, y L. Formiga, "Ajuste subjetivo de pesos para selección de unidades a través de algoritmos genéticos interactivos," *Procesamiento del Lenguaje Natural*, vol. 31, pp. 75–82, Septiembre 2003.

[6] Y. Meron y K. Hirose, "Efficient weight training for selection based synthesis," in *Proceedings of EuroSpeech*, Budapest, Hungría, 1999, vol. 5, pp. 2319–2322.

[7] Nikko Ström, "Automatic continuous speech recognition with rapid speaker adaptation for human/machine interaction - PhD Thesis," 1997.

[8] J Yi y J. Glass, "Natural-sounding speech synthesis using variable-length units," 1998.

[9] M Macon, A. Cronk, y J. Wouters", "Generalization and discrimination in tree-structured unit selection," 1998, note: "Proc. 3 rd Synthesis Workshop, Nov. 1998.

[10] A. Conkie, M. C. Beutnagel, A. K. Sydal, y P. E. Brown, "Preselection of candidate units in a unit selection-based text-to-speech synthesis system," in *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP)*, Beijing, China, 2000, vol. 3, pp. 314–317.

[11] F. Alías, X. Llorà, L. Formiga, K. Sastry, y D. E. Goldberg, "Efficient interactive weight tuning for tts synthesis: reducing user fatigue by improving user consistency," in *Proceedings of ICASSP*, Toulouse, Francia, 2006, vol. I, pp. 865–868.

[12] A. K. Syrdal y A. D. Conkie, "Perceptually-based data-driven join costs: Comparing join types," in *Proceedings of the 9th International Conference on Speech Communication and Technology (InterSpeech)*, Lisboa, Portugal, 2005, pp. 2813–2816.

[13] Dan Klein y Christopher D. Manning, "A* parsing: Fast exact viterbi parse selection," in *Proceedings of HLT-NAACL 03*, 2003.

[14] M. Ostendorf y I. Bulyko, "The impact of speech recognition on speech synthesis," in *Proceedings of 2002 IEEE Workshop on Speech Synthesis*, Santa Monica, USA, 2002.

[15] H. L Lou, "Implementing the viterbi algorithm," *Signal Processing Magazine, IEEE*, vol. 12, no. 5, pp. 42–52, 1995.

[16] Stuart Russell y Peter Norvig, *Artificial Intelligence: A modern approach*, Prentice Hall, 1995.

[17] T Dutoit y H. Leich, "Mbr-psola: Text-to-speech synthesis based on an mbe re-synthesis of the segments database," *Speech Communication*, vol. 13, pp. 435–440, Junio 1996, note:.

FIRST EXPERIMENTS ON AN HMM BASED DOUBLE LAYER FRAMEWORK FOR AUTOMATIC CONTINUOUS SPEECH RECOGNITION

Albino Nogueiras, Marta Casar, José A. R. Fonollosa, Mónica Caballero

TALP Research Center
Universitat Politècnica de Catalunya (UPC). Barcelona, Spain.
{albino,mcasar,adrian,monica}@gps.tsc.upc.edu

RESUMEN

The usual approach to automatic continuous speech recognition is what can be called the acoustic-phonetic modelling approach. In this approach, voice is considered to hold two different kinds of information—acoustic and phonetic—. Acoustic information is represented by some kind of feature extraction out of the voice signal, and phonetic information is extracted from the vocabulary of the task by means of a lexicon or some other procedure. The main assumption in this approach is that models can be constructed that capture the correlation existing between both kinds of information.

The main limitation of acoustic-phonetic modelling in speech recognition is its poor treatment of the variability present both in the phonetic level and the acoustic one. In this paper, we propose the use of a slightly modified framework where the usual acoustic-phonetic modelling is divided into two different layers: one closer to the voice signal, and the other closer to the phonetics of the sentence. By doing so we expect an improvement of the modelling accuracy, as well as a better management of acoustic and phonetic variability.

Experiments carried out so far, using a very simplified version of the proposed framework, show a significant improvement in the recognition of a large vocabulary continuous speech task, and represent a promising start point for future research.

1. INTRODUCTION

The acoustic-phonetic approach, mainly using hidden Markov models (HMM), has shown to be a powerful tool for speech recognition [1]. In controlled conditions—i.e., one speaker, using the same recording framework, and in the same favourable environment—recognition accuracies are high, even in very large vocabulary recognition tasks, say automatic dictation. But when conditions degrade and/or change, performance goes down, even for simple tasks as digit strings recognition.

The standard way of dealing with more-than-one speaker, and/or multiple or changing conditions, is to embed all this variability in the model that links the acoustic and the phonetic informations. The framework does not change much whether we expect the speaker to be always the same, using always the same microphone in the same environment, or not. Just the training material changes, incorporating samples of whichever condition we wish our system to be robust.

In this paper we propose the use of a double layer approach to acoustic-phonetic modelling in order to cope—at least partially—with some of the factors that contribute to degrade performance on speaker and recording conditions independence. The main idea is to divide into two layers the standard acoustic modelling: an upper layer, closer to the lexicon; and a lower layer, closer to the acoustic features.

In Section 2, the double layer framework is presented. Section 3 presents the experimentation carried out and the results achieved. Finally, some conclusions and future work are discussed in Section 4.

2. HMM BASED DOUBLE LAYER FRAMEWORK FOR SPEECH RECOGNITION

2.1. Speech Recognition Using HMM's

A standard speech recognition system is based on a set of so called acoustic models that link the observed features of the voice signal with the expected phonetics of the sentence. The most usual implementation of this link is probabilistic, namely HMM's. In this kind of system we can recognise three levels: phonetics, acoustic model, and acoustic features.

The main assumption in standard acoustic phonetic modelling is that phonetic units can be selected such that each word of any vocabulary can be completely expressed by means of them, and that we can estimate the probability density function of these units in the feature space. If this is the case, minimum risk Baye's rule can be effectively used to determine the most probable meaning for a given utterance.

2.1.1. The Phonetics

In the standard approach, it is necessary to establish the expected phonetic transcription of both the training material and the vocabulary to be recognised. In general it

can be accepted that languages are quite well represented by means of known phonetic rules. Thus, any task can be expressed in terms of phonetic units that are only dependent on its vocabulary. Nevertheless, it is also well known that there are many exceptions, variants and a wide range of specific problems. For instance, the phonetic transcription rules may change with the speaker's dialect or age group.

Variation at the phonetic level can be treated in several ways. If the speaker is always the same, or we can expect the users of the system to share the same dialectal influences, or we can determine these influences, then we can build a dialect dependent system, with just one transcription per word.

When dialectal dependence is not possible, a common approach is to add alternative pronunciations to the lexicon. But this approach has a big disadvantage: it increases notably the complexity of the grammar, leading to bigger perplexity and higher computational cost. The worst thing here is that, many times, the increased perplexity produces an increase in the error rate that the higher phonetic accuracy does not compensate.

Finally, the third and most usual way of dealing with phonetics variation is to ignore it. In spite of its simplicity, ignoring dialect variations may be enough in many languages and circumstances. The idea of the approach is training with many realisations of each of the possible dialects present in the scope of the task. In this way it is expected that the model embed the different variations providing a single distribution function for all of them. The drawback is straightforward: the acoustic model have to deal with samples of different behaviour, so it becomes more general, less precise. The main advantage is precisely its simplicity. It does not rely in knowing a priori the dialect, does not increase the perplexity of the task, and does not require higher CPU resources.

2.1.2. The Model

An HMM is a collection of *states*. Each frame of voice can be in one and just one of the states at any time. Each HMM is formed of two different parts: a *transition matrix*, and a set of *emission probability functions*. The transition matrix of an N-state HMM is an $N \times N$ matrix. Each element of the matrix represents the probability of moving from one of the states to another. This transition matrix is common to all kind of HMM's, and has shown little influence on the final results.

Each state in the model contains a set of emission probability functions that provide the probability with which this state generates any frame. There are several ways of defining emission probability functions, but the most usual way is by means of mixtures of Gaussian densities, trained with the expectation maximisation algorithm. Yet two alternatives are possible: continuous HMM's, and semi-continuous HMM's.

In continuous HMM's each state is modelled with a mixture of private Gaussians. Gaussians are constructed with diagonal covariance matrix, so independence between components is assumed for each of them. But not for the mixture itself, that will not usually have diagonal covariance. As the number of Gaussians rapidly grows when the number of units and/or states grows, it is usual to tie together groups of them. This means that several states of several units share some of the Gaussian distributions, but not the mixture weights, which are still private.

In semi-continuous HMM's all the Gaussians are shared, and form a vector quantifier. For each frame, the probability density of all the Gaussians is calculated, and a score vector is formed out of the highest of them. The probability of this frame being in a given state is equal to the sum-product of the score vector that represents the frame, and the mixture vector that represents the state. As in the case of continuous models, each Gaussian assumes component independence, but the mixture of them does not. Yet, vector quantisation is rather difficult when the dimension of the vectors grows, so it is usual to assume feature independence. This means, for instance, that the probability of spectrum and energy are calculated at each state separately, and the state probability is given by the product of both.

2.1.3. The Acoustic Features

No matter what structure is used, the probability density functions in the states of the HMM have to model features extracted from the voice signal. The goal of these features will be providing the maximum information about the phonetics of the utterance, while maximally neutralising the effects of the remaining information present in it. Thus, we expect features to be sex and age independent, to be robust in front of noisy or changing conditions, etc. Unfortunately, it is hard to remove all the spurious components, without damaging the needed information.

Features normally used can be grouped in two ways: spectral/energy features, and static/dynamic features. Spectral features are aimed at modelling the spectral envelope, where it is expected to be the maximum phonetic information. Energy is also useful in the characterisation of some sounds and silence. Both spectral and energy features can be static, which means that the feature reflects the behaviour in the current frame; or dynamic, which means that the feature reflects its temporal evolution.

2.2. The Double Layer Framework

In the standard speech recognition framework using HMM's seen above, all the variation in the phonetics and acoustics is put into the same place: the acoustic model. A same model is used for all kind of speakers in all kind of conditions. This produces an intra-unit variance that can be in the same order of magnitude than the variance between units.

The objective of the double layer framework is to reduce the variance of the modelling by separating the acoustic modelling and the phonetic modelling into two linked layers. The lower layer is the acoustic layer. In this layer we pretend to capture all the variability present in the acoustic signal, either it will be considered in the upper layer or not. It has the structure of an acoustic-phonetic classifier, whose results is a score vector formed with the probabilities of each frame being at each state of the classifier. The upper layer is a standard semi-continuous HMM based recogniser, where the feature vectors are no longer probabilities in the feature space, but probabilities in a fuzzy space where the different codewords are assigned to the states of the lower layer classifier.

2.2.1. The Acoustic Layer

In the lower layer of the framework a phonetic-acoustic classifier is required. We call this classifier $\Lambda = \{\lambda_i\}$, where λ_i is the model of the ith unit in the lower layer classifier. At time n, a signal frame $x(n)$ has a probability of being at each of the states of the classifier. Being $q(n)$ the index of the state visited at time n, and q_{ij} the index of the jth state of unit i, we can calculate a score for each frame and state as:

$$\begin{aligned} \mathcal{S}(n,i,j) &= P(x, q(n) = q_{ij}/\Lambda) && (1) \\ &= \alpha_n(x, q_{ij})\beta_n(x, q_{ij}) && (2) \end{aligned}$$

Where $\alpha_n(q_{ij})$ is the standard *forward* probability—the probability of being at state q_{ij} at time n, given $x(k)$ from $k = 1$ to n—, and $\beta_n(q_{ij})$ is the *backward* probability—the probability of arriving from state q_{ij} at time n to the end of the utterance—.

In expression 2 the knowledge of the whole utterance is needed to estimate the score, so it cannot be used in real time applications. One possible alternative is to reduce the scope in which the score is calculated, considering only a certain context of each frame. Defining $x_N(n) = \{x(n-N)\dots x(n)\dots x(n+N)\}$, i.e. the context of x centred at $x(n)$ and of $2N+1$ frames, the score calculation becomes:

$$\begin{aligned} \mathcal{S}(n,i,j) &= P(x_N, q(n) = q_{ij}/\Lambda) && (3) \\ &= \alpha_{N+1}(x_N(n), q_{ij})\beta_{N+1}(x_N(n), q_{ij}) && (4) \end{aligned}$$

The structure of the acoustic models λ_i is free, as long as a score similar to that in expression 4 can be calculated. It is valid for both continuous and semi-continuous hidden Markov models.

2.2.2. The Phonetic Layer

The phonetic layer is a standard semi-continuous HMM framework. An interesting interpretation of semi-continuous models is to consider the vector quantisation step as a part of the feature extraction phase. In this way, speech is no longer represented by its raw features, but by the scores for the Gaussians that form the codebook. Yet, the quantisation step can be also placed in the recognition phase, but separated from the acoustic models. This would be equivalent to consider a double layer framework, where the lower layer is a *blind* classifier: the vector quantisation.

The double layer framework proposed is equivalent to substituting the blind classifier provided by the vector quantisation, with the scores calculated using equation 4. The upper layer must be constructed with semi-continuous HMM's because the new space does not support a reasonable distance definition.

2.3. Expected Advantages of the Double Layer Framework

By itself, the double layer framework is not substantially different from a standard semi-continuous framework. A founded critic is that it does not increase the knowledge given to the upper phonetic layer. It is identical to the standard single layer framework, trained with the maximum likelihood criterion, but not fed with the observation vectors but with other models also trained with the same criterion.

In our opinion, it is true that the double layer framework does not increase the freedom of the system. But information is provided to the system in the form of restrictions. Instead of modelling distribution functions that must embed a lot of variability, making it difficult to estimate them, and making them become wider and less selective, very specialised units are used in the lower level. This specialised units act like a smoothing: once given the probability of being at each of the low level states, the actual position of the frame in the feature space becomes irrelevant.

The strategy is, thus, a may-loose-at-first in order to win-at-the-end one. At first, little gain is expected. From this starting point we plan to add information, otherwise unconsidered, to improve performance. The advantages we expect to get using the double layer framework are of two natures: an increase in robustness, and the capability of modelling information at present not used.

2.3.1. Increased Robustness

We expect to obtain a direct gain in robustness in front of the standard framework because of the above mentioned smoothing effect. Well behaved frames will have a high probability in those states with the highest probability in the upper state. On the contrary, bad behaved frames—due to noise, distortion or bad pronunciation—will have the highest score for the wrong states, forcing the upper level model to accept these confusions as probable during the training phase, but not forcing it to learn the exact position of the erroneous frame.

Besides that, in the lower layer, closest to voice signal, there is no need to use the same task or transcription scheme as in the upper layer. Actually, both the training

and the testing material will be re-coded with the results of a classification process, and many of the original phonetic information of the training material will be lost in this re-coding. Thus, in the lower layer it is desirable that the classifier or classifiers used are as accurate as possible, but no matter in which transcription scheme. This can be used in increasing the robustness of the double layer framework in four ways:

- ◇ Building independent classifiers for each type of feature.
- ◇ Training the lower layer with the result of the decoding.
- ◇ Profiting non transcribed material.
- ◇ Using cross-validation.

By means of using independent classifiers we can get the maximum information from each of the features, while discarding the information not really relevant for distinguishing one sound from another. We may achieve this by using different acoustic units sets adapted to the characteristics of each feature. For instance, it is expectable that different upper-level phonetic units share the same spectrum or energy. If this shared feature is used in building different acoustic models, there is a chance that the wrong decision is made because of marginal differences in the probability distribution functions. But if this sharing is identified in the unit selection process, it can be neutralised by using the same acoustic unit. This procedure may be carried only in the lower layer, not in the upper, where different acoustic units must have different models if any of the features is different. By the same reason, it could neither be used in single layer frameworks.

If we do not really matter what does each classifier decode and, in the end, the results of the decoding will be the only acoustic information driven to the upper layer recogniser, we can increase the accuracy of what the decoder is able to distinguish by training it not with the theoretical phonetic transcription, but with the decoding results of the classifier. Later on this paper it will be experimentally shown that this approach is possible, and it even improves the accuracy of the recognition.

Moreover, if the transcription used in the lower layer is the result of a phonetic decoding, and all the original phonetic information will be discarded, we do not need this information. Thus, non transcribed material may be used to train the lower layer models.

One characteristic of the double layer framework is that both layers may be trained using different training material. In the recognition step, speaker independent decoding will be used to perform speaker independent recognition. In the training step, if the training material is equal in both layers, speaker dependent decoding is used. One way to solve this, and increase the robustness of the framework, is training with different material the upper and the lower layers, performing a sort of cross-validation.

Combining the use of the decoding result as transcription, the use of non transcribed training material, and cross-validation, a not-so-complex training scheme may be proposed. First, initial low level classifiers are trained using the available transcribed data. This initial classifiers are used to provide a transcription to the non-transcribed data. The definitive classifiers are then trained using the non-transcribed data. Finally, the upper layer models are trained with the transcribed data, using the definitive classifiers as lower layer.

2.3.2. Variability Modelling

In the standard single layer framework all the variability modelling effort relays in the acoustic model. Yet, it is well known that there are several variables that, if considered, lead to significant improvement of recognition performance. The most notable of these variables are sex, age, dialect and recording conditions. Incorporating these variables in the single layer framework—or in the upper layer of the herein proposed double layer framework—is very difficult. Instead, including it in the lower layer of the double layer framework is straightforward. We can have, for instance, sex dependent models in the layer closest to the voice signal. In the upper layer, that closest to the phonetics, we may have sex dependent or independent models.

Another source of variability very difficult and expensive to model using single layer frameworks is interword contexts. Context is known to have a strong effect on acoustics. Modelling context dependent units leads to notorious improvements over context independent ones, but this modelling is only easy to implement inside the word boundaries. Yet, it is known that context effects are as critical between words as inside them. As in the case of the speaker or environment variables, introducing this information in the lower layer is straightforward, regardless of it being considered or not in the upper layer.

3. EXPERIMENTATION

3.1. The Recognition Task

The task used to test the capabilities of the double layer framework is the recognition of telephonic free speech dialogues in a tourism information retrieval semantic domain. Both the training and the test material used were recorded inside the scope of the European Commission funded project LC-STAR [2], which is devoted to collecting lexica and corpora for automatic speech-to-speech translation.

The corpus is composed of 211 dialogues (422 different speakers), of which 16 define the standard test material. The remaining 195 dialogues are used for training the acoustic models. In mean, each dialogue lasts 9 minutes and is composed of 45 utterances, each of about 30 words. In total, 8418 utterances (29h45m, 230,000 words) were used for training, and 1040 utterances (3h40m, 28,000 words) were used as test material.

The objective of the task is recognising the words pronounced during the tourism information retrieval dialogues. The vocabulary of the task is composed of 7466 words, and the grammar perplexity is around 70. Both vocabulary and grammar were obtained from the training material. The test material presents an out-of-vocabulary rate over 1 %.

3.2. The Baseline System

The system used as baseline is the semi-continuous HMM based system RAMSES [3], using demiphones as acoustic units [4]. The main features of this system are:

- ◇ Speech is windowed every 10ms with 30ms window length. Each frame is parameterised with the first 14 mel-frequency cepstral coefficients (MFCC) and its first and second derivatives, plus the first derivative of the energy.
- ◇ Spectral parameters are quantified to 512 centroids, energy is quantified to 128 centroids. Frames are quantified to 16 centroids for spectral parameters, and to 4 for energy.
- ◇ 1000 demiphones are selected using a minimum entropy decision tree. Left demiphones are modelled with 5 states, right demiphones and silence are modelled with 6 states.
- ◇ State tying is applied following a minimum entropy criterion to reduce the number of distributions from 22,000 (1,000 units $\times$ 5.5 states $\times$ 4 kinds of feature) to just 2,100.

3.3. The Double Layer Alternatives

Implementing the full double layer framework as described in this paper is a really hard task. At this moment we have just studied some very simplified versions of it in order to assess the utility of the approach.

The main simplification is to reduce the scope of the low level decoding to just on frame, i.e. fixing N in equation (4) to zero. This is a very heavy simplification that reduces greatly the complexity of the system, but will probably have its cost in terms of recognition accuracy.

One implication of reducing the scope to just one frame is that, if the lower level classifiers are feature independent semi-continuous HMM's, the double layer framework is equivalent to a single layer semi-continuous framework using the original features and quantifier. Information added by the double layer will be, if any, in the form of smoothing or restrictions.

The second relevant simplification is that the original transcription using 1,000 demiphones is used for all four kinds of feature, and no information about sex, age, environment or context across words is used.

The five following alternatives have been tested:

Baseline The baseline system with no distribution tying.

Base Tied Baseline with distributions tied to 2,100 vectors.

DL Joint The double layer framework using as classifier the same distributions as used in **Base Tied**.

DL Indep The same framework as **DL Joint**, but training the classifier of each kind of feature independently.

DL Trans The same framework as **DL Indep**, but using as training transcription the phone decoding result.

baseline experiments	Baseline	36.2 %
	Base Tied	38.6 %
double layer frameworks	DL Joint	38.0 %
	DL Indep	37.3 %
	DL Trans	35.9 %

Tabla 1. *Word error rate in the five systems.*

4. CONCLUSIONS AND FUTURE WORK

As it can be readily seen from Table 1, tying the probability density functions produces an increase of almost 2.5 % in the error rate of the baseline. If these same distributions are directly used as classifiers in the double layer framework, recognition improves more than half a point. Further improvement is achieved when the lower layer classifiers are re-trained independently. Finally, when the result from recognising the training material is provided to train the lower layer classifiers, the best result is obtained, even outperforming the untied baseline.

In our opinion, these results, achieved with a very simplified version of what we intend to do in the future, confirm the usefulness of the approach, and encourage us to undertake the whole system. Yet, most of the work has to be done still.

5. BIBLIOGRAFÍA

[1] L, Rabiner and B.H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[2] "LC-STAR: Lexicon and corpora for speech to speech translation," IST-2001-32216.

[3] A. Bonafonte et alter, "RAMSES: el sistema de reconocimiento del habla continua y gran vocabulario desarrollado por la UPC," in *VIII Jornadas de Telecom I+D*, 1998.

[4] J.B. Mariño et alter, "The demiphone: a new contextual subword unit for continuous speech recognition," *Speech Communication*, vol. 32, no. 3, pp. 187–197, October 2000.

IMPLEMENTACIÓN DE UN RECONOCEDOR DISTRIBUIDO DE VOZ EN TIEMPO REAL SOBRE IP

Juan A. Morales Cordovilla[1], *Timo Bauman*[2], *José L. Pérez Córdoba*[1], *Antonio M. Peinado Herreros*[1], *Ángel M. Gómez García*[1]

[1]Dpto. de Teoría de la Señal, Telemática y Comunicaciones. Universidad de Granada. España
[2] Hamburg University. Alemania

RESUMEN

Este trabajo presenta una implementación de un sistema de reconocimiento distribuido del habla en tiempo real para su aplicación en Internet. El diseño se ha realizado tratando de conseguir hacer una aplicación escalable, versátil, eficiente y que sea portable entre diferentes sistemas operativos. El resultado ha sido una aplicación tipo cliente-servidor para el reconocimiento distribuido, que permite su aplicación a diferentes tareas de reconocimiento y la utilización de diferentes tipos de reconocedores, ejecutándose en varios sistemas operativos. El cliente hace uso del *front-end* estándar definido por la ETSI (ES 201 108), incluye un detector de voz para enviar sólo información cuando el locutor habla. La transmisión de la información se hace de acuerdo al RFC 3557. En el servidor se realiza el reconocimiento que incluye a su vez un detector de voz y utiliza la técnica de mitigación de pérdida de paquetes propuesta en el estándar de la ETSI. Las pruebas de reconocimiento reales y de simulación han demostrado una alta fiabilidad en varias condiciones de transmisión.

1. INTRODUCCIÓN

El desarrollo de dispositivos portátiles con conexión a Internet abre un campo amplio a las aplicaciones del reconocimiento de voz. Estos dispositivos no suelen disponer de una gran potencia de cálculo y/o memoria por lo que es preferible implementar un sistema de reconocimiento de voz remoto. La Figura 1 muestra un diagrama de dicho sistema. Como se observa en la citada figura, se hace uso del paradigma cliente-servidor seguido en el desarrollo de aplicaciones en Internet. En este caso, el cliente se encarga de recoger la señal de voz y enviarla codificada al servidor (reconocimiento de voz basado en red, NSR, siglas inglesas para *Network based Speech Recognition*) o bien de obtener un conjunto de parámetros que representan a la voz (reconocimiento de voz distribuido, DSR, *Distributed Speech Recognition*). Esta filosofía tiene la ventaja adicional de que el usuario no debe preocuparse por el sistema de reconocimiento y permite diseñar de forma separada el cliente y el servidor así como realizar el mantenimiento y actualización de forma separada tanto del cliente como del servidor.

Este trabajo se ha realizado con la financiación del MEC/FEDER dentro del proyecto TEC2004-03829/TCM..

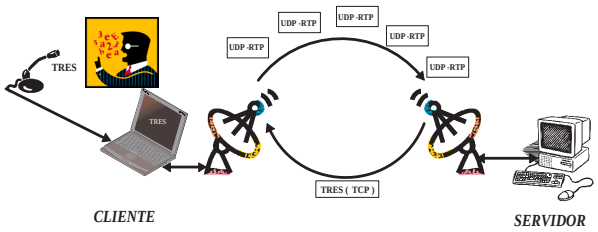


Figura 1. Esquema de un sistema de reconocomiento a través de IP.

Cada uno de los dos procedimientos de abordar el reconocimiento tiene sus ventajas e inconvenientes. Por ejemplo, DSR requiere la definición de un *front-end* estándar a seguir por todos los clientes. El Instituto Europeo de Estándares de Telecomunicación, ETSI (*European Telecommunications Standard Institute*) creó un grupo de trabajo, *Aurora DSR Working Group*, que ha publicado un conjunto de estándares sobre DSR. Sin embargo, es necesario que los fabricantes incluyan software y hardware que soporten dichos estándares.

Por otro lado, el uso de NSR no requiere la utilización de dichos estándares ya que, por ejemplo, en un teléfono móvil sólo hay que utilizar los codificadores de voz ya instalados. Sin embargo, estos codificadores de voz introducen una degradación en el reconocimiento además de requerir una tasa de bits superior.

Un ámbito de aplicación directa del DSR lo constituyen los ordenadores portátiles y PDA's. El abaratamiento de los precios, la extensión de su uso y la facilidad de conexión a Internet crean un entorno de desarrollo de aplicaciones de DSR. La implementación de los estándares ETSI para los *front-end* de reconocimiento en un portátil no ofrece ningún problema tanto del punto de vista de recursos hardware como software. Estos dispositivos vienen equipados con el hardware necesario y la potencia de cálculo que ofrecen excede la necesidades mínimas exigibles para la implementación de un *front-end*. La implementación del servidor no ofrece ningún problema ya que se puede situar en cualquier ordenador potente con cone-

xión directa a Internet en cualquier parte del mundo.

Este trabajo presenta una implementación de un reconocedor distribuido del habla sobre IP que trabaja en tiempo real. El sistema DSR se implementa como una aplicación cliente-servidor y hace uso de la interfaz de programación de aplicaciones API (*Application Programming Interface*) BSD socket para la conexión a través de la pila de protocolos de Internet. El cliente se puede instalar en cualquier computador personal que posea un equipo de audio que permita grabar la voz de un locutor. En el cliente se instala el *front-end* estándar de la ETSI, ES 201 108 [1] y la transmisión de las características de la señal de voz se transmiten de acuerdo al RFC 3557 [2].

El servidor lleva a cabo la tarea de reconocimiento del habla, tarea que se puede basar en el uso de modelos ocultos de Markov o en cualquier otra técnica. La aplicación implementada para demostrar el buen funcionamiento del sistema DSR a través de Internet es el reconocimiento de los primeros 20 dígitos más la palabra "PARA" utilizada para indicar el final de la locución al reconocedor. Para ello se ha utilizado programación dinámica mediante el algoritmo DTW (*Dynamic Time Warping*) [3]. Pero hay que dejar claro que la aplicación desarrollada permite utilizar cualquier otra técnica de reconocimiento así como realizar tareas de reconocimiento más complejas y todo ello sin que se vea afectado el cliente ni el procedimiento de transmisión.

Este artículo se organiza como se indica a continuación. En la Sección 2 se describe de forma general las características del sistema de reconocimiento. Las secciones 3 y 4 describen los detalles de la implementación del cliente y el servidor respectivamente. Los resultados de rendimiento (tasa de reconocimiento) se presentan en la Sección 5. Finalmente, la Sección 6 contiene las conclusiones de este artículo.

2. DESCRIPCIÓN DEL SISTEMA DSR

En este apartado se van a describir las características globales del sistema de reconocimiento distribuido desarrollado. El sistema desarrollado como un sistema DSR se implementa como una aplicación cliente-servidor y hace uso de la interfaz de programación de aplicaciones API (*Application Programming Interface*) BSD socket para la conexión a través de la pila de protocolos de Internet.

El diseño de la aplicación se ha realizado con el objetivo de cumplir tres objetivos básicos:

- Escalabilidad y versatilidad (posibilidad de modificar o ampliar partes del sistema sin que se vean afectadas las partes que no se cambian).
- Eficiencia (simplificar las tareas con el objetivo de lograr la mayor rapidez posible y el mayor aprovechamiento de memoria y CPU).
- Claridad en la programación (hacer una programación que dé facilidad comprensiva).

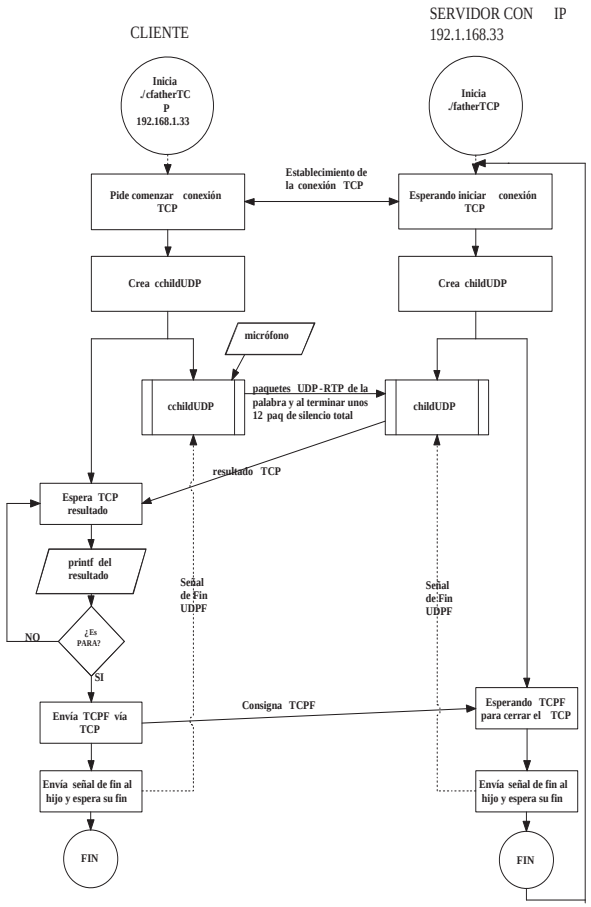


Figura 2. Esquema general del sistema cliente/servidor DSR.

Además, el sistema es capaz de trabajar tanto en el sistema operativo (SO) Windows XP como en Linux. Es posible tener el servidor trabajando en Linux y el cliente en Windows (o viceversa). Los programas han sido hechos específicamente para trabajar en Linux. La portabilidad a Windows se consigue a través de la plataforma Cygwin [4]. La captura de muestras de voz se hace a través de la librería de código abierto PortAudio [5] disponible también para varios SO (multiplataforma).

La comunicación entre el cliente y el servidor es a través de TCP y UDP. El esquema general de funcionamiento del cliente y del servidor se muestra en la figura 2. La conexión TCP sólo se utiliza para el control previo a la comunicación y el envío con seguridad de los resultados del reconocimiento del servidor al cliente. La conexión UDP, que no es segura, se utiliza para enviar/recibir las características de la señal de voz sin pausa y en tiempo real con ayuda del protocolo RTP.

Se ha utilizado multiprogramación (padre e hijo). Los padres sólo se encargan de comunicar y representar los resultados del reconocimiento vía TCP mientras que los hijos se encargan básicamente de la comunicación UDP y funcionan al ritmo del micrófono.

A continuación se indican otras características a des-

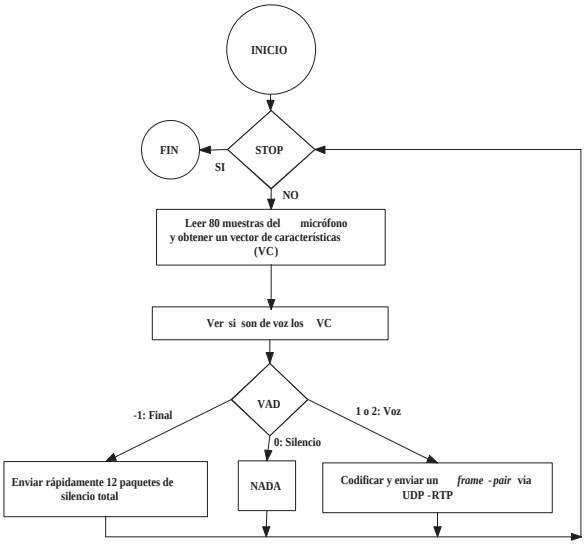


Figura 3. Organigrama del programa hijo del cliente UDP.

tacar de este sistema DSR. Utilización de un algoritmo de detección de actividad de voz, VAD (*Voice Activity Detection*), que permite reducir el número de paquetes enviados al servidor ya que sólo se envían éstos cuando el locutor está hablando. Utilización de un algoritmo de mitigación de errores que trata de mejorar la tasa de reconocimiento cuando se producen pérdidas de paquetes. Para evitar el problema de la llegada desordenada de los paquetes éstos se numeran haciendo uso de la cabecera del protocolo RTP. Por último, indicar que al hacer una programación tan modularizada se permite escalar y hacer cambios con mucha facilidad, como por ejemplo, cambiar fácilmente el reconocedor DTW por uno basado en Modelos ocultos de Markov.

3. CLIENTE DSR

Esta es la parte que se encarga de captar las muestras de voz del micrófono, extraer los vectores de coeficientes, detectar si son de voz o no y en caso de que lo sean codificarlos y enviarlos. En la figura 3 se muestra el organigrama del cliente UDP así como todas las partes que funcionan conjuntamente con éste.

Como se ha comentado antes, para facilitar la programación se utiliza la librería PortAudio que es un conjunto de APIs de sonido en código C abierto que pueden ser llamadas desde casi cualquier SO. PortAudio proporciona una serie de funciones que permiten interactuar con la tarjeta de sonido para configurar la lectura de muestras y la propia lectura.

3.1. Extracción de características

Este apartado resume el estándar de la ETSI ES 201 108 [1], que especifica el *front-end* utilizado para la extracción de las características utilizadas en el reconocedor.

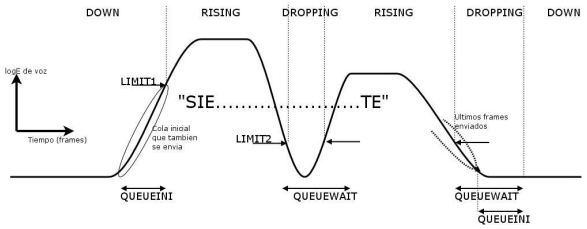


Figura 4. VAD del cliente.

Según el estándar de la ETSI, para una frecuencia de muestreo de 8000 Hz. (empleada en este trabajo) cada vector de características se forma con las 200 últimas muestras (trama de 25 ms) y se forma uno nuevo cada 80 muestras (10 ms), generando 13 coeficientes Melcepstrales, MFCC (*Mel Frequency Cepstral Coefficients*), incluido el de orden 0, además del logaritmo de la energía de dicha trama. Con el fin de cuantificar eficientemente dichos parámetros se utiliza una cuantificación del tipo SVQ (*Split Vector Quantization*), aplicando diccionarios de 64 centros para los MFCC del 1 al 12, mientras que el coeficiente cepstral de orden 0 y la energía emplean diccionarios de 256 centros. La tasa final de este estándar es de 4.4 kbps.

3.2. Detector de actividad de voz, VAD

Mediante el detector de actividad de voz del cliente se realiza una estimación de si hay señal de voz en cuyo caso se prepara una transmisión de vectores de características al servidor. En caso contrario no realiza ninguna transmisión. El funcionamiento del VAD se explica por medio de tres posibles estados (DOWN, RISING y DROPPING) en los cuales se entra según sea el valor del logaritmo de la energía (LogE) de la trama actual y circundantes. En la figura 4 se representa un ejemplo en el que se muestra el LogE de cada trama respecto al tiempo. La curva corresponde aproximadamente a la palabra SIETE.

Normalmente el estado del VAD es DOWN, es decir, que no hay voz y sí ruido. La función VAD() va revisando continuamente el valor actual LogE de la señal de voz. Cuando el frame actual supera un valor (LIMIT1) bastante alto con respecto al ruido de fondo se considera que hay voz y se pasa al estado RISING (que hay voz elevada). En el estado RISING se van transmitiendo tramas a medida que van llegando. En este estado estará hasta que la curva decrezca y se pase el límite LIMIT2. Entonces se pasa al estado DROPPING (que está desapareciendo la señal de voz). Al entrar en este estado se espera varias tramas (QUEUEWAIT=11) antes de terminar y devolver -1 indicando que la palabra ha finalizado. La utilidad de la espera final es prevenir que el VAD piense erróneamente que ha terminado la palabra por el simple hecho de que se desvanezca la señal durante unos milisegundos como por ejemplo ocurre en la ”t” de la palabra siete.

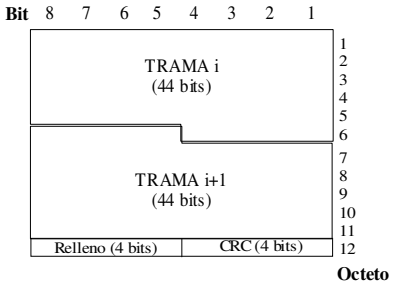


Figura 5. Formato de un *frame-pair*.

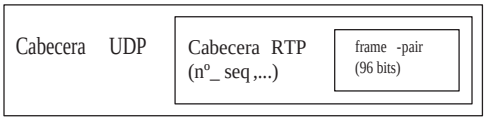


Figura 6. Encapsulamiento de un *frame-pair* vía RTP-UDP.

Finalmente una vez que se ha detectado el final de una locución se envían consecutivamente unas 12 tramas de silencio para indicar al VAD del servidor que ha terminado la palabra.

3.3. Paquetes RTP

Los vectores de coeficientes se envían de dos en dos tal como lo recomienda el estándar de la ETSI (RFC 3557). La disposición de la codificación de las dos tramas (un *frame-pair*) para su envío se muestra en la figura 5.

Como se observa, a continuación de las dos tramas codificadas se utilizan 4 bits para la comprobación de errores en el decodificador más 4 bits de relleno (tasa de bits final de 4,6 kbps). Si se detecta que los bits han sufrido alguna alteración por el camino se aplica el algoritmo de mitigación de errores. Esta comprobación se implementa a través de un código polinómico o CRC cuyo polinomio generador es el polinomio $g(X) = 1 + X + X^4$.

El *frame-pair* se transmite haciendo uso del protocolo RTP [2]. De los campos de la cabecera del protocolo RTP sólo se va a usar el campo de 'número de secuencia' para asegurar que los paquetes llegan ordenados. Posteriormente, el paquete RTP se transmite en el campo de datos de un datagrama UDP como se muestra en la figura 6.

4. SERVIDOR DSR

Dentro de un sistema DSR, el servidor es la parte encargada de llevar a cabo el reconocimiento propiamente dicho. Sin embargo, no es esta la única tarea a realizar. El programa hijo del servidor está encargado de recibir paquetes UDP, aplicarles el algoritmo de ordenación RTP, decodificarlos, aplicarles el algoritmo de mitigación de errores, realizar el reconocimiento si son paquetes de voz y enviar el resultado correspondiente. En la figura 7 se

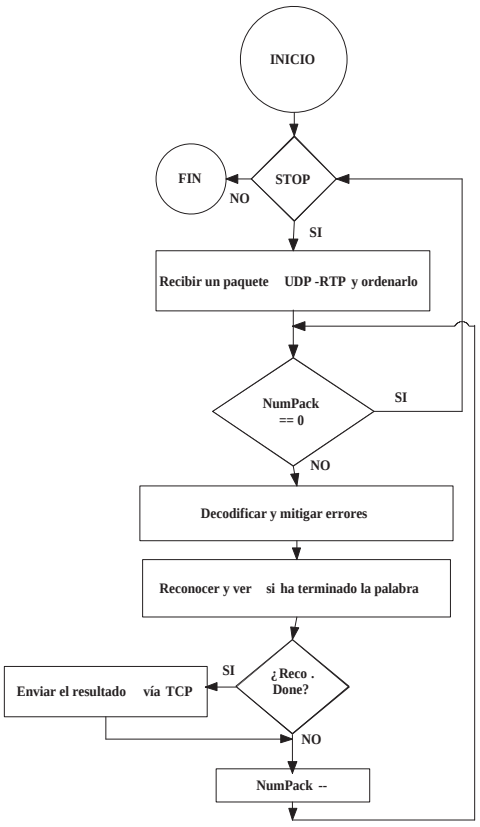


Figura 7. Organigrama del programa hijo del servidor UDP.

muestra el organigrama del servidor UDP así como todas las partes que funcionan conjuntamente con éste.

Según la figura, tras la inicialización se entra en el bucle principal en el que lo primero que se hace es esperar recibir un paquete UDP en modo bloqueante. Recibido el paquete, se le aplica el algoritmo de ordenación RTP. Este algoritmo suele devolver como salida un solo paquete RTP para ser procesado pero algunas veces puede que no devuelva ningún paquete (si ha recibido un paquete retrasado) o un número determinado de paquetes.

El paquete o paquetes devueltos se decodifican aplicándoles el algoritmo de mitigación de errores. El resultado suele ser dos tramas aunque algunas veces puede que no se devuelva nada, como es el caso en el que el *frame-pair* tenga el CRC erróneo, o un número determinado de tramas (si ha llegado un *frame-pair* con CRC bueno tras otros de CRC erróneo). El resultado pasa al detector de voz del servidor que se encarga de ir pasando tramas al reconocedor cuando detecta que son de voz. Cuando termina el reconocimiento de la palabra se envía el resultado vía TCP al cliente.

4.1. Algoritmo de mitigación de pérdidas

El problema de la transmisión a través de IP es la pérdida de paquetes. Como se ha visto, el *frame-pair* incluye un campo para la detección de errores en la transmisión.

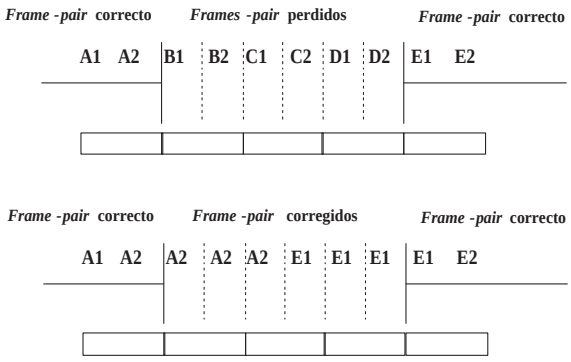


Figura 8. Ejemplo de corrección de *frames-pairs* erróneos.

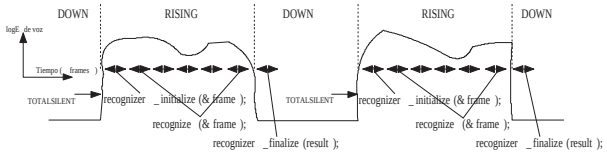


Figura 9. VAD del servidor.

Para seguir la compatibilidad con el estándar, los paquetes perdidos se van a considerar que han llegado con errores de forma que se marcan como que ha fallado la comprobación de su CRC.

Los *frame-pair* erróneos no se descartan inmediatamente si no que se van almacenando en un buffer a la espera de que llegue uno correcto con la esperanza de corregirlos. Al llegar uno correcto se ejecuta la función encargada de hacer la corrección y de devolver las tramas corregidas que habían quedado en espera en el buffer. La corrección aplicada se muestra en la figura 8. Se observa que la corrección consiste en sustituir la primera mitad de las tramas erróneas por la última trama correcta (A2) y la segunda mitad por la trama actual correcta (E1).

4.2. El reconocedor y el VAD asociado

La idea de utilizar un VAD en el reconocedor es tratar de disminuir el consumo de CPU por parte de la tarea de reconocimiento de forma que sólo se ejecuta esa tarea cuando el VAD asociado detecta *frame-pairs* que contienen voz. El funcionamiento de este VAD es más simple que el VAD del cliente. Sólo dos estados DOWN (no hay voz) y RISING (hay voz). En la figura 9 se muestra un ejemplo de funcionamiento del VAD del servidor. Si la energía de las tramas que se reciben no supera el limite TOTALSILENT se clasifican como tramas de silencio total y no se hace nada con ellas.

Como se ha comentado anteriormente, el reconocedor implementado es un reconocedor que hace uso del algoritmo DTW. Sin embargo, dada la alta escalabilidad de la programación de esta aplicación, podríamos utilizar otro reconocedor como, por ejemplo, uno basado en modelos ocultos de Markov.

5. RESULTADOS DE RECONOCIMIENTO

En este apartado se muestran los resultados de reconocimiento obtenidos en la ejecución de la aplicación DSR. Se comprueba tanto la efectividad del reconocedor como la del algoritmo de mitigación de errores que trabaja conjuntamente con el algoritmo de ordenación RTP. Se han hecho dos tipos de medidas. En un principio se han obtenido resultados de reconocimiento simulando un canal IP para varias probabilidades de pérdidas de paquete y diferentes longitudes de ráfagas.

El segundo tipo de medida de resultados se ha hecho en un entorno real. El cliente se ha ejecutado en un ordenador portátil conectado a Internet a través de una red inalámbrica WiFi. El servidor se ha instalado en un ordenador personal conectado permanentemente a Internet. Se han realizado varias pruebas de reconocimiento bajo diferentes niveles de potencia de la señal recibida, tratando situaciones en las que calidad de la señal recibida era muy buena (0 % de paquetes perdidos) a situaciones en que la calidad era muy mala (con un tanto por ciento de paquetes perdidos apreciable).

La tarea de reconocimiento para la que se han realizado las pruebas consiste en el reconocimiento monolocutor de 22 palabras (los números del CERO al VEINTE y la palabra PARA). Para obtener los datos de reconocimiento en la simulación del canal se han tomado cuatro condiciones de canal:

- Canal con una tasa de perdidas de paquetes UDP-RTP del 20 % y una la longitud media de las ráfagas perdidas de 1 paquete (condición marcada como 'Canal 20-1').
- Canal con un 30 % de perdidas y una longitud media de 2 (condición 'Canal 30-2').
- Canal con un 40 % de perdidas y una longitud media de 2 (condición 'Canal 40-2').
- Canal con un 50 % de perdidas y una longitud media de 4 (condición 'Canal 50-4').

Los resultados de reconocimiento se muestran en la figura 10. En dicha gráfica se muestran resultados de reconocimiento cuando no se utiliza algoritmo de mitigación ni tampoco el de ordenación RTP (ya que si no se introducirían paquetes falsos que empeoraría aun más el reconocimiento). Se observa un pobre resultado en el reconocimiento ya que la tasa de reconocimiento disminuye considerablemente a medida que el canal sufre un incremento en la tasa pérdidas, llegando a una tasa de reconocimiento del orden del 52,3 % (casi la mitad de las palabras fallan al ser reconocidas) bajo la ultima condición de canal ('Canal 50-4').

Cuando se utiliza el algoritmo de mitigación y el de ordenación RTP se comprueba una mejora considerable en la tasa de reconocimiento. Además, el sistema se muestra más robusto a un aumento a la tasa de pérdida de paquetes y/o la longitud de la ráfaga. Es de destacar que para

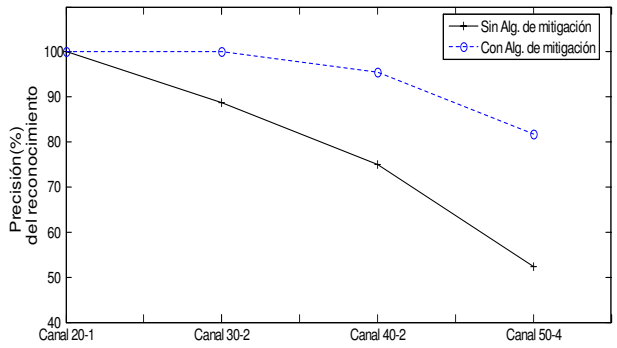


Figura 10. Resultados de reconocimiento. Canal X-Y hace referencia a un canal con una tasa de pérdidas del X % y una longitud de ráfaga de Y paquetes

la ultima condición de canal 'Canal 50-4' la tasa de reconocimiento aumenta desde el 52,3 % del caso anterior a un 81,8 %.

Las medidas de resultados en un entorno real han contado con la dificultad de tener el control sobre los errores ya que los errores dependen de la intensidad de la potencia de la señal. Estos son apreciables para una potencia baja o muy baja. Los errores que se cometen son principalmente ráfagas de paquetes perdidos más que errores puros de CRC. Los resultados también demostraron que el algoritmo de mitigación ayudaba a mejorar el reconocimiento, sobre todo cuando las ráfagas de error por palabra no eran mayores de 8 paquetes.

6. RESUMEN

En este trabajo se ha presentado una implementación de un sistema de reconocimiento distribuido del habla en tiempo real para su aplicación en un entorno de Internet. Desarrollado como una aplicación cliente-servidor, el cliente hace uso del *front-end* estándar definido por la ETSI. Incluye un detector de voz para sólo enviar información cuando el locutor habla y la transmisión de la información se hace de acuerdo al RFC 3557. El servidor realiza el reconocimiento utilizando programación dinámica, incluye a su vez un detector de voz y utiliza la técnica de mitigación de pérdida de paquetes propuesta en el estándar de la ETSI. La modularidad del diseño permitela utilización de cualquier otro reconocedor sin que por ello se vean afectados los clientes. Las pruebas reales de funcionamiento para la aplicación particular desarrollada han demostrado una alta fiabilidad en varias condiciones de transmisión.

7. BIBLIOGRAFÍA

[1] ETSI ES 201 108, "Speech processing, transmisión and quality aspects (stq); distributed speech recognition; front-end feature extraction algorithm; compression algorithms," 2000.

[2] IETF. RFC 3557, "Rtp payload format for etsi european standard es 201 108 distributed speech recognition encoding," 2003.

[3] C. S. Myers y L. R. Rabiner, "A comparative study of several dynamic time-warping algorithms for connected word recognition," *The Bell System Technical Journal*, vol. 60(7), pp. 1389–1409, September 1981.

[4] "http://www.cygwin.com/," .

[5] "http://www.portaudio.com/," .

LA CONVERSIÓN DE TEXTO EN HABLA MULTIDOMINIO: PRINCIPIOS Y PORTABILIDAD

Francesc Alías y Joan Claudi Socoró

Departamento de Comunicaciones y Teoría de la Señal
Enginyeria i Arquitectura La Salle. Universitat Ramon Llull
Pg. Bonanova 8. 08022 Barcelona
{falias,jclaudi}@salle.url.edu

RESUMEN

La conversión de texto en habla (CTH) multidominio persigue conseguir una calidad sintética cercana a la de los sistemas de CTH diseñados para un determinado ámbito o aplicación, aumentando su flexibilidad al considerar distintos dominios (estilos de locución, emociones, temáticas, etc.) para la síntesis. En este trabajo, se presentan las motivaciones de esta estrategia desarrollada como evolución paralela a los sistemas orales multidominio, junto a distintas reflexiones sobre su flexibilidad y portabilidad para el diseño de nuevos sistemas de CTH a partir de las conclusiones obtenidas hasta el momento.

1. INTRODUCCIÓN

El propósito final de todo CTH es la generación de habla sintética completamente natural a partir de un texto de entrada cualquiera. Para lograr este objetivo, históricamente, la investigación en el ámbito de la CTH ha primado la capacidad del sistema de sintetizar *cualquier* mensaje sobre la naturalidad del mismo, es decir, la *flexibilidad* de la síntesis ante su *calidad* (*Aproximación A* en la figura 1) [1, 2]. Este enfoque se debe a que, ya desde sus inicios, los sistemas de síntesis fueron capaces de generar voz razonablemente inteligible a partir de una entrada de texto no restringida [2]. En el contexto de la CTH, este proceso se ha articulado, fundamentalmente, mediante el desarrollo de sistemas de conversión de texto en habla de propósito general (CTH-PG). Sin embargo, tiempo después, aparecieron nuevas aplicaciones de la CTH con un ámbito de funcionamiento controlado o restringido (p.ej. servicios de información meteorológica, de tráfico, etc.). En este contexto, se puede conseguir una elevada naturalidad de la señal sintética a cambio de reducir la generalidad de la síntesis, al utilizar textos pertenecientes al dominio. Esta filosofía constituye la segunda línea de investigación seguida en el camino hacia la consecución de una síntesis genérica *perfecta —Aproximación B* en la figura 1, introducida en [1] y posteriormente recogida en [2]. En este caso, el camino para llegar al objetivo marcado se articula, desde un punto de vista más práctico, a través de la consecución de productos de calidad notable

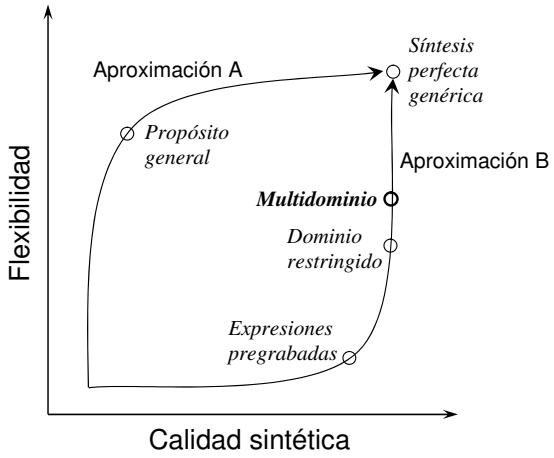


Figura 1. Aproximaciones seguidas en la investigación hacia la consecución de una CTH genérica perfecta, representadas según el binomio *calidad sintética - flexibilidad* del sistema — figura adaptada de [2].

(restringidos a aplicación) a medida que se avanza en la investigación —a diferencia de la *Aproximación A*, más teórica, donde, para obtener mejoras de calidad significativas, resulta necesario realizar grandes esfuerzos (p.ej. paso de la síntesis concatenativa basada en difonemas a síntesis basada en selección de unidades). Bajo esta segunda línea de investigación se ubican los CTH de dominio restringido (CTH-DR) (ver figura 1). Siguiendo esta segunda aproximación, se presentó la conversión de texto en habla multidominio (CTH-MD) en [3], con el objetivo de aumentar la flexibilidad de la síntesis a partir de la generalización de los resultados obtenidos mediante la síntesis de carácter restringido, manteniendo su calidad sintética. Aunque, inicialmente, esta aproximación sólo se diseñó para trabajar con subcorpus independientes bajo el marco de la CTH basada en corpus, posteriormente se observó que constituye un marco genérico para el desarrollo de cualquier tipo de sistema de CTH. En este trabajo se discute esta cuestión junto a las motivaciones de partida que dieron lugar al diseño de la nueva propuesta de filosofía de CTH desarrollada.

2. MOTIVACIÓN

En este apartado se describen los aspectos que han motivado el desarrollo de la CTH-MD respecto a la evolución de otros sistemas orales, así como las distintas propuestas de tipologías de corpus para síntesis existentes.

2.1. Sistemas orales multidominio

El desarrollo de sistemas multidominio es una de las nuevas líneas de investigación en el ámbito de los sistemas de lenguaje hablado (SLH) (o *spoken language systems*, en inglés) [4], entre los que destacan los sistemas de diálogo, los de traducción del habla y los de enrutamiento de llamadas [5]. La mayoría de los SLH, excluyendo los sistemas de dictado de propósito general, trabajan sobre un conjunto finito de dominios en los que el usuario puede hacer las consultas pertinentes a través del correspondiente sistema de reconocimiento automático del habla (RAH), p.ej. diferentes destinatarios en el enrutamiento de llamadas, distintas temáticas para los sistemas de traducción, o varios subdominios en los sistemas de diálogo complejos [6]. Conocer el dominio de la conversación permite mejorar el rendimiento y la eficacia de los módulos que conforman estos sistemas, por ejemplo, escogiendo el modelo del lenguaje más adecuado al dominio del reconocedor automático del habla, adaptando la estrategia de diálogo del gestor de diálogo si se produce un cambio de dominio dentro del discurso del usuario, o reduciendo los recursos utilizados por el SLH, al cargarlos dinámicamente según las necesidades particulares del diálogo en cada instante de la interacción (se adaptan al ámbito de la consulta o conversación) [4].

2.1.1. El RAH multidominio

En [7] se presentan dos aproximaciones distintas para la construcción de un reconocedor automático del habla multidominio (RAH-MD): *i*) trabajando con distintos RAH paralelos adaptados a dominio, seleccionando como resultado del reconocimiento aquél que aporte un mayor grado de confianza, o bien *ii*) se pueden combinar los datos de entrenamiento (léxicos, modelos de lenguaje, etc.) de los distintos reconocedores dependientes de dominio para construir una única red de búsqueda multidominio. Por un lado, la búsqueda paralela por dominio permite integrar distintos RAH ya existentes, una vez diseñados y optimizados para su dominio, restringiendo así el número de hipótesis por palabra a considerar. Sin embargo, la arquitectura en paralelo puede aumentar el coste computacional del proceso de reconocimiento del mensaje al realizar la búsqueda de la misma palabra en varias redes, complicándose la elección de la mejor hipótesis al tener que normalizar las medidas de confianza entre dominios. Por otro lado, el hecho de trabajar con una única red multidominio presenta las ventajas e inconvenientes complementarios a los de la arquitectura paralela, destacando el compromiso existente entre la reducción del coste computacional de la búsqueda y el aumento de la perplejidad en la resolución de la tarea del reconocimiento (ver [7] para más detalles).

Asimismo existe otra estrategia para el RAH-MD que consiste en dividir el proceso de reconocimiento del mensaje oral en dos fases: primero se aplica un sistema de reconocimiento *genérico* (independiente de dominio), y a continuación, se aplican los RAH entrenados sobre cada dominio (p.ej. ver [8], donde esta estrategia se emplea para sistemas de comprensión multidominio). En este contexto parece necesario incorporar un módulo de detección de dominio —generalmente, temático— para poder escoger, a partir del resultado del reconocedor genérico, el RAH dependiente de dominio más adecuado a la consulta del usuario [5].

2.1.2. Aproximaciones al corpus multidominio

Uno de los problemas fundamentales de los sistemas de CTH basados en selección de unidades (CTH-SU) es la pérdida de calidad sintética cuando el dominio del que proviene el texto de entrada no se ajusta al del corpus grabado [9], cuestión que perjudica la calidad tanto de los sistemas de CTH-PG [10, 11] como, evidentemente, la de los sistemas de dominio restringido [12]. En este contexto, se han presentado distintas aproximaciones con el objetivo de adaptar un CTH-PG a un determinado dominio, generalmente, mediante la incorporación de pequeños subcorpus dedicados a los dominios deseados [10, 11]. En estos trabajos se demuestra el efecto positivo que tiene la adecuación del corpus de voz al dominio deseado, con la consiguiente mejora de la calidad sintética obtenida.

Con este mismo objetivo, se define el concepto de *corpus de voz multidominio* [3, 13]. La idea consiste en disponer de unidades de voz correspondientes a distintos dominios (estilos de locución, temáticas, emociones, etc.) coexistiendo dentro de un mismo corpus de voz. En este ámbito, se han presentado dos estrategias distintas para el diseño de este tipo de corpus: *i)* mezclar todos los dominios en un único corpus con un contenido predominante de unidades genéricas [10, 14, 11], o *ii)* disponer de tantos subcorpus como dominios [15, 16, 17]. Estos dos enfoques —o tipologías de corpus multidominio— han sido denominados *blending* y *tiering*, respectivamente, en la literatura de síntesis basada en corpus [9]. Según las características de los estilos de locución a considerar será más conveniente utilizar una u otra tipología de corpus. Asimismo, se puede observar que los sistemas de CTH-MD siguen un enfoque similar al de los sistemas de RAH-MD.

3. ARQUITECTURA DEL SISTEMA DE CTH-MD

La investigación en el ámbito de la CTH no había incorporado, hasta nuestro trabajo, la filosofía multidominio incorporada en otros sistemas orales de interacción persona-máquina, en su camino hacia mejorar la naturali-

dad y la usabilidad de los mismos. Esta situación ha sido motivada, fundamentalmente, por dos cuestiones. Primero, el hecho de que los CTH fueron capaces de abordar la síntesis de propósito general desde sus inicios, a diferencia de los sistemas de RAH, que tuvieron que restringir ya de entrada el dominio de funcionamiento para ser eficientes; y segundo, debido al papel secundario que ha tenido la CTH en el contexto de los SLH multidominio, donde la CTH podría tomar en consideración el dominio de la conversación para adaptar el mensaje de salida del sistema (selección de dominio supervisada), pero que, en general, ha sido utilizada como simple medio de transmisión de la información consultada por el usuario (síntesis genérica), por lo que tampoco se ha abordado la CTH multidominio.

A grandes rasgos, la introducción de esta nueva estrategia de síntesis multidominio en los sistemas de CTH implica, básicamente:

- Disponer de una arquitectura de CTH flexible, que permita incorporar, por un lado, cualquier estrategia de síntesis, y por otro, seleccionar la estructura y el contendido del corpus de voz según las necesidades del sistema o aplicación en la que el CTH-MD se enmarque.
- Incorporar información del dominio del texto de entrada mediante un módulo de clasificación de dominio, con el objetivo de mejorar la flexibilidad de la síntesis, sin perder calidad de los mensajes sintéticos (equivalente a la de los CTH-DR).

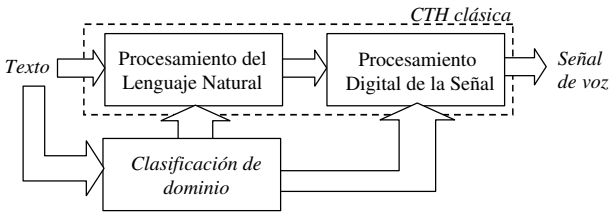


Figura 2. Diagrama de bloques de la arquitectura de un conversor de texto en habla multidominio con clasificación automática de dominio.

4. DESIGNACIÓN AUTOMÁTICA DE DOMINIO

La CTH-MD se asienta sobre la idea que la calidad de la CTH se puede mejorar si se conoce cuál es la *forma* más adecuada de pronunciar los textos a sintetizar [9] (p.ej. en [17] se escogen los textos que formarán parte del corpus de voz primando su facilidad para inducir la emoción deseada). Es decir, no todos los textos pueden ser pronunciados de cualquier forma (estilo, emoción, énfasis,...), ya que, por un lado, existen mensajes cuyo significado hace que sea inapropiado pronunciarlos de una determinada manera [18] (p.ej. órdenes militares *vs.* tristeza o miedo [15], mensajes conceptualmente complejos *vs.* voz de niño [9]), y por otro lado, existen mensajes que presentan una clara correlación con el modo de locución a utilizar [17] (p.ej. mensajes positivos o negativos *vs.* a unos patrones prosódicos determinados [14, 19], o frases más típicas de un niño o de una niña [9]). No obstante, no hay que olvidar que también existen mensajes que, según el contexto de la comunicación en el que se emitan, pueden cambiar de significado [16] (p.ej. "*Veo que hay mucha comida en la nevera*", en tono alegre o sarcástico). En este caso, escoger el modo de locución más apropiado para el mensaje pasa por disponer de información paralingüística (estado de humor del hablante, intencionalidad del mensaje, relación entre los interlocutores, ...), así como algunos parámetros extralinguísticos (edad, sexo, personalidad, ...), que también pueden afectar a la comunicación [20], cuestiones que quedan fuera del alcance del presente trabajo de investigación.

4.1. A partir de voz (en RAH-MD)

En el contexto de los sistemas multidominio guiados por voz, el módulo de detección y asignación automática de dominio (generalmente, una temática) a partir de las locuciones del usuario toma un papel relevante [4]. Conocer la temática de la consulta permite mejorar la precisión del sistema de reconocimiento, ya que permite escoger el modelo de lenguaje mejor adaptado a la tarea, con lo que se reduce tanto la perplejidad del modelo como la tasa de error de reconocimiento o WER [5].

La detección automática del dominio del mensaje en el ámbito de los SLH se realiza a partir de una selección de dominio *explícita*, mediante un conjunto predefinido de comandos (palabras clave), o bien, deduciendo los cambios de dominio de forma *implícita*, a partir de las hipótesis de reconocimiento recogidas durante la interacción del usuario con el sistema [7, 4, 5]. Cabe añadir que, a diferencia de los sistemas de detección de temática que trabajan con grandes volúmenes de datos (p.ej. artículos periodísticos, noticias o transcripciones de las mismas), en el contexto de interacción persona-máquina, normalmente se trabaja con un volumen mucho más reducido de datos [5].

4.2. A partir de texto (en CTH-MD)

De forma paralela, en el contexto de la CTH-MD también resulta necesario disponer de algún módulo que indique al CTH el dominio más adecuado sobre el que llevar a cabo el proceso de síntesis del texto de entrada. Este módulo puede ser externo al CTH, mediante una selección manual [16] o supervisada, como por ejemplo, en sistemas de diálogo —suelen trabajar con un conversor de concepto en habla (*concept-to-speech*, en inglés) [2]— o algunas aplicaciones audiovisuales con mensajes sintéticos controlados [15].

Asimismo, este proceso puede formar parte del sistema de CTH-MD mediante la inclusión de un módulo de clasificación automática de textos, como se propuso en [3]. Las particularidades de la tarea —textos cortos y bajo coste computacional— han guiado el desarrollo de

una nueva propuesta de sistema de clasificación de textos adaptada al problema de la CTH-MD (ver [21, 22] para más información). Fundamentalmente, la propuesta incorpora información temática y estilística de los textos (p.ej. las coocurrencias entre las palabras, los signos de puntuación, etc.) con el objetivo de modelar completamente el texto a sintetizar (típicamente de longitud muy corta, p.ej. 1 frase). Esta propuesta está claramente influenciada por la estrategia de síntesis utilizada, en este caso, la síntesis basada en corpus, siendo además el corpus multidominio implementado según la estrategia *tiering*. En el caso de trabajar con estrategias de síntesis más flexibles (p.ej. basadas en HMM), no resultaría tan crítico que el sistema de clasificación tomara en consideración todas y cada una de las palabras del texto.

5. PORTABILIDAD DE LA PROPUESTA

Aunque, inicialmente, la filosofía de CTH-MD se definió sólo pensando en el ámbito de la síntesis sobre distintos dominios independientes [3, 13], posteriormente se observó que constituye un marco genérico para el desarrollo de cualquier tipo de sistema de CTH. A continuación se presentan algunas ideas que describen la portabilidad de la propuesta más allá de lo presentado en los trabajos desarrollados hasta el momento.

5.1. Arquitectura flexible

La flexibilidad de esta arquitectura permite abordar desde la CTH-PG (un único dominio genérico), pasando por la CTH-DR (un único dominio restringido), hasta la CTH para distintos dominios (estructurados o no jerárquicamente, según su contenido y/o sus características acústicas). Estos dominios pueden ser incorporados explícitamente como subcorpus independientes (p.ej. [17, 9]), o como pequeños apéndices acompañando a un corpus genérico (p.ej. [10, 11, 14, 23]), así como, para dividir un mismo corpus (con la misma calidad vocal) en distintas temáticas (p.ej. periodísticas: *política, sociedad, cultura, deportes, . . .*, entre otros). Por otro lado, esta arquitectura también permite la coexistencia de filosofías de síntesis distintas, desde la síntesis basada en corpus (donde esta estructura se puede replicar y/o profundizar tanto como se quiera, siempre que el subdominio tenga suficientes unidades para la síntesis), pasando por la síntesis basada en modelos ocultos de Markov (con modelos adaptados a cada dominio), hasta soluciones híbridas (p.ej. síntesis genérica más subdominio adaptado a tarea, síntesis genérica más transformación de voz, entre otras). Simplemente, el CTH deberá tener las herramientas necesarias para abordar de forma eficiente la gestión de los datos con los que se diseñe.

Asimismo, por el momento, la designación de dominio provoca elegir un subcorpus u otro dentro del corpus multidominio (estrategia *tiering*). Sin embargo, como se acaba de comentar, la arquitectura diseñada permite disponer de corpus multidominio mixtos en los que se mezcle o complemente un corpus con un dominio u otro de forma que, para un determinado texto de entrada, el módulo de selección escoja la mejor secuencia de unidades considerando toda la información contenida en el corpus (estrategia *blending*). En este contexto, se puede realizar el proceso de selección de unidades mediante la incorporación de pesos relacionados con el grado de pertenencia del texto a sintetizar respecto a cada dominio —con una filosofía similar a la descrita en [14, 23]—, flexibilizando la preselección de las unidades que implica la selección de dominio según la estrategia *tiering*.

5.2. Aplicabilidad de la detección de dominio

Por el momento, la detección automática de dominio ha sido sólo utilizada para escoger el modelo prosódico y el subcorpus correspondientes al dominio identificado. No obstante, el hecho de poder determinar el dominio del texto de entrada permite mejorar la manera de sintetizarlo desde otros módulos del sistema de CTH, como por ejemplo: *i)* ayudando a desambiguar el mensaje en la fase de normalización del texto (p.ej. si se determina que el texto de entrada pertenece al dominio *matemático*, el texto $3/4$ se deberá convertir en "*tres cuartos*", evitando transcribirlo como "*tres de abril*"); *ii)* considerando varios perfiles prosódicos en la búsqueda (p.ej. como en [24]); *iii)* guiando el proceso de selección de unidades mediante los pesos pertinentes (en una estrategia de CTH-SU con un corpus genérico más varios subcorpus ad hoc a dominio, p.ej. [14]); *iv)* controlando las modificaciones a realizar por el módulo de procesamiento digital de la señal según el dominio del texto de entrada (p.ej. existen estilos de locución con calidades vocales particulares, donde grandes modificaciones de la señal pueden empeorar claramente la calidad sintética [25]); *v)* activando el módulo de transformación de voz para generar el estilo del dominio detectado (si el CTH dispone de un corpus genérico más un postprocesamiento mediante conversión de voz), etc. Según las características y particularidades del CTH sobre el que se incorpore el módulo de clasificación de dominio, éste tendrá un impacto mayor o menor en el funcionamiento del sistema de síntesis.

5.3. Optimización del proceso de selección

La propuesta de CTH-MD permite abordar el problema de la optimización del coste computacional de los sistemas de CTH basados en selección de unidades desde un punto de vista diferente al convencional en el contexto. Según la implementación de la CTH-MD *tiering*, al mismo tiempo que se escoge el dominio (subcorpus) más adecuado para realizar la síntesis, se consigue reducir el coste computacional del proceso de selección [3]. Esto es debido a que, una vez escogido uno de los subcorpus del corpus multidominio *tiering*, el conjunto de unidades sobre el que se realiza la búsqueda es de tamaño considerablemente menor respecto al total de unidades presentes

en el corpus —el coste computacional del proceso de clasificación es mínimo, como se demuestra en [21].

5.4. Enriquecimiento del análisis del texto de entrada

En el mismo camino en el que se encuentra la filosofía de CTH-MD descrita, han aparecido, recientemente, distintos trabajos en el ámbito de la investigación en tecnologías del habla que, mediante un análisis del texto más allá del típico para la CTH (del que se encarga el módulo de procesamiento del lenguaje natural —ver figura 2), pretenden dotar de mayor información al sistema de síntesis con el objetivo de mejorar la calidad sintética de salida. En este ámbito destacan los trabajos que pretenden estimar, a partir del texto a sintetizar, la actitud o postura del autor [19] o la emoción subyacente en el mensaje [26, 23], con el objetivo de mejorar la naturalidad de la síntesis. En [19] se demuestra la correlación entre las variaciones prosódicas y la aparición de adjetivos que expresan una actitud positiva o negativa del mensaje con mayor o menor intensidad —regulada mediante los adverbios que los acompañan. En [23] se diseña un corpus con tres emociones (neutra, alegre y enfado) a partir de 400 frases extraídas de textos periodísticos, agrupadas en un único corpus (estrategia *blending*). Mediante un diccionario (*Dictionary of Affect*) se determina el grado de emotividad de cada palabra y se ajusta la función de coste para realizar la búsqueda de unidades según el tipo y/o grado de emotividad de la palabra. Como conclusión del trabajo, se demuestra la viabilidad subjetiva de la propuesta y se indica que cuanto mayor sea el número de unidades emotivas presentes en la frase sintetizada, mayor percepción de esa emoción tendrá el usuario.

En el ámbito de los CTH dependientes de aplicación, destaca la investigación en el ámbito de la lectura de cuentos infantiles [26, 27]. En estos trabajos se pretende determinar la emoción más adecuada a cada pasaje del cuento a partir de las palabras y la estructura del texto (p.ej. longitud de la frase, análisis de las dependencias, puntuación de la frase, etc.), incorporando al análisis del texto conocimiento externo al mismo (p.ej. WordNet). Por otro lado, se encuentran [28] y [29], que incorporan una red semántica y una red de conocimiento genérico (*common sense*) al problema, respectivamente —ver [27], para más detalles sobre la detección de emociones a partir de texto.

6. DISCUSIÓN

No hay que olvidar que el objetivo final de la propuesta es obtener voz sintética de alta calidad. Hasta el momento, las pruebas realizadas (a nivel de frase) para estudiar la propuesta de CTH-MD se han sustentado en un elemento clave: la correlación entre el dominio del texto y el estilo de locución utilizado [30]. Debido a las características vocales de los dominios del corpus de voz utilizado (organizados según la estrategia *tiering)*, se ha escogido en tiempo de ejecución un dominio u otro para llevar a cabo la síntesis (evitando mezclar unidades en la síntesis con calidades vocales muy distintas), seleccionando la secuencia de unidades con menor número de concatenaciones a partir de una función de coste muy simple. Esta relación es fundamental para que la tarea del clasificador de textos propuesto seleccione automáticamente uno u otro estilo de locución a partir del texto de entrada. Por lo tanto, cabe diferenciar entre los textos que pueden mapearse directamente a un estilo de locución determinado, de los textos que no tienen una relación tan directa, como se ha comentado en la introducción del presente capítulo.

Tomando esta cuestión en consideración, en [30] se observa que cuando la CTH-MD funciona adecuadamente (es decir, su calidad es equivalente a la de un CTH-DR), los resultados obtenidos demuestran la mejora de la flexibilidad manteniendo una elevada calidad en la síntesis, gracias a adaptar el dominio de la selección de unidades —junto a la prosodia— al dominio del corpus al que mejor se ajusta el texto de entrada. Estos resultados se encuentran en la línea de otros trabajos, como los de [10, 11, 14], en los que la síntesis obtenida al adaptar la CTH al dominio del texto de entrada mejora claramente la obtenida a partir de sistemas de CTH-PG. Por otro lado, cuando el clasificador de textos asigna el texto de entrada a un dominio distinto al que fue grabado (error de clasificación), los resultados obtenidos muestran la dificultad de la clasificación de estas frases. Concretamente, los evaluadores muestran un criterio de selección mucho más vago que en el primer caso, ya que, en este caso, se está comparando el resultado de dos síntesis de dominio restringido, por lo que la selección se basa en lo apropiado que es el estilo al mensaje a transmitir —cuestión que pertenece más a un plano cognitivo que a un ámbito de calidad sintética.

En un futuro se pretende realizar un nuevo conjunto de pruebas que permitan comparar la calidad obtenida por la estrategia actual con la conseguida al trabajar con todas las unidades en un mismo corpus multidominio (estrategia *blending*), seleccionando las unidades incorporando información sobre el dominio en la función de coste considerada —al estilo de lo descrito en [23].

7. CONCLUSIONES

De algún modo, se puede argumentar que la CTH-MD permite dar un paso más en la convergencia de los sistemas de RAH y los de CTH, que se ha venido observando en la última década [31], en este caso en el desarrollo de sistemas orales multidominio. En este trabajo, una vez detalladas las motivaciones de la propuesta, se ha descrito cómo la filosofía de CTH-MD permite ajustar el sistema de CTH a las características de los datos disponibles, las necesidades del sistema de síntesis (orientado a aplicación, con varios dominios, con enfoque de propósito general más adaptación a un dominio, etc.) y las estrategias de síntesis más adecuadas, en lugar de tener que crear un nuevo sistema de CTH para cada nueva aplicación.

De este modo, la CTH-MD consigue integrar en un mismo CTH dominios que deben estar explícitamente representados en el corpus para ser reproducidos fidedignamente [9] (p.ej. determinadas emociones como la alegría, según [32] o estilos de locución con características vocales propias, según [25, 20]), junto a otros dominios que pueden ser generados sintéticamente de forma *bastante* realista (p.ej. tristeza desde un corpus neutro mediante modificaciones prosódicas [32], o incorporando pequeños subcorpus a un corpus de propósito general de tamaño mayor, p.ej. buenas o malas noticias [14]).

Por lo tanto, pensamos que la CTH-MD puede constituir un marco genérico para el desarrollo de futuras aplicaciones donde intervenga la síntesis de voz, incorporando o no (según el caso) la detección automática de dominio, adaptada a las características de los datos de trabajo.

8. BIBLIOGRAFÍA

[1] J. Yi y J. Glass, "Natural-sounding speech synthesis using variable-length units," in *Proceedings of ICSLP*, Sydney, Australia, 1998, pp. 1167–1170.

[2] P. Taylor, "Concept-to-Speech synthesis by phonological structure matching," *Philosophical Transactions of the Royal Society, Series A*, vol. 356, no. 1769, pp. 1403–1416, 2000.

[3] F. Alías, I. Iriondo, y P. Barnola, "Multi-domain text classification for unit selection Text-to-Speech Synthesis," in *The 15th International Congress of Phonetic Sciences (ICPhS)*, Barcelona, 2003, pp. 2341–2344.

[4] K. Rüggenmann y I. Gurevych, "Assigning domains to speech recognition hypotheses," in *Proceedings of HLT-NAACL Workshop on Spoken Language Understanding for Conversational Systems and Higher Level Linguistic Information for Speech Processing*, Srinivas Bangalore y Hong-Kwang Jeff Kuo, Eds., Boston, Massachusetts, USA, 2004, pp. 70–77, ACL.

[5] I.R. Lane, T. Kawahara, T. Matsui, y S.Ñakamura, "Dialogue Speech Recognition by Combining Hierarchical Topic Classification and Language Model Switching," *IEICE Transactions on Information and Systems*, vol. E88D, no. 3, pp. 446–454, 2005.

[6] I.R. Lane, T. Kawahara, T. Matsui, y S.Ñakamura, "Out-of-domain detection based on confidence measures from multiple topic classification," in *Proceedings of ICASSP*, Montreal, Canadá, 2004, vol. 1, pp. 757–760.

[7] T.J. Hazen, I.L. Hetherington, y A. Park, "FST-Based Recognition Techniques for Multi-Lingual and Multi-Domain Spontaneous Speech," in *Proceedings of EuroSpeech*, Aalborg, Dinamarca, 2001, vol. 2, pp. 1591–1594.

[8] Grace Chung, *Towards multi-domain speech understanding with exible and dynamic vocabulary*, Ph.D. thesis, Massachusetts Institute of Technology, Junio 2001.

[9] A.W. Black, "Unit Selection and Emotional Speech," in *Proceedings of EuroSpeech*, Geneve, Suiza, 2003, pp. 1649–1652.

[10] M. Chu, C. Li, P. Hu, y E. Cahng, "Domain adaptation for TTS systems," in *Proceedings of ICASSP*, Orlando, USA, 2002, vol. 1, pp. 453–456.

[11] V. Fischer, J. Botella, y S. Kunzmann, "Domain Adaptation Methods in The IBM trainable Text-To-Speech System," in *Proceedings of ICSLP*, Jeju Island, Corea del Sur, 2004, pp. 1165–1168.

[12] W. Hamza y J.F. Pitrelli, "Combining the flexibility of speech synthesis with the naturalness of pre-recorded audio: a comparison of two approaches to phrase-splicing TTS," in *Proceedings of InterSpeech*, Lisboa, Portugal, 2005, pp. 2585–2588.

[13] F. Alías, X. Sevillano, P. Barnola, y J.C. Socoró, "Arquitectura para conversión texto-habla multidominio," *Procesamiento del Lenguaje Natural*, vol. 31, pp. 83–90, Septiembre 2003.

[14] W. Hamza, R. Bakis, E. M. Eide, M. A. Picheny, y J. F. Pitrelli, "The IBM Expressive Speech Synthesis System," in *Proceedings of of ICSLP*, Jeju Island, Corea del Sur, 2004, pp. 2577–2580.

[15] W.L. Johnson, S.Ñarayanan, R. Whitney, R. Das, M. Bulut, y C. LaBore, "Limited domain synthesis of expressive military speech for animated characters," in *Proceedings of IEEE Workshop on Speech Synthesis*, Santa Monica, USA, 2002, pp. 163 – 166.

[16] N. Campbell, "What type of inputs will we need for Expressive Speech Synthesis?," in *Proceedings of 2002 IEEE Workshop on Speech Synthesis*, Santa Monica, USA, 2002.

[17] A. Iida, N. Campbell, F. Higuchi, y M. Yasumura, "A corpus-based Speech Synthesis System with Emotion," *Speech Communication*, vol. 40, no. 1,2, pp. 161–187, 2003.

[18] J. Yamagishi, K. Onishi, T. Masuko, y T. Kobayashi, "Acoustic Modelling of Speaking Styles and Emotional Expressions in HMM-based Speech Synthesis," *IEICE Transactions on Information and Systems*, vol. E88D, no. 3, pp. 502–509, 2005.

[19] Y. Sagisaka, T. Yamashita, y Y. Kokenawa, "Generation and perception of F_0 markedness for communicative speech synthesis," *Speech Communication*, vol. 46, no. 1, pp. 376–384, 2005.

[20] N. Campbell, "Developments in Corpus-Based Speech Synthesis: Approaching Natural Conversational Speech," *IEICE Transactions on Information and Systems (Invited paper)*, vol. E88D, no. 3, pp. 376–383, Marzo 2005.

[21] F. Alías, X. Gonzalvo, X. Sevillano, J.C. Socoró, J.A. Montero, y D. García, "Clasificación de Textos Adaptada para Conversión de Texto en Habla Multidominio," *Procesamiento del Lenguaje Natural*, vol. 37, pp. 267–274, Setiembre 2006.

[22] F. Alías, X. Sevillano, y J.C. Socoró, "Text Classification based on Associative Relational Networks for Multi-Domain Text-to-Speech Synthesis," in *SIGIR-2006 Workshop on Stylistics for Text Retrieval in Practice*, Seattle, USA, Agosto 2006.

[23] G. Hofer, K. Richmond, y R.A.J. Clark, "Informed blending of databases for emotional speech synthesis," in *Proceedings of InterSpeech*, Lisboa, Portugal, 2005, pp. 501–504.

[24] F. Campillo y E. Rodríguez Banga, "Combined prosody and candidate unit selections for corpus-based Text-to-Speech systems," in *Proceedings of ICSLP*, Denver, USA, 2002, vol. 1, pp. 141–144.

[25] O. Turk, M. Schröder, B. Bozkurt, y L.M. Arslan, "Voice quality interpolation for emotional Text-to-Speech synthesis," in *Proceedings of InterSpeech*, Lisboa, Portugal, 2005, pp. 797–800.

[26] C. Ovesdotter, D. Roth, y R. Sproat, "Emotions from text: machine learning for text-based emotion prediction," in *Proceedings of HLT/EMNLP*, Vancouver, Canadá, 2005, pp. 579–586.

[27] V. Francisco y P. Gervás, "Análisis de dependencias para la marcación de cuentos con emociones," *Procesamiento del Lenguaje Natural*, vol. 37, pp. 137–144, Setiembre 2006.

[28] Z.-J. Chuang y C.-H. Wu, "Emotion recognition from textual input using an emotional semantic network," in *Proceedings of ICSLP*, Denver, USA, 2002, pp. 2033–2036.

[29] H. Liu, H. Lieberman, y T. Selker, "A model of textual affect sensing using real world knowledge," in *Proceedings of ICSLP*, Miami, USA, 2003, p. 125132.

[30] F. Alías, J.C. Socoró, X. Sevillano, I. Iriondo, y X. Gonzalvo, "Multi-domain Text-to-Speech Synthesis by Automatic Text Classification," in *Proc. of InterSpeech*, Pittsburgh, USA, 2006.

[31] M. Ostendorf y I. Bulyko, "The impact of speech recognition on speech synthesis," in *Proceedings of 2002 IEEE Workshop on Speech Synthesis*, Santa Monica, USA, 2002.

[32] I. Iriondo, F. Alías, J. Melenchón, y M. A. Llorca, "Modeling and Synthesizing Emotional Speech for Catalan Text-to-Speech Synthesis," *Tutorial and Research Workshop on Affective Dialog Systems*.

PONDERACIÓN DE LAS PROBABILIDADES DE OBSERVACIÓN DE LOS HMM PARA VERIFICACIÓN DE LOCUCIONES EN SISTEMAS DE RAH

Marta Casar, José A. R. Fonollosa

Centre TALP, Universitat Politècnica de Catalunya
C. Jordi Girona 1-3, 08034, Barcelona
email: {mcasar,adrian}@gps.tsc.upc.edu

RESUMEN

Las tecnologías de reconocimiento del habla requieren sistemas fiables capaces de funcionar correctamente en diferentes tareas y entornos. Al no ser posible conseguir un reconocimiento libre de errores en la mayoría de casos, surge la necesidad de disponer de una medida de la fiabilidad del sistema. En este artículo se propone la ponderación de la probabilidad de observación de los modelos de Markov, equilibrando la diferencia entre rangos dinámicos de las probabilidades de transición y de observación. El parámetro definido *OPW* es independiente de la gramática utilizada, a diferencia de otras soluciones similares. Dado que se espera que las variaciones introducidas por este nuevo parámetro tengan mayor influencia en las palabras incorrectamente reconocidas, se propone un sistema de verificación de locuciones basado en una 'segunda opinión'. Para ello se utiliza la salida de decodificar utilizando diferentes $OPW \neq 1$. Resultados experimentales conseguidos hasta el momento muestran la validez de la propuesta.

1. INTRODUCCIÓN A LOS HMM

La mayoría de sistemas de reconocimiento del habla actuales se basan en un conjunto de modelos estadísticos (también llamados modelos acústicos) que relacionan las características observables de la señal de voz con el conjunto de unidades fonéticas teóricas. La implementación más usual de estos modelos estadísticos se basa en la utilización de los Modelos Ocultos de Markov (HMM) [1].

Los HMM resultan una potente herramienta estadística para modelar señales de voz. Consisten en una cadena de Markov, la probabilidad de salida de la cual es una variable aleatoria X generada mediante una función de probabilidad de salida asociada a cada estado. Así pues, no existe una correspondencia unitaria entre la secuencia de observaciones y la secuencia de estados, de forma que esta secuencia de estados permanece 'oculta'. Formalmente, un HMM se define como:

$$\lambda = (A, B, \pi)$$

Este trabajo ha sido parcialmente subvencionado por el MCyT a través del proyecto TIN-2005-08852

siendo A la distribución de probabilidad de transición entre estados $A = a_{ij}$, B la distribución de probabilidad de observación de símbolos en el estado j, $B = b_j(k)$, y π la distribución de probabilidad inicial para cada estado $\pi = \pi_i$.

En procesado del habla, la secuencia de estados subyacente asociada a los modelos HMM se caracteriza por su configuración temporal (izquierda-a-derecha): a medida que se incrementa la variable temporal, el índice de estados se incrementa a su vez o bien se mantiene igual. En un instante de tiempo, cada trama de voz puede estar en un único estado.

En HMM semicontinuos (SCHMM) cada estado del modelo contiene un conjunto de funciones de probabilidad de emisión, generalmente funciones densidad de probabilidad de mezclas de gaussianas. Para cada estado, estas funciones proporcionan la probabilidad con la cual ese estado puede generar cada trama. Las funciones de densidad de probabilidad están ligadas de forma conjunta para todos los modelos, formando un codebook que permite mapear el vector de características de entrada continuo X y transformarlo en O_k. A partir de este vector, podremos utilizar una distribución de probabilidad de salida discreta $b_j(k)$.

La forma más intuitiva de calcular la probabilidad $P(\mathbf{O}|\lambda)$ de la secuencia de observaciones $\mathbf{O} = (O_1, O_2, \ldots, O_T)$ dado un HMM λ, consiste en sumar las probabilidades de todas la secuencias de estados posibles:

$$P(\mathbf{O}|\lambda) = \sum_{allS} P(S|\lambda)P(\mathbf{O}|S,\lambda) \qquad (1)$$

Para una secuencia de estados particular $S = (s_1, s_2, \ldots, s_T)$, la probabilidad de salida conjunta se puede expresar como:

$$P(\mathbf{O}|S,\lambda) = \prod_{i=1}^{T} P(O_t|s_t,\lambda) = b_{s_1}(O_1)b_{s_2}(O_2)\ldots b_{s_T}(O_T)$$

y así, podemos reescribir la ecuación (1):

$$P(\mathbf{O}|\lambda) = \sum_{allS} a_{s_0 s_1} b_{s_1}(O_1)\ldots a_{s_{T-1}s_T} b_{s_T}(O_T) \quad (2)$$

2. PONDERACIÓN DE LAS PROBABILIDADES DE OBSERVACIÓN DE LOS MODELOS

En la mayoría de sistemas de reconocimiento del habla, la decodificación consiste en un proceso de búsqueda de la secuencia de palabras $\hat{\mathbf{W}} = \mathrm{w}_1\mathrm{w}_1 \dots \mathrm{w}_m$ de máxima probabilidad a posteriori $P(\mathbf{W}|\mathbf{O})$ correspondiente a una determinada observación acústica $\mathbf{O} = O_1 O_2 \dots O_n$. Expresado formalmente:

$$\begin{aligned}\hat{\mathbf{W}} &= \underset{\mathrm{w}}{\operatorname{argmax}} P(\mathbf{W}|\mathbf{O}) = \underset{\mathrm{w}}{\operatorname{argmax}} \frac{P(\mathbf{W})P(\mathbf{O}|\mathbf{W})}{P(\mathbf{O})} \\ &= \underset{\mathrm{w}}{\operatorname{argmax}} P(\mathbf{W})P(\mathbf{O}|\mathbf{W}) \end{aligned} \quad (3)$$

donde $P(\mathbf{W})$ representa la probabilidad del modelo de lenguaje, y $P(\mathbf{O}|\mathbf{W})$ la probabilidad del modelo acústico.

A partir de esta ecuación parece que ambas probabilidades puedan ser combinadas mediante una simple multiplicación. Sin embargo, al trabajar con los HMM como modelos acústicos las probabilidades acústicas suelen estar subestimadas (ver [1]). Al combinarlas con las probabilidades de los modelos de lenguaje, se acaba dando a estos modelos un peso insuficiente. Además, los rangos dinámicos de ambas probabilidades difieren al trabajar con modelos continuos o semicontinuos. Por consiguiente, es práctica corriente ponderar la probabilidad del modelo de lenguaje mediante un peso LW (*language model weight*) determinado de forma empírica (típicamente > 1), transformando $P(\mathbf{W})$ en $P(\mathbf{W})^{LW}$.

De forma equivalente, la idea de ponderar el valor de las probabilidades de observación de los HMM surge de la definición (y construcción) de estos modelos, centrándonos en la contribución de los diferentes tipos de probabilidades. Como ya se ha señalado en la introducción, se puede entender la probabilidad de una secuencia de observaciones como la suma de contribuciones de las probabilidades de transición entre estados y observación de cada estado. La ecuación (2) se puede reescribir para una determinada secuencia S_i mediante logaritmos:

$$\begin{aligned} logP(\mathbf{O}|\lambda) &= log(a_{s_0 s_1} b_{s_1}(O_1) \dots a_{s_{T-1} s_T} b_{s_T}(O_T)) \\ &= P_A + P_B \end{aligned}$$

$$\begin{aligned} \text{donde} \quad P_A &= log(a_{s_0 s_1} a_{s_1 s_2} \dots a_{s_{T-1} s_T}) \\ P_B &= log(b_{s_1} b_{s_2} \dots b_{s_T}) \end{aligned}$$

Las probabilidades de transición están asociadas a la gramática utilizada, de forma que la relación entre los rangos dinámicos de P_A y P_B se puede ver desequilibrada por restricciones impuestas por dicha gramática. Introduciendo un factor de ponderación de las probabilidades de observación *OPW* (*Observation Probability Weight*) la contribución de las probabilidades de observación de los símbolos ser verán modificadas, pasando a ser $\mathrm{OPW} \cdot P_B$. *OPW* será determinado de forma empírica con el objetivo de corregir la diferencia de rangos dinámicos que pudiese existir entre ambas probabilidades. *OPW* se asemeja así al factor *LW* presentado, con la diferencia de ser independiente de la gramática.

Obsérvese que este desarrollo sirve tanto para HMM semicontínuos (para los que se ha desarrollado la experimentación) como para modelos discretos o continuos.

3. RECONOCIMIENTO DEL HABLA UTILIZANDO OPW

Partiendo de un conjunto de HMMs convencionales se han ponderado las probabilidades de observación de estos modelos utilizando el factor *OPW* definido. Los modelos resultantes han sido utilizados para el reconocimiento de cadenas de dígitos. Las pruebas de reconocimiento realizadas nos han permitido estudiar la viabilidad de utilizar una decodificación alternativa basada en *OPW* para verificación de locuciones.

3.1. Sistema de reconocimiento del habla

El sistema de reconocimiento del habla utilizado para estos experimentos es RAMSES [2], basado en SCHMM. Las principales características de este sistema son:

- La señal de voz es enventanada cada 10ms con ventanas de 30ms de longitud. Cada trama se parametriza utilizando coeficientes cepstrales melfrequency (MFCC) y su primera y segunda derivadas, junto a la primera derivada de la energía.
- Los parámetros espectrales son cuantificados utilizando 512 centroides, y la energía utilizando 64.
- Las unidades acústicas utilizadas en los HMM son semidígitos. Se entrenan 40 modelos de semidígitos, además de un modelo ruidoso para cada dígito, modelados mediante 10 estados. Se utilizan también modelos de silencio y de relleno, modelados mediante 8 estados.
- Se utiliza un algoritmo de Viterbi en decodificación, implementando búsqueda en haz para limitar el número de caminos. Para ello, las tramas se cuantifican utilizando 6 centroides para los parámetros espectrales, y 2 para la energía.

3.2. Bases de datos

Dos bases de datos diferentes han sido utilizadas a lo largo de estos experimentos. Primero, a partir de la base de datos en español de SpeechDat [3] se ha creado un subconjunto de entrenamiento con 11443 frases y un subconjunto de test con 3405 frases, todas ellas formadas por secuencias de dígitos. Los resultados obtenidos a partir de este primer subconjunto de test se han utilizado para seleccionar (de forma empírica) el valor óptimo de *OPW*.

A continuación, los nuevos modelos se han puesto a prueba utilizando una base de datos independiente, que llamaremos DigitVox, obtenida mediante una aplicación telefónica de reconocimiento de voz. Esta base de datos contiene 5317 frases correspondientes a DNIs (cadenas de 8 dígitos) grabadas en condiciones ruidosas. Estos experimentos tienen como objetivo probar la independencia de los modelos obtenidos, trabajando en condiciones

Configuración	Tasa de reconocimiento de FRASES	Tasa de reconocimiento de PALABRAS	Sustituciones	Inserciones	Borrados
modelos originales (*OPW* =1)	93.304 %	98.73 %	0.24 %	0.97 %	0.06 %
OPW = 0.2	93.831 %	99.09 %	0.23 %	0.40 %	0.28 %
OPW = 0.5	93.511 %	98.76 %	0.23 %	0.84 %	0.17 %
OPW = 2.0	90.856 %	98.26 %	0.25 %	1.45 %	0.03 %

Tabla 1. Tasas de reconocimiento con diferentes valores de *OPW* en la ponderación de las probabilidades de observación

'reales'. Además, se pretende comprobar si los resultados obtenidos para ciertos valores de *OPW* en los primeros tests se deben al sobreentrenamiento o adaptación de los modelos, que serian efectos a evitar.

3.3. Resultados

A partir de los HMM originales entrenados utilizando RAMSES, se han construido nuevos conjuntos de modelos acústicos con diferentes valores de *OPW* entre 0.2 y 5. Los resultados obtenidos nos han permitido analizar la contribución del peso de las probabilidades de observación a las prestaciones de los modelos.

Los modelos ponderados han sido evaluados utilizando DigitVox. La tabla 1 resume los resultados obtenidos, comparándolos con los modelos originales (*OPW* = 1).

Estos resultados muestran una mejora importante para valores de $OPW < 1$, confirmando la hipótesis de que existe un desequilibrio entre las probabilidades de observación y las probabilidades de transición. Cabe recordar que el objetivo de esta propuesta no es la mejora en los resultados de reconocimiento por si misma, sino la obtención de una segunda opinión en vistas a implementar un sistema de verificación de locuciones.

4. VERIFICACIÓN DE LOCUCIONES

En todo sistema de reconocimiento hay un cierto grado de incertidumbre inherente a la decodificación obtenida. Por tanto, parece necesario obtener una medida que nos informe de la correspondencia entre la secuencia de palabras resultante y la señal de voz de entrada. Esta medida nos permitirá decidir si la salida obtenida se considera correcta o incorrecta, de una forma fiable.

Los sistemas estadísticos de reconocimiento del habla se basan en la probabilidad a posteriori $P(\mathbf{W}|\mathbf{O})$ de una palabra $\mathbf{W}$ dada una secuencia de observaciones acústicas $\mathbf{O}$. La secuencia de palabras W_{opt} que maximiza esta probabilidad a posteriori también minimiza la probabilidad de error en la frase reconocida. Por tanto, una estimación precisa de $P(\mathbf{W}|\mathbf{O})$ (ver eq. 3) nos proporciona una medida de confianza fiable, representando el ratio entre la probabilidad asociada a la hipotesis obtenida, $P(\mathbf{W})P(\mathbf{O}|\mathbf{W})$, y la probabilidad acústica. No obstante, en la mayoría de implementaciones se tiende a evitar esta costosa estimación teórica, definiendo otros parámetros y umbrales de aceptación que determinen cuándo aceptar o rechazar la salida del decodificador.

La validez de los resultados de reconocimiento ha sido un tema ampliamente estudiado en la literatura de reconocimiento del habla y enfocado desde diferentes perspectiva, ya sea utilizando modelos de hipótesis alternativas, algoritmos N-best, o grafos de palabras [4]. Uno de los principales problemas a solucionar consiste en hallar las características o parámetros que aportan más información sobre el reconocimiento o las locuciones a verificar. En este sentido, un parámetro muy útil ha sido el *LMJitter* [5] (también llamado *Acoustic Stability* [6]), que se basa en la hipótesis de que las palabras mal reconocidas son más sensibles a las variaciones del *Grammar Scale Factor* (o *LW*, según se le ha llamado en el apartado 2) [7].

Uno de los hándicaps de implementar el *LMJitter* es el elevado coste computacional que representa, para lo cuál se han propuesto algunos algoritmos eficientes [7]. Además, la implementación de sistemas de verificación basados en medidas de confianza suele realizarse utilizando varias de ellas y combinando las salidas mediante algoritmos de decisión, de forma que aumenta la complejidad del sistema.

En lugar de la utilización del *LMJitter* como una medida de confianza, otra opción anteriormente planteada consiste en modificar el peso LM entre decodificaciones. En [8] los autores realizan múltiples procesos de reconocimiento aplicando diferentes pesos para ponderar las probabilidades de los modelos de lenguaje frente las probabilidades de los modelos acústicos. En [6], esta estrategia se compara con un sistema basado en obtener las N-best soluciones de un proceso de reconocimiento trabajando con word-lattices, obteniendo prestaciones casi similares a la vez que se reducía el coste computacional.

No obstante, el coste computacional no representa un obstáculo insalvable, de forma que la utilización de un doble reconocimiento se plantea como una solución eficaz. Nuestro enfoque se centra en la hipótesis de que, igual que ocurre con el *LMJitter*, las palabras incorrectamente reconocidas serán más sensibles a las variaciones del factor de ponderación de las probabilidades de observación definido (*OPW*). La propuesta presentada consiste en desarrollar un sistema de verificación usando una estrategia basada en una 'segunda opinión' [9]. A partir de las decodificaciones obtenidas por dos sistemas de reconocimiento el sistema decidirá sobre la validez de la frase reconocida por el sistema de referencia, utilizando como segunda opinión la decodificación proporcionada por los modelos ponderados con $OPW \neq 1$.

El reconocimiento de cadenas de dígitos, que continúa siendo una aplicación de gran interés práctico, se presenta

Sistema y configuración	Exactas	Errores	Detectadas	Rechazadas	Garbage	TRR	FRR	CER
phone-based filler models	92.55 %	1.28 %	0.30 %	1.02 %	4.85 %	80.09 %	1.09 %	2.30 %
verificación OPW, *OPW* = 0.2	92.57 %	0.79 %	0.49 %	0.73 %	5.36 %	88.10 %	0.78 %	1.52 %
verificación OPW, *OPW* = 0.5	93.19 %	1.15 %	0.19 %	0.11 %	5.36 %	82.84 %	0.12 %	1.26 %

Tabla 2. Resumen de los resultados de verificación con filtrado a nivel de frase

como una primera tarea sencilla a la que enfrentarse para probar la validez de nuestra propuesta.

4.1. Verificación basada en una segunda opinión

En nuestro día a día, cuando no estamos seguros de la validez de una hipótesis concreta buscamos una segunda opinión que nos haga sentir más seguros. Ello nos puede conducir a una mayor inseguridad si la nueva opinión difiere de la primera, e incluso inducirnos a un error si ambas apuntan a una misma dirección incorrecta. Por ello, resulta imprescindible tener una cierta confianza en la fiabilidad de nuestra segunda fuente.

Como ya hemos indicado, la arquitectura de verificación propuesta se basa en una doble decodificación: una primera utilizando HMM convencionales (sin ponderación), y otra utilizando modelos ponderados. Comparando ambas salidas, las frases serán clasificadas como "aceptadas" si hay consenso, o "rechazadas" en caso contrario. Para evaluar la fiabilidad de esta decisión, se etiquetan las frases resultantes en cuatro categorías: *exacta* cuando es una frase correctamente aceptada, *error* cuando ha sido incorrectamente aceptada, *detectada* si se trata de una frase incorrecta que ha sido rechazada, y *falso rechazo* si es una frase rechazada que era correcta. Para ello, las dos decodificaciones se clasifican previamente como correctas o incorrectas mediante un alineado. Así, las fases exactas son aquellas correctamente reconocidas por ambos sistemas; errores las incorrectamente reconocidas por ambos; detectadas las frases incorrectamente reconocidas por el sistema de referencia; y falso rechazo aquellas incorrectamente reconocidas utilizando los modelos ponderados.

Puesto que la salida del reconocedor consiste en una cadena de palabras reconocidas, puede realizarse un primer "filtrado a nivel de frase" previamente a la comparación entre ambas decodificaciones. Se obtiene así una primera decisión de aceptación/rechazo de las hipótesis en base a diferentes reglas (p.ej. longitud de la frase, presencia de palabras no presentes en el vocabulario, etc.). Aquellas frases rechazadas en función de esta decisión se etiquetarán como *garbage*.

4.2. Sistema de referencia de verificación

Con la finalidad de evaluar los posibles beneficios aportados por nuestra propuesta, ésta se ha comparado con un algoritmo de verificación estándar basado en modelos de fonemas de relleno (o *phone-based filler models* [10]). Este método consiste en normalizar las probabilidades de salida del reconocedor mediante una decodificación basada en fonemas independientes, sin utilizar gramáticas ni otros modelos de lenguaje. Una vez normalizadas, dichas probabilidades se convierten en una medida de la calidad del reconocimiento al proporcionar una estimación de la semejanza, a nivel acústico, entre los modelos de fonemas y las modelos utilizados para el reconocimiento.

Los modelos de fonemas de relleno han demostrado un mejor comportamiento que otras soluciones independientes del vocabulario, como serían modelos de relleno de palabras, o anti-modelos [11]. Otras soluciones más complejas presentes en la literatura, como modelos de transformación de parámetros, o lattice-based combination models [6], superan en eficiencia estos métodos, aunque al coste de ser más dependientes de la tarea y el entorno. Nuestro objetivo, en cambio, es definir una solución de verificación de baja complejidad e independiente de la aplicación.

4.3. Resultados de verificación

Comencemos por definir [7] *TRR (True Rejection Rate)*, o tasa de rechazos correctos, como la relación entre el número de hipótesis incorrectas que han sido detectadas por la verificación, y el número total de hipótesis incorrectas: $TRR = D/I$. De forma similar, *FRR (False Rejection Rate)*, o tasa de falsos rechazos, es la relación entre el número de hipótesis correctas rechazadas y el número total de hipótesis correctas: $FRR = R/C$.

A partir de estos conceptos, se define el *CER (Classification Error Rate)*, o tasa de errores de clasificación, como el porcentaje de hipótesis incorrectamente clasificadas por el sistema de verificación respecto al número total de hipótesis reconocidas (ver [7, 4]). Se expresa en función de los dos tipos de errores que se pueden cometer: clasificar una frase correctamente reconocida como incorrecta (*R*), y clasificar una hipótesis incorrectamente reconocida como correcta (*I* - *D*).

$$CER = \frac{R + (I - D)}{I + C} \cdot 100$$

Los resultados obtenidos en verificación utilizando DigitVox se resumen en la tabla 2. La relación entre los parámetros C, D, I y R, y las etiquetas utilizadas por nuestro sistema de verificación es la siguiente: R = *rechazadas*, C = *exactas* + *R*, D = *detectadas* (o D = *detectadas* + *garbage* si se implementa filtrado a nivel de frases), y I = *errores* + D. Podemos observar como la solución basada en una segunda opinión utilizando *OPW* supera claramente los resultados de referencia, tanto considerando valores absolutos (tasa de reconocimientos exactos) como medidas relativas (valores TRR y FRR).

En términos de los parámetros *TRR* y *FRR*, se define la curva ROC (Receiver Operating Characteristic) [12] como la representación gráfica de la relación entre TRR y FRR para cada umbral utilizado, con FRR en el eje x. La figura 1 muestra la representación de la ROC para nuestro sistema de verificación y para el sistema de referencia. Se observa de nuevo la mejora introducida por el sistema basado en una segunda opinión, demostrándose el interés de la propuesta para su utilización en tecnologías de reconocimiento de voz.

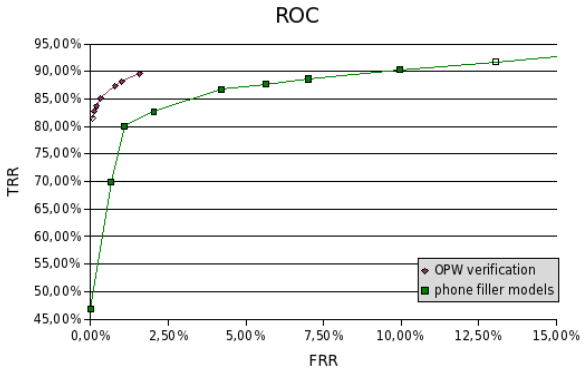


Figura 1. TRR vs. FRR

5. CONCLUSIONES

La mayoría de implementaciones estándar de HMMs para reconocimiento del habla utilizan un factor de ponderado de la gramática (o *LW*) para equilibrar la diferencia de rangos dinámicos entre las probabilidades acústicas y del modelo de lenguaje. En este artículo se propone la implementación de un factor de ponderado de las probabilidades de observación (*OPW*) con el objetivo de suavizar la diferencia en orden de magnitud entre las probabilidades de observación y de transición entre símbolos de los HMM. Los valores óptimos de *OPW* para reconocimiento del habla se han determinado de forma empírica como $OPW < 1$, de forma que se mejoran los resultados de reconocimiento de referencia obtenidos mediante HMM normales (sin ponderar).

A partir de la hipótesis de que las palabras incorrectamente reconocidas serían más sensibles a las variaciones de *OPW* (como ocurre con *LW*) se propone la implementación de un sistema de verificación de locuciones utilizando un sistema basado en una segunda opinión. Así, la etapa de verificación consistirá en comparar dos decodificaciones: una de referencia, resultado del reconocimiento con modelos no ponderados, y la segunda obtenida utilizando los HMM ponderados, considerando diferentes $OPW \neq 1$.

Observando los valores TRR y FRR de los resultados obtenidos, así como su representación gráfica mediante la curva ROC, se observa una importante mejora de nuestra propuesta en comparación con los resultados obtenidos utilizando modelos basados en fonemas de relleno. Podemos concluir, pues, que el sistema de reconocimiento y verificación propuesto se perfila como una solución interesante para tecnologías basadas en el reconocimiento del habla.

6. BIBLIOGRAFÍA

[1] X. Huang, A. Acero and H.W. Hon, *Spoken Language Processing*, Prentice Hall PTR, 1st edition, 2001.

[2] A. Bonafonte et alter, "RAMSES: el sistema de reconocimiento del habla continua y gran vocabulario desarrollado por la UPC," *VIII Jornadas de Telecom I+D*, 1998.

[3] A. Moreno, R. Winksky, "Spanish fixed network speech corpus," *SpeechDat Project. LRE-63314*, 1999.

[4] F. Wessel, R. Schlüter, K. Macherey and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 3, pp. 288–298, March, 2001.

[5] L. Chase, "Word and acoustic confidence annotation for large vocabulary speech recognition," *Proceedings of European Conf. on Speech Technology (EUROSPEECH)*, pp. 815–818, 1997.

[6] T. Schaaf and T. Kemps, "Confidence measures for spontaneous speech recognition," *Proceedings of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. II, pp. 875–878, 1997.

[7] A. Sanchis, "Tesis doctoral. estimación y aplicación de medidas de confianza en reconocimiento automático del habla," *Univ. Politécnica de Valencia. Dep. Sistemas Informáticos y Computación*, 2004.

[8] D. Willett, A. Worm, C. Neukirchen and G. Rigoll, "Confidence measures for hmm-based speech recognition," *Proceedings of IEEE Int. Conf. on Spoken Language Processing (ICSLP)*, 1998.

[9] G. Hernández-Ábrego and J. B. Mariño, "A second opinion approach for speech recognition verification," *Proc. of the VIII SNRFAI*, pp. 85–92, 1999.

[10] S. R. Young, "Detecting misrecognitions and out-of-vocabulary words," *Proceedings of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. I, pp. 21–24, 1994.

[11] L. Jiang and X.D. Huang, "Vocabulary-independent word confidence measure using subword features," *Proceedings of IEEE Int. Conf. on Spoken Language Processing (ICSLP)*, 1998.

[12] J.P. Egan, *Signal detection theory and* ROC *analysis*, Academic Press, 1975.

RENDIMIENTO PERCEPTUAL Y RECONOCIMIENTO CON CODIFICADORES VOIP SOBRE REDES DE PAQUETES

José L. Carmona, Antonio M. Peinado, José L. Perez Córdoba, Ángel M. Gómez, Victoria Sánchez

Dpto. Teoría de la Señal, Telemática y Comunicaciones, Universidad de Granada

RESUMEN

El objetivo de este trabajo es llevar a cabo una exploración sobre la degradación introducida por una red IP sobre la calidad perceptual de la voz decodificada y el reconocimiento a partir de ésta. Destaca la evaluación de los codificadores más empleados en este entorno: G.729, G723.1 e ILBC, así como la amplia casuística de simulaciones de canal, caracterizadas por el porcentaje de pérdida de paquetes y el tamaño medio de las ráfagas. La evaluación de los codificadores se realizó en dos planos: por un lado, se analizó la calidad perceptual de la voz reconstruida mediante el sistema PESQ, mientras que por otro se estudió la eficiencia del sistema de reconocimiento Aurora-2 a partir de la voz sintetizada. Finalmente, se presentan interesantes comparativas de la eficiencia de los codificadores, poniendo de relevancia y justificando sus debilidades y fortalezas.

1. INTRODUCCIÓN

Este trabajo establece una comparativa del rendimiento de distintos codificadores VoIP (*Voice over IP*), con el fin de encontrar un codec que obtenga resultados óptimos tanto en la calidad de la voz reconstruida como en el RAH (Reconocimiento Automático del Habla) a partir de ésta. Esta topología de reconocimiento remoto recibe el nombre de arquitectura NSR (*Network Speech Recognition*), caracterizándose por situar el sistema reconocedor en su totalidad en el lado servidor.

Como alternativa se encuentra la arquitectura DSR (*Distributed Speech Recognition*), mostrada en la figura 1.a, que fracciona el sistema de reconocimiento entre el cliente y el servidor, de modo que la extracción de características es llevada a cabo por el primero y el reconocimiento en sí (comparación de patrones y decisión), por el segundo.

En la arquitectura NSR, la voz es codificada mediante un codec convencional y enviada hasta el extremo del servidor (véase figura 1.b) donde es posible llevar a cabo tareas RAH o reproducir la voz con otros fines. Por este motivo, evaluar la arquitectura NSR en IP exige evaluar los codificadores en reconocimiento y calidad perceptual, sometidos a las degradaciones propias de esta red. Hay

Este trabajo ha sido financiado por el proyecto MEC/FEDER TEC2004-03829.

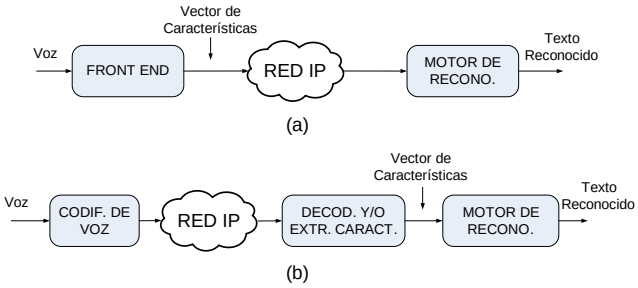


Figura 1. *Arquitecturas de reconocimiento remoto: (a) Topología DSR, (b) Topología NSR.*

que tener en cuenta que los codificadores tradicionales de voz tienen ciertas desventajas en el reconocimiento automático del habla, ya que no se encuentran diseñados a tal efecto. En este caso las redundancias de la voz relativas a la identificación del locutor, que no son necesarias en las tareas de reconocimiento, se encuentran aún en la voz codificada, lo que hace que los *bit-rates* sean mayores en las topologías NSR. Además, el procesado realizado por los codecs introduce ciertas distorsiones sobre la voz decodificada que son perjudiciales en las tareas de reconocimiento.

A pesar de las desventajas de la arquitectura NSR, ésta sigue presentando aspectos de elevado interés. Por un lado, supera las dificultades de implantación de los sistemas DSR, puesto que opera con los codecs convencionales de voz, mientras que al tener éstos como objetivo transmitir la señal con la máxima calidad perceptual posible, incluyen características del locutor que permiten su identificación. Esta última característica permitiría la implementación de servicios tan interesantes, desde el punto de vista comercial, como la firma de contratación de servicios por voz o la autoría de operaciones bancarias. Considérese que aunque existen sistemas DSR que incluyen parámetros relativos al locutor, como la frecuencia de *pitch*, la naturalidad de la voz sintetizada será menor y, por tanto, los procesos de identificación del locutor menos eficientes.

Obviamente, el rendimiento de la arquitectura NSR vendrá determinado por la robustez del codificador de voz y la calidad perceptual de ésta frente a las degradaciones introducidas por la red IP. He aquí la razón que motivó el desarrollo de este estudio comparativo, en el que

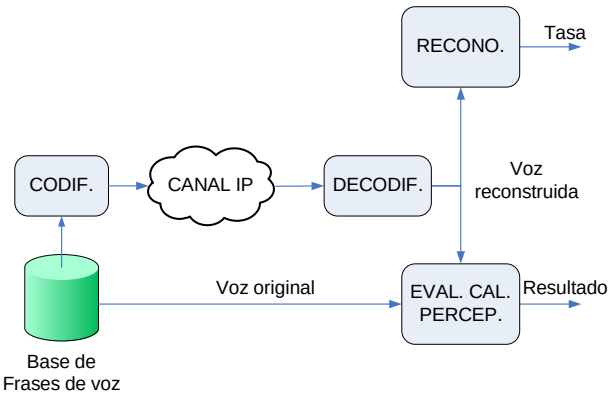


Figura 2. *Esquema general de las pruebas realizadas.*

se explora el comportamiento de los codificadores más utilizados en la actualidad en función de la degradación de la red, considerada ésta como ráfagas de paquetes perdidos.

2. ESQUEMA GENERAL

La estructura del experimento desarrollado se basa en la simulación de un sistema NSR en el que se irán sustituyendo las partes codificadora y decodificadora por los distintos codecs a evaluar. Una vez la información es codificada se segmenta en paquetes, para posteriormente pasar a la simulación de la condición de canal correspondiente. Esta simulación consistirá en marcar un conjunto de paquetes como perdidos, de modo que en el lado receptor éstos no sean utilizados para la síntesis de la señal de voz. En su lugar, cada uno de los codificadores utilizará su algoritmo PLC (*Packet Loss Concealment*) mediante el cual sustituirá la información perdida por aproximaciones derivadas de los paquetes correctamente recibidos con anterioridad. A medida que esta aproximación se desvíe de la información original, la voz sintetizada presentará mayor degradación.

Tal y como se observa en la figura 2, la degradación de la voz reconstruida en el extremo receptor se mide mediante dos sistemas de evaluación: un sistema RAH y un sistema evaluador de la calidad perceptual. El motivo por el que se han utilizado estos dos sistemas de evaluación se debe a que en los sistemas NSR sería posible reconstruir la voz con dos fines bien distintos, por un lado para ser reconocida por un sistema automático y por otro para ser escuchada por un oyente humano o realizar procesos de identificación del locutor. Bajo otra perspectiva, podría decirse que el sistema reconocedor mide la inteligibilidad del mensaje, mientras que el evaluador de la calidad perceptual tendría además en cuenta parámetros relacionados con la naturalidad del habla.

3. SELECCIÓN DE CODECS

Antes de comenzar a detallar los distintos elementos del experimento, es necesario seleccionar qué codificadores serán sometidos a prueba. A continuación se presentan las razones que justifican cada selección, así como las principales características de los codificadores elegidos.

3.1. G.729

Desarrollado por la ITU (*International Telecommunication Union*) y amparado por la recomendación H.323 que define los protocolos necesarios para proveer sesiones de comunicación audio-visual sobre redes de conmutación de paquetes, y que ha ido evolucionando para dirigir las crecientes necesidades de VoIP. Concretamente, G.729 presenta un bit-rate de 8 kbps (aunque posteriormente se han añadido anexos que ofrecen tasas de 6.4 y 11.8 kbps) y, según la recomendación H.323 se adecúa perfectamente a aquellos tipos de aplicaciones basadas sólo en audio y con un bajo requerimiento de ancho de banda. El principio de funcionamiento de este codificador es CS-ACELP (*Conjugate Structure - Algebraic CELP*) y actúa sobre tramas de voz de 10 ms, presentando un retardo algorítmico de 15 ms, ya que para el procesado de una trama precisa de los 5 ms iniciales de la siguiente (*look-ahead* de 5 ms). La definición de la carga útil del protocolo RTP viene definida, para los codificadores de la ITU, por el RFC 3551 que permite la inclusión de una o más tramas por paquete.

3.2. G.723.1

Al igual que G.729, G.723.1 es un estándar desarrollado por la ITU e incluido en la recomendación H.323, con la diferencia de presentar éste dos modos de trabajo. Su esquema de funcionamiento se encuentra basado en ACELP pero, a diferencia de AMR, el tamaño de trama utilizado es de 30 ms. Este codificador lleva a cabo un *look-ahead* de 7.5 ms, presentando, por tanto, un retardo algorítmico de 37.5 ms. Los dos modos de operación se denominan por sus *bit-rates*: 5.3 y 6.3 kbps.

3.3. iLBC

Nuevo codificador diseñado para presentar un óptimo rendimiento en las redes de conmutación de paquetes. En este caso, hace uso de una exclusiva estructura, distinta de la CELP, que codifica parte de la excitación (*start state*) mediante un codificador ADPCM, utilizando posteriormente ésta como diccionario adaptativo para la codificación del resto de la excitación de la trama. La codificación ADPCM condiciona valores altos de *bit-rate*, proporcionando dos modos, 13.33 y 15.2 kbps, con tamaños de trama de 30 y 20 ms respectivamente. En este caso, el codificador no lleva a cabo *look-ahead* en su procesado, limitándose el retardo algorítmico al tamaño de trama utilizado. La creación del RFC 3951, que define

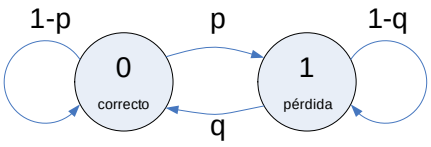


Figura 3. *Modelo de Gilbert.*

las especificaciones de este codec, y el RFC 3952, que marca la distribución de la carga útil del protocolo RTP, han hecho que este codec se difunda rápidamente en aplicaciones de VoIP. La actual predilección en su uso por sistemas tan extendidos como *Google Talk* y *Skype* [1], así como su caracter libre de derechos de autor y su diseño orientado directamente hacia VoIP [2], lo hacen un serio aspirante a convertirse en un codec predominante en este entorno.

De este modo, el conjunto de codecs y modos seleccionado queda de la siguiente manera: G.729 e iLBC {modos 13.33 y 15.2 kbps}. El número de tramas por paquete fue definido del siguiente modo: 1 trama/paquete para iLBC y G.723.1, mientras que 2 y 3 tramas/paquete para G.729. Es necesario remarcar que en el desarrollo de este trabajo cada codec emplea su propio algoritmo de mitigación de pérdidas, basados todos en ellos en técnicas de repetición y *muting*.

4. EMULADOR DEL CANAL IP

El uso de modelos de Markov se adapta bien al caracter rafagueante de las pérdidas [3], ya que son capaces de capturar la dependencia temporal de éstas. Concretamente, en este trabajo fue empleado el más simple de ellos, el modelo de dos estados: recepción correcta (estado 0) o pérdida de paquete (estado 1), presentados en la figura 3. El modelo queda totalmente definido por las probabilidades de transición p y q, tal y como se muestra en la figura.

A partir de estas probabilidades pueden determinarse tanto la probabilidad no condicional de pérdidas ulp (*unconditional loss packet*), como la probabilidad condicional de pérdidas clp (*conditional loss packet*):

$$\begin{array}{c} ulp = \frac{p}{p+q} \\ clp = 1 - q \end{array} \qquad (1)$$

donde ya puede identificarse que la probabilidad de pérdidas total P_{per} del modelo se corresponde con ulp, mientras que el tamaño medio de ráfaga L_{raf} vendrá determinado por clp a través del cálculo:

$$L_{raf} = \sum_{i=1}^{\infty} i \cdot (1 - clp) \cdot clp^{i-1} = \frac{1}{q} \qquad (2)$$

Siguiendo este modelo, se decidió evaluar un amplio abanico de condiciones de canal que vienen determinadas por una probabilidad de pérdidas y un tamaño medio de ráfaga. Concretamente, se realizaron 30 simulaciones de canal, más el canal limpio (sin pérdida de paquetes), obtenidas de evaluar el modelo con la selección de los parámetros L_{raf} y $P_{per}(\%)$ de los conjuntos $[1\ 2\ 4\ 8\ 12\ 16]$ y $[10\ 20\ 30\ 40\ 50]$, respectivamente.

5. PROCESOS DE EVALUACIÓN

En este apartado se describen los métodos de evaluación utilizados (reconocedor y evaluador perceptual) y se presentan las características de la base de datos de voz utilizadas para llevar a cabo las pruebas.

5.1. Sistema de Reconocimiento de Voz

El reconocedor es el provisto por Aurora-2 [4] y utiliza un modelo HMM (*Hidden Markov Model*) continuo de 16 estados para cada una de las once palabras que reconoce (más el silencio y la pausa, que tienen 3 y 1 estados, respectivamente), con 6 gausianas por estado. Tanto el entrenamiento como el conjunto de datos (base de datos de frases) sobre el que realizar la prueba son extraídos de la base de datos Aurora-2. La fase de entrenamiento se realiza sobre una base de 8440 frases en limpio (no contienen ruido), mientras que la prueba se lleva a cabo con el conjunto *test A* que se encuentra formado por 4004 frases en limpio distribuidas en 4 subconjuntos. El vocabulario está formado por los 11 dígitos comprendidos entre el 0 y el 9 (el cero presenta dos descripciones sonoras 'zero' y 'o'), y la duración media aproximada de las frases (dígitos conectados) es de 1.5 s.

La precisión de reconocimiento es medida mediante la tasa de palabras reconocidas (*word accuracy*, W_{acc}), que establece la relación entre el número de palabras correctamente reconocidas y el número total de palabras.

5.2. Evaluación de la Calidad Perceptual

El método más fiable y extendido para la medición de la calidad perceptual es la escala MOS (*Mean Opinion Score*), donde la calificación MOS de una señal se calcula como la media de las valoraciones dadas por los oyentes. Sin embargo, reunir a un número de oyentes tal y como se especifica en la prueba [5] es ciertamente difícil y, cuanto menos, costoso.

El algoritmo PESQ, desarrollado por la ITU en el año 2001, establece un método automático para la evaluación de la calidad perceptual. Puesto que este algoritmo presenta una exactitud aceptable en la estima de la calidad perceptual en un entorno de pérdida de paquetes para codificadores de voz [6], y dado que existe una correspondencia, mediante el documento [7], entre la nota PESQ y la puntuación MOS, se decidió emplear esta herramienta para la evaluación de la calidad perceptual. Aunque como se ha mostrado existe una correspondencia entre la puntuación PESQ y la MOS, es necesario exponer

que los resultados de este estudio son mantenidos como nota PESQ.

La duración media de las frases del *test A* de Aurora, supone un problema para la evaluación de este algoritmo, ya que, según el documento [6], no se encuentra preparado para evaluar frases de tan corta duración. Las longitudes aconsejadas se encuentran comprendidas en el rango de 8 a 20 s, por lo que se decidió agrupar las frases originales en grupos de 7 (resultando 572 frases), incrementando la duración media aproximada a 12 s, con una duración mínima de 7.5 s y máxima de 19.5 s.

Una vez obtenidos los resultados del algoritmo PESQ para los 572 archivos de la base de datos en cada condición de canal, es necesario promediarlos para obtener el resultado final. A tal efecto, se decidió realizar la media ponderada ($\bar{x}$) en función de la longitud de cada archivo:

$$\bar{x} = \sum_i w_i \cdot x_i \tag{3}$$

$$w_i = \frac{l_i}{L} \quad , \quad L = \sum_i l_i \tag{4}$$

donde x_i y l_i hacen referencia a la nota PESQ y la longitud del archivo i, respectivamente, mientras que w_i es el peso de ese archivo.

6. RESULTADOS EXPERIMENTALES

Debe de tenerse en cuenta en este apartado el resultado en limpio obtenido a partir de las frases de voz limpia $W_{acc} = 99{,}02\,\%$, que establece una cota superior en el desarrollo de las pruebas. Además, hay que hacer notar que los resultados PESQ se encuentran comprendidos en el rango de -0.5 a 4.5, siendo este último el valor obtenido por el algoritmo cuando evalúa la voz limpia.

6.1. Resultados en condición de canal limpio

Codec	*G.723.1*		*G.729*	*iLBC*	
Bit-rate (kbps)	5.3	6.3	8	13.2	15.2
$W_{acc}(\%)$	98.17	98.53	98.64	98.76	98.92
PESQ	3.69	3.80	3.95	3.92	3.94

Tabla 1. *Tabla comparativa de W_{acc} y PESQ obtenidos para los distintos codificadores con canal limpio.*

6.2. Resultados sobre canal con pérdidas

6.2.1. Resultados de Reconocimiento

Tasa de pérdidas	*Long. ráfaga*					
	1	*2*	*4*	*8*	*12*	*16*
10 %	95.90	89.67	87.37	86.60	86.46	86.49
20 %	91.26	80.04	76.82	73.93	74.39	74.57
30 %	85.00	69.74	64.73	62.67	62.61	63.24
40 %	76.70	58.37	53.62	52.46	50.28	51.69
50 %	66.05	46.74	41.62	42.35	41.88	41.42

Tabla 2. *Resultados de $W_{acc}(\%)$ con el codec G.729 (8 kbps), empaquetando 2 tramas (2 x 10 ms) por paquete.*

Tasa de pérdidas	*Long. ráfaga*					
	1	*2*	*4*	*8*	*12*	*16*
10 %	82.45	76.03	74.67	73.86	74.64	74.71
20 %	72.57	64.72	62.58	62.19	62.67	62.80
30 %	64.12	52.34	51.69	51.87	50.63	51.26
40 %	57.00	40.96	39.41	41.00	41.72	42.11
50 %	57.00	40.96	39.41	41.00	41.72	42.11

Tabla 3. *Resultados de $W_{acc}(\%)$ con el codec G.729 (8 kbps), empaquetando 3 tramas (3 x 10 ms) por paquete.*

Tasa de pérdidas	*Long. ráfaga*					
	1	*2*	*4*	*8*	*12*	*16*
10 %	93.84	89.46	86.96	85.93	86.15	86.12
20 %	88.31	80.51	75.61	73.68	73.88	74.09
30 %	81.28	70.71	64.23	61.82	62.60	62.23
40 %	72.51	59.47	53.49	50.77	50.09	51.06
50 %	62.32	48.42	41.15	40.11	40.24	40.80

Tabla 4. *Resultados de $W_{acc}(\%)$ con el codec G.723.1 (5.3 kbps).*

Tasa de pérdidas	*Long. ráfaga*					
	1	*2*	*4*	*8*	*12*	*16*
10 %	94.46	89.85	87.56	86.32	86.40	86.32
20 %	89.32	81.16	76.25	73.95	74.18	74.21
30 %	82.83	71.59	64.70	62.02	62.72	62.20
40 %	74.92	61.07	53.56	51.06	50.56	51.24
50 %	65.52	49.53	41.98	40.57	40.50	41.11

Tabla 5. *Resultados de $W_{acc}(\%)$ con el codec G.723.1 (6.3 kbps).*

Tasa de pérdidas	*Long. ráfaga*					
	1	*2*	*4*	*8*	*12*	*16*
10 %	97.83	96.03	92.88	90.06	89.10	88.56
20 %	96.83	93.25	86.97	80.61	79.15	77.88
30 %	95.63	90.32	80.17	71.81	68.92	67.75
40 %	94.21	86.55	73.71	63.94	58.68	57.10
50 %	93.29	82.36	65.90	55.86	51.03	48.41

Tabla 6. *Resultados de $W_{acc}(\%)$ con el codec iLBC en el modo 20 ms (15.2 kbps).*

Tasa de pérdidas	*Long. ráfaga*					
	1	*2*	*4*	*8*	*12*	*16*
10 %	96.41	93.89	90.76	88.73	87.86	88.03
20 %	93.86	89.39	82.95	78.60	77.43	76.63
30 %	91.25	84.19	74.88	69.06	66.65	65.93
40 %	88.42	78.76	67.19	60.23	55.41	54.79
50 %	85.83	73.51	58.65	50.46	47.64	46.33

Tabla 7. *Resultados de $W_{acc}(\%)$ con el codec iLBC en el modo 30 ms (13.3 kbps).*

6.2.2. Resultados con PESQ

Tasa de pérdidas	*Long. ráfaga*					
	1	*2*	*4*	*8*	*12*	*16*
10 %	2.96	2.79	2.84	2.96	3.03	3.08
20 %	2.55	2.25	2.29	2.39	2.46	2.50
30 %	2.29	1.91	1.88	1.98	2.05	2.09
40 %	2.09	1.66	1.60	1.67	1.71	1.77
50 %	1.91	1.48	1.36	1.43	1.49	1.53

Tabla 8. *Promedios del valor medio ponderado PESQ para el codec G.729 con empaquetado de 2 tramas (2x10 ms) por paquete.*

Tasa de pérdidas	*Long. ráfaga*					
	1	*2*	*4*	*8*	*12*	*16*
10 %	2.85	2.81	2.94	3.04	3.10	3.14
20 %	2.38	2.23	2.35	2.49	2.55	2.59
30 %	2.10	1.85	1.93	2.05	2.14	2.18
40 %	1.91	1.58	1.61	1.75	1.78	1.84
50 %	1.74	1.38	1.36	1.48	1.54	1.58

Tabla 9. *Promedios del valor medio ponderado PESQ para el codec G.729 con empaquetado de 3 tramas (3x10 ms) por paquete.*

Tasa de pérdidas	*Long. ráfaga*					
	1	*2*	*4*	*8*	*12*	*16*
10 %	2.97	2.84	2.85	2.90	2.93	2.95
20 %	2.59	2.37	2.34	2.39	2.42	2.45
30 %	2.34	2.03	1.94	1.98	2.04	2.07
40 %	2.15	1.76	1.64	1.68	1.72	1.76
50 %	2.00	1.55	1.37	1.42	1.48	1.51

Tabla 10. *Promedios del valor medio ponderado PESQ para el codec G.723.1 (5.3 kbps).*

Tasa de pérdidas	*Long. ráfaga*					
	1	*2*	*4*	*8*	*12*	*16*
10 %	3.01	2.88	2.90	2.95	2.99	3.02
20 %	2.62	2.39	2.36	2.42	2.45	2.49
30 %	2.35	2.04	1.96	2.00	2.07	2.09
40 %	2.16	1.77	1.65	1.70	1.73	1.78
50 %	2.00	1.55	1.37	1.42	1.49	1.52

Tabla 11. *Promedios del valor medio ponderado PESQ para el codec G.723.1 (6.3 kbps).*

Tasa de pérdidas	*Long. ráfaga*					
	1	*2*	*4*	*8*	*12*	*16*
10 %	3.28	3.14	3.10	3.11	3.13	3.15
20 %	2.95	2.74	2.65	2.60	2.61	2.62
30 %	2.70	2.46	2.32	2.25	2.24	2.25
40 %	2.51	2.25	2.07	1.99	1.94	1.94
50 %	2.32	2.07	1.85	1.75	1.74	1.72

Tabla 12. *Promedios del valor medio ponderado PESQ para el codec iLBC modo 20 ms (15.2 kbps).*

Tasa de pérdidas	*Long. ráfaga*					
	1	*2*	*4*	*8*	*12*	*16*
10 %	3.22	3.12	3.13	3.14	3.15	3.18
20 %	2.88	2.69	2.64	2.63	2.64	2.67
30 %	2.63	2.39	2.28	2.25	2.27	2.27
40 %	2.46	2.16	2.01	1.97	1.94	1.95
50 %	2.30	1.98	1.78	1.72	1.71	1.71

Tabla 13. *Promedios del valor medio ponderado PESQ para el codec iLBC modo 30 ms (13.3 kbps).*

7. DISCUSIÓN DE RESULTADOS

Partiendo de la condición de canal limpia (sin pérdidas de paquetes), mostrada en la tabla 1, se deben de resaltar los magnificos resultados obtenidos por los codificadores CELP G.729 y G.723.1 que haciendo uso de *bit-rates* inferiores a 8 kbps obtienen buenos resultados tanto en reconocimiento como en calidad. Particularmente, hay que resaltar en esta condición ideal los resultados del codificador G.729, que haciendo uso de un *bit-rate* moderado

Cond. Canal	*0*	*1*	*2*	*3*	*4*	*5*
$P_{per}(\%)$	0	10	20	30	40	50
L_{raf}	–	1	2	4	8	16

Tabla 14. *Condiciones de canal representadas.*

(8 kbps) obtiene $W_{acc} = 98{,}64$, compaginado con una calificación PESQ de 3.95. De este modo, obtiene similares resultados a iLBC (13.33 y 15.2 kbps), codificador con *bit-rates* superiores.

Sin embargo, la situación cambia cuando se introducen las pérdidas de paquetes. Para llevar a cabo una comparativa ante esta nueva situación, es necesario agrupar los codificadores de modo que la longitud del segmento de voz correspondiente a un paquete sea igual. Teniendo en cuenta este aspecto, aparecen dos grupos comparativos. Por un lado iLBC (modo 15.2) y G.729 (tomando 2 tramas/paquete), con un empaquetado de 20 ms, mientras que por otro lado se encontrarían G.723.1, iLBC (13.33 kbps) y G.729 (3 tramas/paquete), correspondiéndose este caso con 30 ms.

Es de resaltar el resultado de reconocimiento obtenido por el algoritmo iLBC para pérdidas de un único paquete (columna 1 de las tablas 6 y 7) con $P_{per} = 50\,\%$, donde sólo cae 5.6 % del resultado en limpio en el modo 15.2 kbps (20 ms), frente al 32.6 % de G.729 (2 tramas/paq.) (véase tabla 2). Para el modo 13.3 kbps (30 ms) iLBC cae 12.9 %, frente a la caída del 41.6 % de G.729 (3 tramas/paq.) (tabla 3) y del 35.9 % y 33 % de los modos 5.3 y 6.3 kbps de G.723.1, respectivamente (tablas 4 y 5). iLBC tiene también mejor comportamiento a medida que se aumenta el tamaño de ráfaga, sin embargo, estas diferencias son menos notables.

Paralélamente, los resultados PESQ obtenidos para iLBC son mejores que para el resto de los codificadores. Sin embargo, la calidad perceptual es más sensible a la pérdida de paquetes que la tasa de reconocimiento. Mientras que, como se veía en el parrafo anterior, la degradación causada por $P_{per} = 50\,\%$ y $L_{raf} = 1$ originaba una pérdida del 5.6 % en reconocimiento para iLBC (20 ms), en la nota PESQ provoca una caída de 1.62 puntos (en una escala de 5).

Los valores representados en las figuras 4 y 5 se corresponden con las condiciones de canal expuestas en la tabla 14, siendo este muestreo de las condiciones de canal, en cierto modo, representativo de la degradación sufrida ante porcentajes de pérdidas y tamaños de ráfaga crecientes. En ambas situaciones, tal y como se apuntaba con anterioridad, el codificador iLBC consigue los mejores resultados convirtiéndose en el codificador más robusto ante pérdidas.

En general, los codificadores CELP no obtienen buenos resultados en canales degradados puesto que para la generación de la excitación hacen uso de filtros de *pitch* que propagan el error cometido más allá de los segmentos de voz perdidos. Concretamente, G.729 consigue óptimos

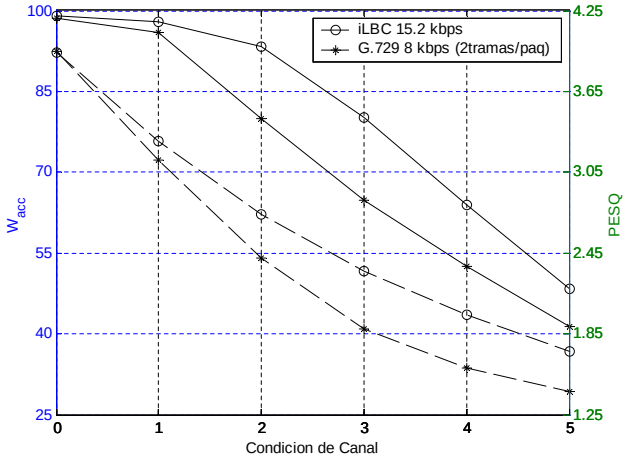


Figura 4. *Comparativa de codificadores con empaquetado de 20 ms. El trazo continuo representa el W_{acc} (escala izquierda), mientas que el discontinuo se corresponde con el promedio PESQ ponderado (escala derecha).*

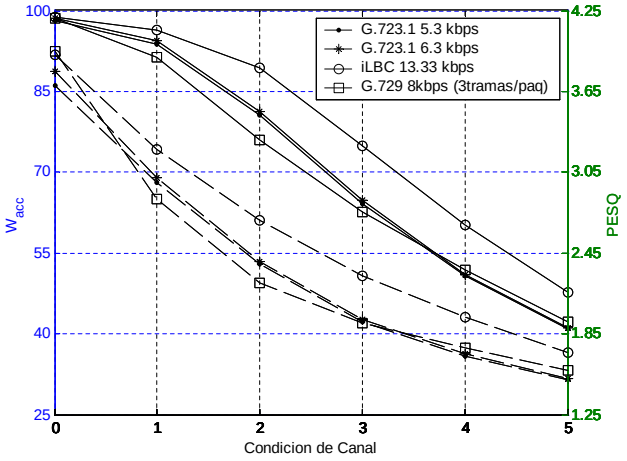


Figura 5. *Comparativa de codificadores con empaquetado de 30 ms. El trazo continuo representa el W_{acc} (escala izquierda), mientas que el discontinuo se corresponde con el promedio PESQ ponderado (escala derecha).*

resultados en limpio transmitiendo los coeficientes LSP, mediante un cuantificador diferencial predictivo. Sin embargo, esta estrategia hace al codec más vulnerable ante la pérdida de tramas consecutivas, puesto que una vez finalizada una ráfaga de pérdidas la predicción LSP del decodificador se verá notablemente degradada, obteniendo peores resultados incluso que el codec G.723.1 que mantiene *bit-rates* inferiores. El codificador iLBC resuelve esta problemática eliminando todo tipo de dependencias intertrama tanto en la generación de la excitación como en la codificación de los coeficientes LSP. El precio pagado es un considerable aumento de la tasa frente a las distintas configuraciones CELP.

8. CONCLUSIONES

En este trabajo se ha presentado una comparativa del rendimiento de los codificadores más extendidos en VoIP, atendiendo a la tasa de reconocimiento a partir de la voz sintetizada y la calidad perceptual de ésta, en un amplio abanico de condiciones de pérdidas de paquetes. Aunque iLBC y G.729 parten de resultados similares para una condición de canal limpia, los resultados obtenidos presentan a iLBC como el codificador más robusto a costa de un mayor *bit-rate*.

9. BIBLIOGRAFÍA

[1] B.W. Wah y B. Sat, "Analysis and evaluation of skype and google talk," Technical report, Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, January 2006.

[2] S.V. Andersen, W.B. Kleijn, R. Hagen, J. Linden, M.N. Murthi, y J. Skoglund, "ilbc - a linear predictive coder with robustness to packet losses," *Proceedings IEEE Speech Coding Workshop*, pp. 23–25, 2002.

[3] M. Yajnik, S. Moon, J. Kurose, y D. Towsley, "Measurement and modelling of the temporal dependence in packet loss," *Proceedings of IEEE INFOCOM*, 1999.

[4] D. Pearce y H. G. Hirsch, "The aurora experimental framework for the performance evaluations of speech recognition systems under noisy conditions," *Proc. ICSLP'00*, vol. 4, pp. 29–32, 2000.

[5] Recommendation ITU-T P.830, "Subjective performance assessment of telephone-band and wideband digital codecs," 1996.

[6] Recomendación UIT-T P.862, "Evaluación de la calidad vocal por percepción: Un método objetivo para la evaluación de la calidad vocal de extremo a extremo de redes telefónicas de banda estrecha y codecs vocales," *Serie P.800*, Febrero 2001.

[7] Recomendación UIT-T P.862.1, "Función de correspondencia para convertir los resultados brutos de la prueba p.862 en nota media de opinión de la calidad de escucha objetiva," *Serie P.800*, Noviembre 2003.

SPANISH KEYWORD SPOTTING SYSTEM BASED ON FILLER MODELS, PSEUDO N-GRAM LANGUAGE MODEL AND A CONFIDENCE MEASURE

Javier Tejedor and José Colás.
Email: javier.tejedor@uam.es, jose.colas@uam.es

Departamento de Ingeniería Informática, EPS Universidad Autónoma de Madrid. Carretera de Colmenar Viejo, km 15. 28049 Madrid, Spain.

ABSTRACT

In order to organize efficiently lots of hours of audio contents such as meetings, radio news, search for spoken keywords is essential. An approach uses filler models to account for non-keyword intervals. Another approach uses a large vocabulary continuous speech recognition system (LVCSR) which retrieves a word string and then search for the keywords in this string. This approach yields high performance but it requires a lot of training data and costly computation. In this paper we present several filler models and a confidence measure explored in a Spanish keyword spotting system. We will also investigate different weights in the grammar used for the language modelling in the keyword spotting system in order to achieve the best results. The keyword technique used is based on Hidden Markov Model (HMM). Test results are reported on a set of data from the geographic corpus of Albayzin speech data base containing 80 keywords taken from the words which most times occurs in the corpus sentences.
KEYWORDS: Speech recognition, word spotting, filler models, pseudo N-gram, confidence measure.

1. INTRODUCTION

Our task in the keyword system developed is to detect a set of keywords from a speech stream. The main challenge in word spotting techniques is to achieve the highest possible keyword detection rate while minimizing the number of false alarm keywords. That's why it is not sufficient to model only the keywords; models of the background are also required. Most of the wordspotters developed for years were variants of HHM-based, continuous speech recognition systems [1,2,3,4]. In such systems, the non-keyword intervals were represented by a variety of filler models, varying from a few phonetic or syllabic fillers to whole words [5]. Several confidence measures have been proposed by authors in order to minimise the false alarm keywords rate [6,7]. The benefits of incorporating a language model for the transitions between the keywords and the filler models were also evaluated for some of the systems [1,2,4] and were found to be substantial. Normally, the large vocabulary continuous speech recognition (LVCSR) systems with a language model component outperform any other configuration. However, the LVCSR approach to word spotting has two important disadvantages, (1) it is computationally very expensive, and (2) it tends to be domain dependent, requiring knowledge of the full vocabulary, and a large body of training data.

In this paper we describe our investigation into the use of different background or filler models in order to compare them in our Spanish keyword spotting system as well as the use of different language models similar to which are proposed in [8] to achieve the best results. We will also have included a confidence measure in order to minimise the false alarm keywords rate with no decreasing of the keyword recognition rate.

Our paper is organised as follows: Section 2 describes the experimental framework used in our system. Section 3 describes the experiments developed with Albayzin data base. Section 4 describes the discussion about the results achieved. Section 5 describes the conclusions and future work and Section 6 contains the references.

2. EXPERIMENTAL FRAMEWORK

2.1. System Description

Some systems [9] described segment-based wordspotters. Others [10] present a hybrid word / phoneme-based approach for word spotting. In our case, the Spanish keyword spotting system, based on the decoder provided by the HTK tool [11], is developed in two recognition processes. The first one is based on a phonetic decoding achieving the phoneme sequence recognized. This process is necessary in order to include the confidence measure system in our keyword spotting system. The second recognition process is the keyword spotting achieving the list of keywords recognized by the Vierbi algorithm included in the decoding process of HTK tool. The system architecture can be seen in Figure 1. The phonetic decoder process is explained in section 2.7. The keyword spotting process is explained in the section 2.8. The confidence measure system which is

composed by the phoneme performance estimator and the confidence measure module which retrieves the final output is explained in the section 2.9.

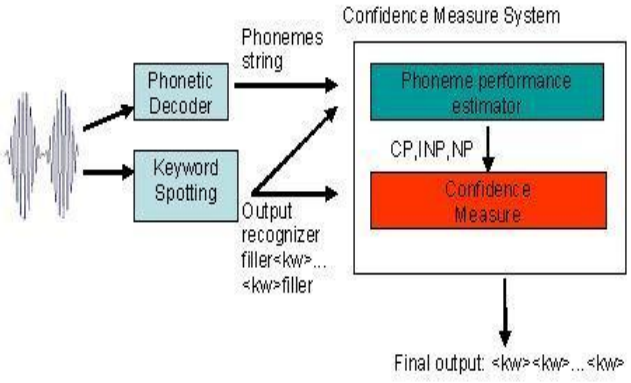


Figure 1. *Keyword Spotting System architecture*

kw denotes a keyword and filler denotes a filler model in our keyword spotting arquitecture.

The recognition network for the wordspotter is shown in Figure 2 for N keywords as well as M filler models.

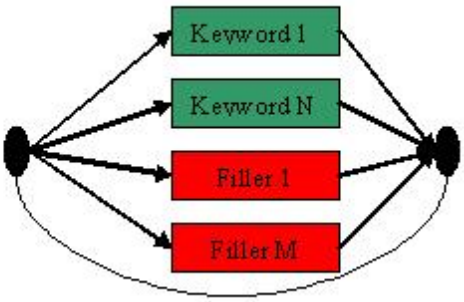


Figure 2. *Recognition network for wordspotting system*

Any transition between keywords and filler models is allowed as well as self transitions for both keywords and fillers. This configuration allows multiple keywords to exist in a single utterance, as well as multiple instances a keyword within the same utterance. For the experiments described in the next section, we used 3, 10, 25 and 49 filler models with 80 keywords.

2.2. Signal Representation and Features

For each of the utterance in the phonetic corpus for training in Albayzin data base as well as each one of the subset of the geographic corpus in the same data base used for the test of the system, 12 Mel-Frequency Cepstral Coefficients (MFCCs) plus energy and their first and second derivates were extracted to characterize the input signal. So a set of 39 MFCCs were used to represent the input signal. The initial signal has 16 Khz and 16 bits. Next, we pre-emphasise the speech signal and Hamming window is taken. The window size is 25 msec and once every 10 msec features are computed for a frame of 25 msec speech samples.

2.3. Training

We have used the phonetic corpus for training in Albayzin data base. In order to build the different filler models we have trained various filler models:

In a first step, 50 Allophones Models (AM) context-independent were trained taking into account the different phonological rules in Spanish language. These models include the initial silence, the short silence and the final silence. They were also used to run the phonetic decoder in the first process of the system.

In a second step, the theoretical set of Phonemes Models (PM) which are 26 and do not cover all the phonological rules in Spanish language were trained. They include the initial silence, the short silence and the final silence.

In a third step, we have clustered all the phonemes (Broad phonetic Models) (BM) in Spanish language according to the following classes were trained: nasals, closed vowels, opened vowels, median closed vowels, deaf fricatives, deaf oclusives, sound oclusives and liquids. The initial silence, the short silence and the final silence are also include in this configuration.

In the last step we have trained one Average Phonemes Model (APM) containing all the phonemes. The initial silence, the short silence and the final silence also belong to this configuration.

Each phoneme is modelled as CHMM (Continuous Hidden Markov Model) and each phoneme model is defined as 3-state left-to-right (no skip path) model, 15 mixtures, each of the two silence models, one for the initial silence and another for the final silence, as 3-state model, 15 mixtures and the short silence, as 1-state model, 15 mixtures (skip path). The phonetic units trained for all these cases were context-independent phones.

2.4. Keyword models

The keyword models were represented by concatenations of phonetic units. They were expanded into a pronunciation network based on their phonetic transcription in Spanish. A grapheme-to-phoneme translator was used to do it. It was also added the short silence to the end of each keyword. In our case it is not necessary to exist this short silence at the end of each keyword pronounced by the speakers due to the short silence has a skip path. The phonetic units used to represent these keyword models were the 50 Allophones Models (AM).

2.5.Filler models

We have investigated four different filler models in our system in order to compare the results achieved with each one. As filler models, we have taken the models resulted from the training of the system. We have compared the AM (Allophones Models) consists

of 49 phonemes models, PM (Phonemes Models) consists of 25 phoneme models, BM (Broad phonetic Models) consists of 10 phonemes models and the APM (Average Phonemes Models) consists of 3 phonemes models. In these filler models we have not included the short silence model which has been included at the end of the keyword as it was explained in the previous section.

2.6.Language Modelling for Keyword Spotting

As it is well-known, filler models in this kind of systems are based on the phoneme network. Keyword spotting systems tend to retrieve the sequence of phonemes instead of the keyword associated to.

To deal with this problem, a pseudo N-gram has been introduced in order to give different weights to the filler models and the keywords. We have investigated different weights combinations (1,2,4,6,8,9,10,11 and 12 represented in X axis in Figure 3 and 4) in order to achieve the best results. These different weights allowed the system to retrieve more keywords when the probability increases between these values. For example, a 'z' as value of the probability means that retrieving a keyword is 'z' times more probable than retrieving the phoneme or sequence of phonemes associated to. These different values are used as the grammar in the Viterbi algorithm in the keyword spotting process. Values higher than 12 have not been investigated because, as it can be seen in the Table 1, the system seems to achieve similar results from 11. In this way, we used a 1-gram, 2-gram, 4-gram, etc until 12-gram as language modelling.

2.7 Phonetic decoder process

First of all, a phonetic decoder based process using the Viterbi algorithm in HTK tool was developed in order to achieve the sequence of phonemes in the set of the test sentences used in the keyword spotting system. This phonetic decoder will allow us to define the confidence measure system explained in the section 2.9. The phonetic decoder will use a phonetic bigram as language model. This grammar was built taking the whole geographic corpus in Albayzin data base. The set of phonemes used in this phase was the 50 Allophones Models (AM) explained in the section 2.3.

2.8.Keyword Spotting process

The Viterbi algorithm proposed in the decoding phase in the HTK tool is used to find the best path through the labelled segment network, with the pronunciation network and the language model serving as constraints. The output is a continuous stream of fillers and keywords. The confidence measure system proposed in the next section will confirm or do not the keyword proposed by the Viterbi algorithm.

2.9. Confidence measure system

Some authors have proposed several confidence measures in order to minimise the false alarms rate in keyword spotting [6,7]. Here, we propose a new measure confidence to achieve the same benefit. We have considered only the keyword retrieved by the decoder as correct if these conditions are true:

1) CP > INP + F + abs (NP-CP-INP).

2) CP > (NP / 2).

where CP is the number of correct phonemes retrieved by the first process (phonetic decoder process) of the system in the samples where the second phase has detected the keyword,

INP is the number of incorrect phonemes of the first phase,

F is a factor which represents the difference allowed between correct and incorrect phonemes,

abs is the absolute value and

NP is th number of the phonemes of the word.

The number of correct and incorrect phonemes in the phonetic decoder process as well as the number of phonemes of each keyword are calculated in the phoneme performance estimator module of our system.

3) If the number of the phonemes of the keyword is less or equal than a constant K1, the phonemes recognized must be equal to the keyword phonetic transcription. It allowed us to deal with short words and classify them better as correct or false alarm keywords.

The steps 1), 2) and 3) are the confidence measure module of our system. It retrieves the keywords that keep these three steps and also eliminates the filler models retrieved by the keyword spotting system to achieve the final output.

A correct phoneme is defined as the phoneme recognized belongs to the keyword recognized between two positions previous the correct position and two positions next to it. An incorrect phoneme is defined as the phoneme recognized does not belong to the keyword recognized between two positions previous the correct position and two positions next to it. We will also have grouped the different phones which represent a same phoneme to not consider as errors the confusability between these phones. In this way, there are four phones for each vowel that are represented by the appropriate vowel, the two phones for the phonemes 'b' and 'd'are also grouped in phoneme 'b' and 'd', etc.

3. EXPERIMENTS

3.1 Task

All experiments were performed in the geographic Albayzin domain. The task was the detection of 80 keywords pronounced by the speakers in this data base.

The keyword consists of mountain, river and city names. They were chosen based on their high frequency of occurrence and the observation that they may constitute a sufficient set for a hypothetical spoken language system that will allow anybody to make searchs in these geographic elements. The training set was composed by 4 speakers who told 200 sentences each one and 160 speakers who told 160 sentences each one. So 4800 sentence phonetically balanced were used to train the phonetic units explained in the section 2.3. The test set was composed by sentences which belong to the geographic corpus in Albayzin data base. 48 speakers told 50 sentences to get the 2400 sentences used to test the keyword spotting system.

3.2 Performance Measures

The performance of the proposed keyword spotting system was measured using these calculations: At first, we present a graphic (Accuracy Keyword Spotting) which shows the percent of correct keywords recognized by the system respect to all the keywords to be recognized. At second we will also present another graphic (FA rate) which shows the percent of false alarms respect to the all the keywords recognized. These two graphics are shown varying the probability of the language modelling explained in the section 2.6. A keyword is considered successfully if it belongs to the sentence of the speaker which is proccessing. All the experiments were run in a Intel Pentium IV-PC 3.00 Ghz, 480 MB RAM.

3.3 Phonetic decoder results

The experiments made in the phonetic decoder process allowed us to build the confidence measure. The vocabulary in this phase consisted of the 50 Allophones Models (AM) trained according to the different phonological rules in Spanish. In this phonetic decoder, a word penalty of 0.0 was inserted and a grammar scale factor of 2.0. These values were chosen due to previous experiences in phoneme recognition with which we achieved a 78% of phonemes recognized correctly. The result of this phase is the sequence of phonemes recognized for the test sentences in the geographic corpus in Albayzin.

3.4 Keyword Spotting with AM and PM as Filler Models

The keyword spotting with the Allophones Models (AM) as filler models used to build the keywords was developed first in order to have a measure to which compare the rest of the filler models investigated. The theoretical set of Phoneme Models (PM) contains less phones due to not all the phonological rules are considered, and each phone represents each phoneme in Spanish. The vocabulary in these experiments contains the 80 keywords to be recognized and the set of 49 AM for AM and 80 keywords and the set of 25 PM for PM, filler models explained in section 2.5. The keywords and the set of AM and PM were expanded into a pronunciation keyword. A final short silence is added at the end of each keyword. The output of this keyword spotting system is a sequence of phonemes and keywords. Two factors control the decision of hypothesizing a keyword instead of hypothesizing the underlying string of phones. The first one is related to the decoder phase of the Viterbi algorithm in HTK and refers to the factors p and s called as word insertion penalty and grammar scale factor. As in the phonetic decoder these values are set to 0.0 and 2.0 respectively. The second one is related to the different probabilities associated to each arc which represents transitions between phones, transitions between keywords and transitions between keywords and phones. In this way, we have varied this probability according to the explanation in the section 2.6. The AM Keyword Spotting achieved 71.52% as its best keyword recognition rate and 18.82% as false alarms rate whereas the PM Keyword Spotting achieved 78.96% as its best keyword recognition rate and 24.79% as false alarms rate.

3.5 Keyword Spotting with General Filler Models

We have designed two set of general models consisting of 10 phonemes and 3 phonemes called BM and APM as it was explained in the section 2.5. A language model similar to the previous section was used. In the first case, the BM Keyword Spotting used the 80 keywords and a set of 10 classes of phonemes in its vocabulary. In the second case, the APM Keyword Spotting used the 80 keywords and a set of 3 classes (a generic model, the initial silence and the final silence) in its vocabulary. The decision between retrieving a class of phoneme or the generic model depending on the general filler model or a keyword is equals to the previous section. In these two cases, we have achieved the following results: The BM Keyword Spotting achieved 84.33% as its best keyword recognition rate and 41.44% as false alarms rate whereas the APM Keyword Spotting achieved 83.90% as its best keyword recognition rate and 55.72% as false alarms rate. The Accuracy Keyword Spotting rate as well as the False Alarm (FA) rate showing all the results depending on the language modelling used are shown in the Figure 3 and Figure 4 respectively.

3.6 Factors in the Confidence Measure

We have proposed in the section 2.9 a confidence measure in order to minimise the false alarms rate in our system. We have considered three issues in order to consider a keyword is correct. The first one explained in the section 2.9 requires a factor introduced which allows us to be more restrictive when we have to accept a keyword as correct. This factor represents the difference

allowed between the number of correct phonemes and the number of incorrect ones retrieved by the phonetic decoder associated to the keyword retrieved by the wordspotter to consider it as correct. Several experiments made before this process allowed us to set this factor to 1 in order to get a good rate between the correct keywords loosed and the false alarm keywords detected due to this factor. The another factor is the constant K1. In our experiments developed, a value 4 for it gave us the best results in the confidence measure system.

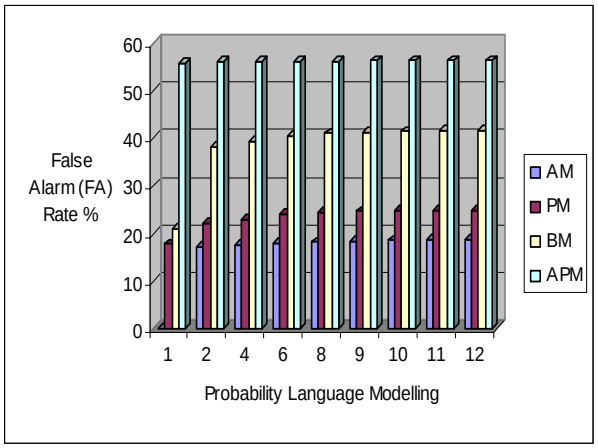


Figure 3. *Accuracy Keyword Spotting Rate*

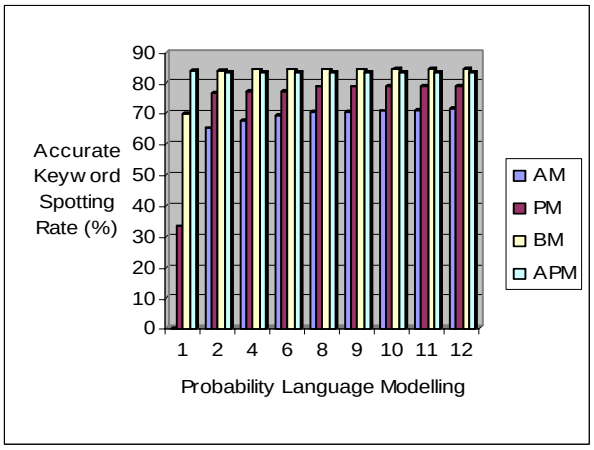


Figure 4. *False Alarms (FA) Rate*

4. DISCUSSION

There is a clearly correlation between the accuracy keyword spotting rate, the percent of correct keywords recognized of all the keywords in the test evaluation, and the false alarms rate, the percent of incorrect keywords recognized of all the keywords recognized in the test evaluation. The more keywords are recognized correctly, the more false alarms the system retrieves. As we can see in Figure 3 and Table 1, the BM filler model retrieves the best accuracy keyword spotting rate, with the 84% of keywords recognized. This rate is achieved when the Probability Language Modelling (PLM) varies from the range of 2 to 12. PLM in Table 1 represents the Probability Language Modelling explained in the section 2.6. As we can see in the Figure 4 and Table 1, the less false alarms rate we achieve is 17.32 %, achieved with the AM model. In this model the accuracy keyword spotting in this case decreases to 65.18%. Depending on the kind of the global system, for example different audio web searchs, we can choose the best filler model for each situation. In case of the false alarm keywords rate does not important, we would choose the BM filler model due to it retrieves the best accuracy keyword spotting rate. In case of the false alarm keywords rate is important in the system, we would choose the AM model due to it minimise the false alarms rate, despite that retrieving the less accuracy rate. We can see two relevant things in this table. The first one is that when the probability given to the language model is the same for the filler model and for the keyword, the system does not retrieve any keyword for the AM filler model. It is due to the Viterbi algorithm prefers the sequence of phonemes instead of the keyword represented by the concatenation of these phonemes. That's why both the accuracy and the false alarms rate is 0%. The second one is that when we use a Average Phoneme Model (APM) as filler model, the false alarms rate increases a lot. That's why a unique model that represents all the phonemes in the sentences produces that the system prefers a keyword due to the great distance measured by the Viterbi algorithm between the samples in the sentences and the unique model. We can also see that when the language modelling probability to retrieve a keyword increases compared with the filler model one, both the accuracy keyword spotting rate and the false alarms rate also increases. It is due to the system prefers to retrieve a keyword instead of the sequence of phonemes because the probability assigned to it is greater.

PLM	AM Filler Model (Accur / FA)	PM Filler Model (Accur / FA)	BM Filler Model (Accur / FA)	APM Filler Model (Accur / FA)
1	0% / 0%	33,23% / 17,79%	70,18% / 21,05%	83,9% / 55,72%
2	65,18% / 17,32%	76,46% / 22,2%	84,09% / 38,1%	83,23% / 56,15%
4	67,87% / 17,55%	77,43% / 23,07%	84,21% / 39,45%	83,29% / 56,15%
6	69,15% / 17,94%	77,44% / 23,97%	84,21% / 40,37%	83,35% / 56,17%
8	70,37% / 18,21%	78,6% / 24,57%	84,27% / 41,09%	83,29% / 56,23%
9	70,55% / 18,34%	78,6% / 24,66%	84,27% / 41,24%	83,29% / 56,32%
10	70,79% / 18,52%	78,72% / 24,77%	84,33% / 41,44%	83,29% / 56,32%
11	71,22% / 18,72%	78,9% / 24,72%	84,33% / 41,57%	83,23% / 56,34%
12	71,52% / 18,82%	78,96% / 24,79%	84,33% / 41,67%	83,23% / 56,33%

Table 1. *Summary of Accuracy Keyword Spotting and False Alarms Rate for the Filler Models according to the Probability Language Modelling (PLM)*

5. CONCLUSIONS AND FUTURE WORK

There is a correlation between the accurate keyword spotting rate and the false alarms rate in our system due to an increase in the accurate keyword spotting produces an increase in the false alarms rate. A compromise between the accuracy keyword spotting and the false alarms rate depends on the global system where the keyword spotting process is integrated, but an acceptable compromise between the accuracy keyword spotting and the false alarms rate seems to be the PM Filler Model, achieving a 78,96% of accuracy and a 24,79% of false alarms rate.

As feature work we are going to try to reduce the false alarm keywords rate for the filler models investigating other confidence measures.

6.REFERENCES

[1] R. Rose, “Definition of subword acoustic units for wordspotting”, Proc. EUROSPEECH’93, pp, 1049-1052, September 1993.

[2] P. Jeanrenaude, K. Ng, M. Siu, J.R. Rohlicek, and H. Gish, “Phonetic-based word spotter: various configurations and application to event spotting”, Proc. EUROSPEECH’93, pp 1057-1060, September 1993.

[3] E. Lleida, J.B. Marino, J. Salavedra, A. Bonafonte, E. Monte, and A. Martinez, “Out of Vocabulary word modelling and rejection for keyword spotting”, Proc. EUROSPEECH’93, vol. 2, pp. 1265-1268, September 1993.

[4] M. Weintraub, “Keyword-spotting using SRI’s DECIPHER large-vocabulary speech recognition system”, Proc. ICASSP’93, vol. 2, no. 2, pp. 463-466, April, 1993.

[5] H. Cuayáhuitl and B.Serridge, “Out-of-vocabulary Word Modeling and Rejection for Spanish Keyword Spotting Systems”, Proc. MICAI’02, vol. 2313, pp.156, 2002.

[6] T. Schaaf, T. Kemp, “Confidence Measures for Spontaneous Speech Recognition”, Proc. ICASSP’97, vol 2, pp. 887-890, April, 1997.

[7] S. Cox and R. Rose, “Confidence Measures for the Switchboard database”, Proc. ICASSP’96, vol. 1, no. 1, pp. 511-515, May, 1996.

[8] Joo-G. Kim, Ho-Y. Jung and Hyun-Y. Chung, “A Keyword Spotting Approach based on Pseudo N-gram Language Model”, Proc. SPECOM’04, pp.156-159, September 2004.

[9] A. S. Manos and V. W. Zue, “A segment-based wordspotter using phonetic filler models”, Proc. ICASSP’97, vol. 2, no. 2, pp. 899-902, April 1997.

[10] P. Yu, F. Seide, “A hybrid word / phoneme-based approach for improved vocabulary-independent search in spontaneous speech”, Proc. ICSLP’04, vol. 13, no. 5, pp. 635-643, October 2004.

[11] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D, Povey, V. Valtchev and P. Woodland. *HTK Book v.3.2.1*, Microsoft Corporation and Cambridge University Engineering Department, December 2002.

SPEAKER RECOGNITION EXPERIMENTS ON A BILINGUAL DATABASE

Marcos Faundez-Zanuy, Antonio Satué-Villar

Escola Universitària Politècnica de Mataró, Universitat Politècnica de Catalunya (UPC)
Avda. Puig i Cadafalch 101-111, E-08303 Mataró (BARCELONA), SPAIN
e-mail: {faundez, satue}@eupmt.es http://www.eupmt.es/veu

ABSTRACT

This paper presents some speaker recognition experiments using a bilingual speakers set (49), in two different languages: Spanish and Catalan. Phonetically there are significant differences between both languages. These differences have let us to establish several conclusions on the relevance of language in speaker recognition, using two methods: vector quantization and covariance matrices.

1. INTRODUCTION

This paper deals with speaker recognition [1] (identification and verification) with fully bilingual speakers. Thus, we extend our previous results published in 1999 [2].

We have done a set of experiments with a bilingual database in order to establish if the language of the speaker has relevance in a speaker identification and verification application (mainly if it is more suitable one language than other, and if it is possible to recognize with different training and testing languages).

Phonetically there are significant differences between both languages. Mainly, the Catalan language has eight vowels (see figure 1) and Spanish only five. Although there are only nine million people of Catalan speakers in front of four hundred million people of Spanish, both languages can be used for our purpose. The differences between both languages have let us to establish several conclusions on the relevance of language in speaker recognition.

Another important question is that for bilingual speakers in conversational speech is quite common the change from one language to the other, so it is interesting to evaluate if this fact can affect a speaker recognizer.

For these experiments we have used our previous database [3]. An interesting fact is that the Spanish sentences have been balanced, but the Catalan ones have been merely translated from Spanish. Thus, the database consists of the same texts recorded in both languages in the same day, one language after the other. Speaker could freely choose which the first recording language was.

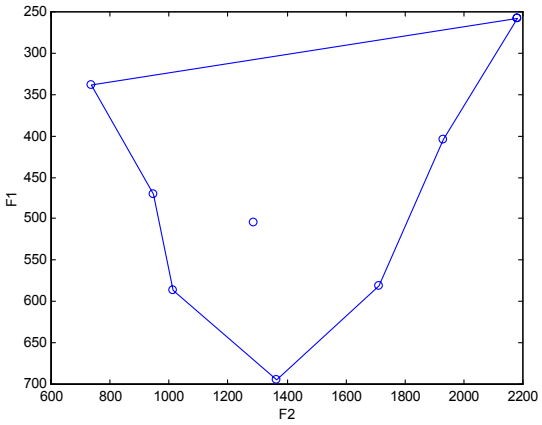


Figure 1: Formants of the Catalan vowels

2. DATABASE

The main characteristics of the database are:

- 4 sessions with different tasks in each session (isolated numbers, connected numbers, sentences, text, conversational speech, etc.)
- In each session, tasks were sequentially collected in both languages (Catalan and Spanish), uttered from the same speaker. Each task was simultaneously acquired with two microphones (SONY ECM-66B and AKG C-420).

This paper presents results of the fourth session using the common text (aprox. 1 min) and the first five sentences (approx, 4 seconds lasting each one).

3. SPEAKER RECOGNITION EXPERIMENTS

With this database we have made several tests:

- Speaker recognition with each language: train and test in Catalan (CC), train and test in Spanish (SS)
- Speaker recognition with different train and test conditions: train in Catalan and test in Spanish (CS), train in Spanish and test in Catalan (SC).

Two speaker recognition methods have been used:
1. Vector quantization [4] with LBG [5] algorithm for codebook generation (1 codebook for each speaker). The number of parameters used in each model is:

$$parameters = 2^{No} \times P \quad (1)$$

where P is the analysis order of the parameterization (dimension of LPCC vectors) and *No* is the number of bits of the codebook ranging from 0 to 8.
2. Arithmetic-harmonic sphericity measure [6], which implies the computation of a covariance matrix for each speaker, and the following measure distance:

$$\mu\left(C_j C_{test}\right) = \log\left[tr\left(C_{test}C_j^{-1}\right)tr\left(C_j C_{test}^{-1}\right)\right] - 2\log(P) \quad (2)$$

where C_j and C_{test} are covariance matrices and P is its size.

The trace of the matrices can be computed as:

$$tr\left(YX^{-1}\right) = 2\sum_{i=1}^{P}\sum_{j=1}^{i-1} y_{ij}\tilde{x}_{ij} + \sum_{k=1}^{P} y_{kk}\tilde{x}_{kk} \quad (3)$$

where, x_{ij}, $\tilde{x}_{ij}$, y_{ij}, $\tilde{y}_{ij}$ are respectively the elements of the matrices X, X^{-1},Y and Y^{-1}.

The number of parameters for each speaker is (the covariance matrix is symmetric):

$$parameters = \frac{P^2 + P}{2} \quad (4)$$

Although these methods are not the state-of-the-art in speaker recognition, they require lower computational time than GMM. On the other hand, we are interested on relative comparisons, the recognition algorithm not being a critical issue.

4. RESULTS

The results have been obtained with the following parameters:

- 49 bilingual speakers.
- 1 read text (about 1 minute and the same text for all speakers) for computing the models.
- 5 different test sentences (the same for all speakers).
- *No*=number of bits of the codebook, from 0 to 8.
- Silence removal
- Frames of 240 samples with an overlap of 2/3.
- Hamming window and pre-emphasis of 0.95.

We have evaluated the identification and verification results. Verification systems can be evaluated using the False Acceptance Rate (FAR, those situations where an impostor is accepted) and the False Rejection Rate (FRR, those situations where a user is incorrectly rejected), also known in detection theory as False Alarm and Miss, respectively. This framework gives us the possibility of distinguishing between the discriminability of the system and the decision bias. The discriminability is inherent to the classification system used and the discrimination bias is related to the preferences/necessities of the user in relation to the relative importance of each of the two possible mistakes (misses vs. false alarms) that can be done in verification. This trade-off between both errors has to be usually established by adjusting a decision threshold. The performance can be plotted in a ROC (Receiver Operator Characteristic) or in a DET (Detection error trade-off) plot [7]. DET curve gives uniform treatment to both types of error, and uses a scale for both axes, which spreads out the plot and better distinguishes different well performing systems and usually produces plots that are close to linear. DET plot uses a logarithmic scale that expands the extreme parts of the curve, which are the parts that give the most information about the system performance. For this reason the speech community prefers DET instead of ROC plots.

We have used the minimum value of the Detection Cost Function (DCF) for comparison purposes. This parameter is defined as [7]:

$$DCF = C_{miss} \times P_{miss} \times P_{true} + C_{fa} \times P_{fa} \times P_{false} \quad (5)$$

where C_{miss} is the cost of a miss (rejection), C_{fa} is the cost of a false alarm (acceptance), P_{true} is the a priori probability of the target, and $P_{false} = 1 - P_{true}$. $C_{miss} = C_{fa} = 1$.

4.1 Vector quantization results

Table 1 summarizes the results for a vector quantization speaker identification method, with parameterizations LPCC-12, 16 and 20, and for codebooks ranging from 0 to 8 bits.

Although vector quantization (VQ) performs well for identification task, reaching identification rates up to 100%, the verification task is not so successful, and it is outperformed by the covariance matrices (CM) method.

4.2 Covariance matrices

The parameter that can be adjusted for modeling the speakers is the prediction order (P). That is, the dimension of the LPCC vectors.

We have studied several P values (table 2). It is important to see that a frame length of 240 samples is used, so for a correct LPC parameter estimation, the prediction order must not be higher than 24, because then the autocorrelation used in the Levinson-Durbin recursion can not be properly estimated. For this reason, the recognition rates drop for high P values.

Another important fact is that a covariance matrix assumes that the modeled distribution is symmetrical. This assumption is not made in the VQ approach. Thus, for nonsymmetrical distributions the VQ approach could be more accurate.

No	Num. Parameters (aprox.)	P
0	12	4
1	24	6
2	48	9
3	96	13
4	192	19
5	384	27
6	768	39
7	1536	55

Table 3: # of parameters used in VQ (P=12) and CM

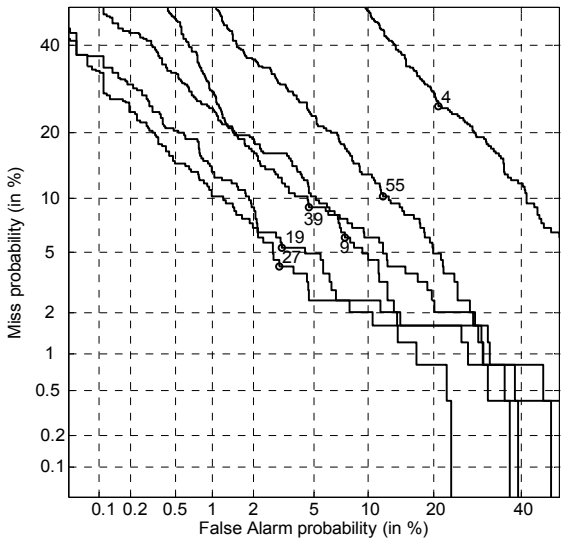


Figure 2. DET plots for Covariance matrices of sizes 4, 9, 19, 27, 39 and 55.

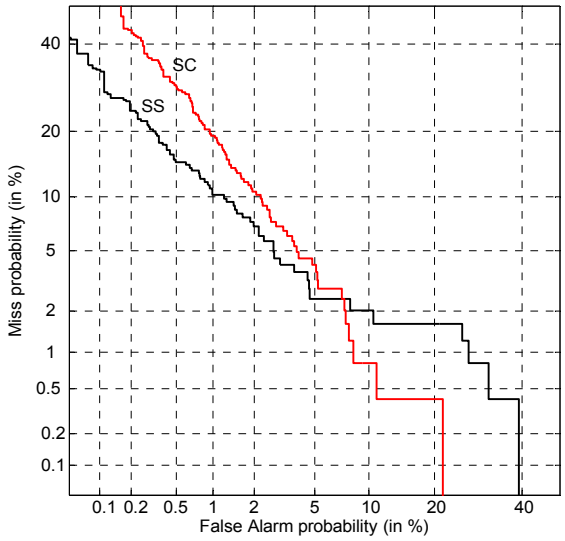


Figure 4. DET plots for Covariance matrices of size 27 and two training/testing scenarios (SC and SS).

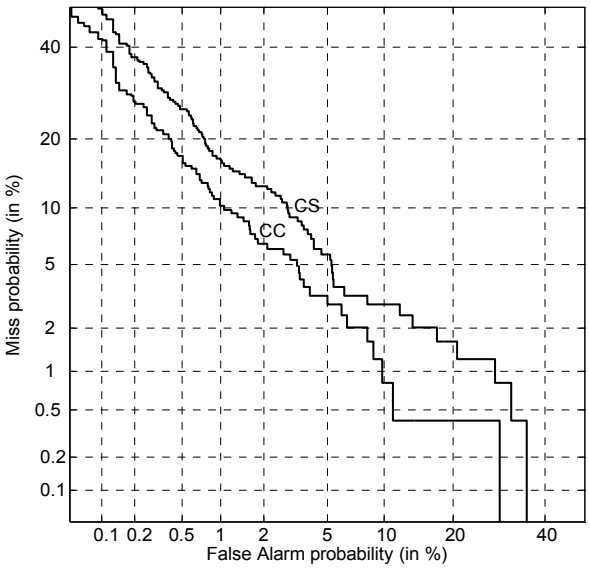


Figure 3. DET plots for Covariance matrices of size 27 and two training/testing scenarios (CC, CS).

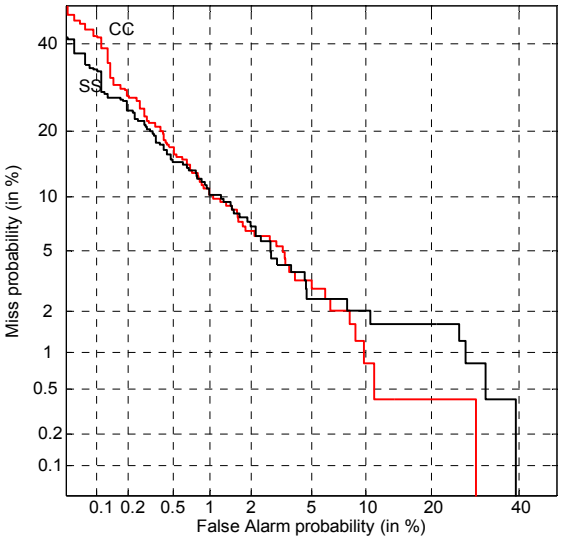


Figure 5. DET plots for Covariance matrices of size 27 and two training/testing scenarios (SS and CC).

Figure 2 compares DET plots for CM-4, 9, 19, 27, 39 and 55 using training and testing in Spanish. Figures 3, 4 and 5 compare several training and testing scenarios (CC, CS, SC and SS) when using CM-27.

Figures 3 and 4 show a slight degradation when training and testing languages are different. In fact, the minimum Detection Cost Function (DCF) increases from 3.6% to 4.5% and 4% respectively in figures 3 and 4.

Figure 5 shows that both plots intersect. Thus, we cannot affirm that one language produces always better results than the other one.

For comparing both methods (vector quantization and covariance matrices), we have used parameters No and P that require the same storage memory, as we see in table 3

5. CONCLUSIONS

In this paper we have studied speaker identification and verification tasks using a bilingual speaker data set. The main conclusions are:

- The Catalan database yields higher identification rates than the Spanish one for a high number of parameters. Otherwise the Spanish language achieves better rates. We think that this is due to the higher number of vocalic phonemes (8 in Catalan against 5 in Spanish).
- With different test and train conditions there is a little decrease in identification rate (about 1% for

high resolution codebooks, and greater values for other models and methods)

- For VQ better results are obtained when increasing the codebook size. Thus, best results are obtained for the larger size: 8 bits. On the other hand, for CM, the model size is related with the parameterization order (*P* value) and the optimal is obtained around P=27 for both tasks, identification and verification.
- Although VQ achieves the highest identification rates, the CM method is faster and in most cases requires less parameters for modeling each speaker. Additionally, CM provides better verification results, evaluated with the minimum DCF value.

6. ACKNOWLEDGEMENTS

This work has been supported by FEDER and the Spanish grant MEC TIC2003-08382-C05-02

7. REFERENCES

[1] M. Faundez-Zanuy "State-of-the-art in speaker recognition". IEEE Aerospace and Electronic Systems Magazine. Vol.20 nº 5, pp 7-12, May 2005

[2] A. Satue-Villar, M. Faundez-Zanuy "On the relevance of language in speaker recognition" Proc. EUROSPEECH'99 Budapest, Vol. 3 pp.1231-1234, September 1999.

[3]J. Ortega-García, J. González-Rodríguez and V. Marrero-Aguiar"AHUMADA: A Large Speech Corpus in Spanish for Speaker Characterization and Identification". Speech communication Vol. 31 (2000), pp. 255-264, June 2000

[4] F. K. Soong et alt. "A vector quantization approach to speaker recognition". pp.387-390. ICASSP 85

[5] A. Gersho and R.M. Gray. Vector Quantization and Signal Compression, Kluwer Academic Publishers, 1991.

[6] F. Bimbot, L. Mathan "Text-free speaker recognition using an arithmetic-harmonic sphericity measure. pp.169-172. EUROSPEECH 91

[7] A. Martin et alt. "The DET curve in assessment of detection performance", V. 4, pp.1895-1898, European speech Processing Conference Eurospeech 1997

		No=0		No=1		No=2		No=3		No=4		No=5		No=6		No=7		No=8	
P	Train/test	Iden	ver	Iden	ver	Iden	ver	Iden	ver	Iden	ver	Iden	ver	Iden	ver	Iden	ver	Iden	ver
12	CC	58.8	36.6	69.8	27.9	79.6	21.1	93.1	13.8	98.4	11.7	98	10.3	98.8	9.1	99.6	8.0	100	7.1
12	CS	49.8	33.7	65.7	25.4	68.6	18.9	87.8	13.4	93.9	12.0	96.3	10.5	97.6	10.0	98.0	9.5	98.8	9.0
12	SS	64.1	33.0	74.7	24.8	83.7	18.1	93.5	12.4	97.1	11.1	98.8	9.7	99.2	8.2	98.8	7.6	99.2	7.3
12	SC	47.3	36.9	59.2	28.4	73.1	21.0	90.2	12.2	95.1	12.8	95.9	12.0	98.4	10.7	98.8	10.1	98.4	10
16	CC	62.4	35.6	72.7	27.0	83.3	20.1	93.5	13.4	99.6	11.9	99.2	11.3	99.6	9.7	100	8.4	100	7.5
16	CS	55.9	32.6	68.6	24.3	73.9	18.9	88.6	13.6	95.5	12.4	96.7	11.7	97.6	10.1	99.2	9.9	99.2	9.7
16	SS	68.2	31.7	77.6	23.0	84.5	17.4	94.3	12.5	96.7	11.1	98.4	10.3	98.8	8.4	98.8	8.0	99.2	7.4
16	SC	53.5	35.3	64.9	26.9	77.6	20.6	92.7	14.7	96.7	13.7	97.1	12.3	98.4	11.7	98.4	10.5	98.8	10.2
20	CC	64.9	34.6	74.3	26.4	86.1	20.9	93.1	15.1	99.6	12.7	100	11.0	100	10.6	100	9.0	100	8.3
20	CS	57.1	31.9	70.6	24.3	77.1	19.7	88.6	14	96.3	12.9	97.1	11.3	98.8	10.5	99.6	10.0	99.6	9.8
20	SS	69.0	30.4	80.0	23.1	85.3	17.8	94.3	13.2	97.6	11.1	98.4	9.9	98.8	9.1	99.2	8.1	99.6	7.3
20	SC	54.3	34.8	64.5	26.6	80.4	20.9	93.9	16.4	97.6	14.6	98.0	12.6	98.0	11.3	98.8	11.0	99.2	10.2

Table 1: Identification rates and DCF (verification) using VQ (S=Spanish C=Catalan),

	P=4		P=6		P=9		P=13		P=19		P=27		P=39		P=55	
Train/test	Iden	ver	Iden	ver	Iden	ver	Iden	ver	Iden	ver	Iden	ver	Iden	ver	Iden	ver
CC	22.0	25.8	55.9	13.9	82	7.4	91.0	6.4	96.7	5.2	99.2	3.6	99.2	6.6	92.7	9.7
CS	22.4	28.1	52.7	15.4	77.6	7.6	84.9	6.5	91.8	5.4	95.9	4.5	92.7	8.0	86.1	12.5
SS	27.3	23.1	65.3	12.3	87.3	6.8	92.7	5.1	97.1	4.2	98.8	3.6	95.9	6.8	90.6	11.0
SC	22.0	28.4	50.6	16.9	78.4	7.6	88.2	7.4	95.5	6.0	98.4	4.0	95.9	7.5	88.6	11.4

Table 2: Identification rates and DCF (verification) using CM (S=Spanish C=Catalan)

Time course of Intonation Processing in European Portuguese: a Gating paradigm approach

Isabel Falé[1,2], *Isabel Hub Faria*[1]

[1]Onset-CEL, Lab. Psicolinguística, Universidade de Lisboa; [2]Universidade Aberta

ABSTRACT

EP listeners seem to pay attention to phonetic detail specificities of the speech signal, which allow them to start building their internal representation of the intonation contour very early in the sentence.
Using the Gating Paradigm methodology (Grosjean 1980, 1996), 20 European Portuguese (EP) native speakers were presented with auditory speech stimuli gated in specific sentence locations, which they had to classify within a category and after it to rate their own answer confidence level.
The influence and weight of lexical cues, especially *wh*-words, for early sentence type recognition was considered.
Results showed that the lexical cues power for sentence type processing is stronger than the prosodic ones. Sentences starting with *wh*-words, whatever semantic or syntactic function or prosodic information they have tend to be classified as questions. The interplay of all cues seems to occur later in the sentence, after more information becomes available.

1. INTRODUCTION

Prediction is a very important issue in speech processing and it has effects on verbal communication accuracy: it prevents silent gaps on speech interaction and speeds speech processing, diminishing the interpretation hypotheses and leaving the listener free to perform other on-going cognitive processes. For prediction to occur, linguistic information must be available early in the speech signal so that processing may start immediately.

Until recently in European Portuguese (EP), most relevant intonation movements were believed to occur locally, at the end of the sentences [1, 2, 3]. This would imply that EP listeners would have to wait till the end of an utterance in order to interpret it as a sentence type. We would also expect to find frequent silent gaps within verbal interaction, corresponding to the sentence intonation processing which would only begin after sentence end. In this study, as well as in some previous ones we will show that this not the case [4].

Recent studies have been reporting the importance of phonetic detail for speech perception and processing. These studies argue that listeners process phonetic detail [5] available in the speech signal to start building the internal representation of the linguistic entities. This assumption is in accordance with the prediction phenomenon.

Our main hypothesis is that EP speech signal has early prosodic information that provides listeners with enough data to access and recognize sentence type before its end.

Another hypothesis is related to the possible lexical cues interference in the course of speech processing. In a previous EP perception study in this language [4], lexical cues didn't show to be as important as researchers think they are. In fact, subjects didn't achieve better identification and classification scores when, for instance, sentences started with what was thought to be strong lexical cues to specific interpretation. So, in order to get a better understanding of the way lexical cues interact with prosodic ones, we have studied the interplay between these cues, assuming that if lexical cues restrict speech processing then listeners will only need to have access to lexical data to be able to interpret and to classify sentences within categories.

2. METHOD

For hypothesis testing, we have chosen the linguistic intonation contrast between declaratives and interrogative sentences in EP reported by [1, 2, 3] using the *Gating Paradigm* methodology [6, 7], which was initially developed for word recognition.

In this experimental paradigm, target words are cut into smaller pieces of increasing duration. For instance, words can be gated with a fix interval of 30 msec: the first piece will have 30 msec, the second 60 msec, the third 90 msec and so on till the end of the stimulus. These gated stimuli are presented to subjects in a sequential (increasing duration) order and they have to guess which the word is. Just after that, subjects have to rate their confidence level on their own answer to the identification or guessing task.

Previous works within this paradigm have selected the temporal criterion for stimulus gating. However, most of these studies work at lexical word level. This

criterion was transposed to bigger units of grammatical analysis [7].

Based on the assumption that stressed vowels in EP are strong informative prosodic units in an intonation contour [1], we developed a phonetic criterion to gate our sentences instead of the temporal one. Sentences were gated at the end of a stressed vowel within a word in the sentence.

The identification point of a sentence is a function of two independent measures: *isolation point (IP)* and *recognition point (RP)*. *Isolation point* refers to the location on the stimulus where correct identification is achieved and maintained over fragments of the same sentence till its final fragment. Correct identification, however, does not inform us on how confident subjects are on their answers. The rating of the answers confidence level by subjects will determine the *recognition point*. *Recognition point* is then reached when a stimulus is first rated as 'sure' and this rating is not changed till the end of the sentence.

2.1. Stimuli

For early prosodic identification hypothesis research, five pairs of stimuli (10 sentences) with variable duration were gated in diverse locations. In each pair of stimuli (A, B, C, D, E) there were two segmental identical sentences, both produced by the same speaker that differed in their intonation contour: one had question contour and the other a statement one. Both sentences in the pair were cut in the same locations. (See Table 1, for an example of a sentence gating).

Sti	Gate	Sentence Fragment
1	FSV	Os golfi
2	SV1	Os golfinhos cinzen
3	SV2	Os golfinhos cinzentos anima
4	SV3	Os golfinhos cinzentos animaram o jogador
5	SV4	Os golfinhos cinzentos animaram o jogador e o adi
6	LST=SF	Os golfinhos cinzentos animaram o jogador e o adido naval

Table 1. *Example of sentence gating*

For lexical cues hypothesis testing, sentences and sentence fragments beginning with *wh*-words (9 sentences in three sentence groups), either inducing question or statement interpretation, were selected. Part of these sentences introduced true *wh*-questions and another part was constituted by fragments initiated by *wh*-words that, although phonetically equal to true *wh*-words, have different semantic and syntactic functions and diverse prosodic features that we expected to be enough to block immediate question interpretations (See Table 2, for an example of the sentences used to test this hypothesis).

Sentences – stimuli	Types
[O Pedro viu] quem (*who)* animou os golfinhos.	I
Quem *(Who)* animou os golfinhos [fez um bom trabalho].	II
Quem *(Who)* animou os golfinhos?	III

Table 2. *Example of stimuli with wh-words*

Type I stimuli refer to sentence fragments beginning with *wh*-words that were extracted from medial position in the original statement sentence. Type II stimuli were extracted from statement sentences beginning with a *wh*-word. Type III stimuli were true questions starting with *wh*-words.

2.2. Tasks and experimental procedures

Sentence fragments with increasing duration determined by stressed vowels location were presented sequentially to subjects that had to perform two tasks. First, they had to identify the fragment with a sentence type in a two-choice forced task, where possible answers were 'statement' and 'question'. Secondly, they had to rate their confidence level for the given answer on fragment identification. A simple two-scale was available for this purpose: answers could be rated either as 'sure' or as 'unsure'.

Subjects were instructed to listen to each stimulus and to proceed to its classification (statement or question) by pressing a computer key. Just after it, they had to rate the confidence level of their own answer.

The experimental procedure was developed in *E-Prime* [9]. Stimuli were auditory presented through the computer, via headphones. Responses were registered through the computer keyboard. All technical tests were previously performed to guarantee adequate data collection quality by the computer.

Subjects

Twenty European Portuguese native speakers (10 female), aged between 27 and 44, with no history of hearing or language deficits or disorders, participated in the experiments. All, except one, had a graduate degree. Experiment was run individually in one session.

3. RESULTS

3.1. Early prosodic identification hypothesis

3.1.1. Isolation Point

Graphs in figures 1-5 present *IP* results' for the 5 pairs of stimuli for all subjects. In graphs, black bars are for statement sentences and white bars to questions. Gates legend is the following: FSV - first stressed vowel; SVx - stressed vowel x; LSV - last stressed vowel; and SF - sentence final.

In pairs A, B, D and E, the last stressed vowel occurs in the final syllable of the sentence, so the last stimulus gate for these sentences was also its ending. In pair C, the last stressed vowel occurs in the penultimate syllable of the sentence, so there was an extra stimulus gate to reach the end: SF.

In statements, *IP* is achieved in the very first stressed vowel (FSV) for all pairs (see figures 1-5). By contrast, in questions, *IP* tends to occur later in the sentence, near its end. The exception goes to the question in pair C which is 'isolated' sooner (figure 3).

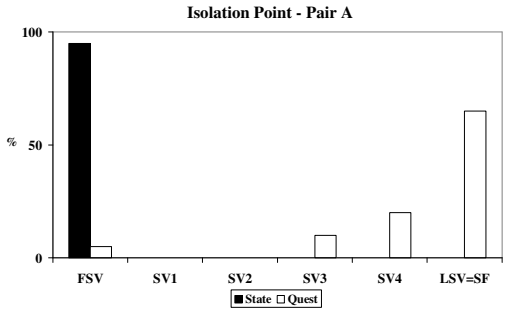


Figure 1. *Percent results for IP in pair A (all subjects included)*

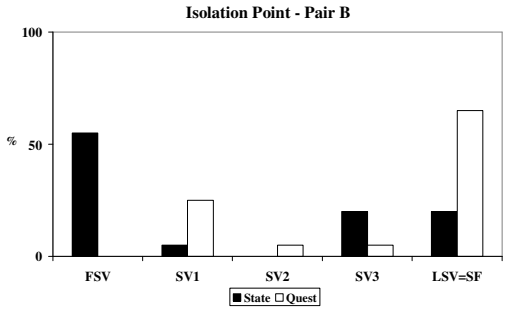


Figure 2. *Percent results for IP in pair B (all subjects included)*

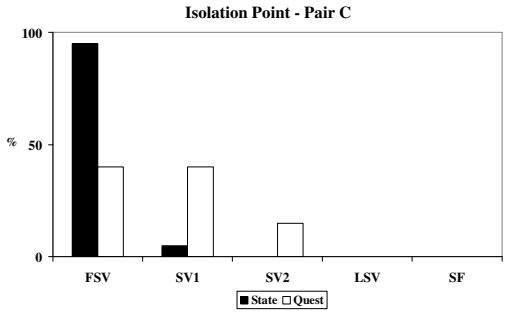


Figure 3. *Percent results for IP in pair C (all subjects included)*

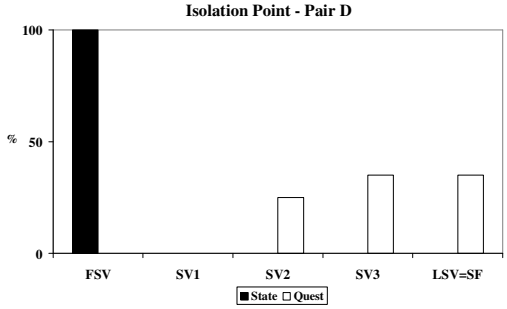


Figure 4. *Percent results for IP in pair D (all subjects included)*

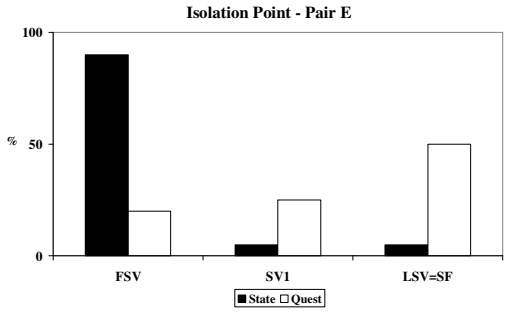


Figure 5. *Percent results for IP in pair E (all subjects included)*

3.1.2. Recognition Point

These results are more disperse than the *IP* ones (figure 6). In general, *RP* occurs later than *IP* (Table 3) which is explained by the different processing nature of these tasks. The processing time for decision making in the *IP* task is shorter than for the *RP* task because *RP* requires an explicit memory recalling that takes longer.

There is, however, an exception in the pair B question, where *RP* appears sooner than *IP*. Subjects tend to wait for more intonation data availability in order to decide whether they are sure or not about their own identification answer.

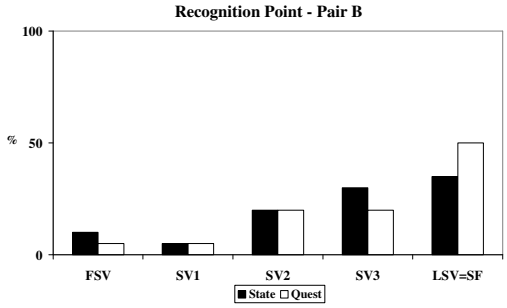


Figure 6. *Percent results for RP in pair B (all subjects included)*

There is no systematic relation between the distribution of *IP* and *RP* results. However, there is a tendency for statements to be recognized earlier (Table

3). At first we thought that this tendency could be a *default* answer effect but we observed that this was not what happened in other circumstances, for instance, when the sentences begin with *wh*-words. So, we didn't consider it to be a *default* answer.

Locations for the *IP* and *RP* on Table 3 were the result of the sum of percent data: whenever the percent data sum reached at least 50% (above chance level), we have considered that the *IP* or the *RP* of the sentence had been achieved and registered the gate in the sentence location where that happened.

	Statement		Question	
	IP	***RP***	***IP***	***RP***
A	FSV	SV1	LSV=SF	LSV=SF
B	FSV	SV3	LSV=SF	SV3
C	FSV	SV2	SV1	SV2
D	FSV	SV2	SV3	LSV=SF
E	FSV	SV1	LSV=SF	LSV=SF

Table 3. *Location results for IP and RP. Shadowed cells refer to sentence end (all subjects included)*

3.2. Lexical cues hypothesis

3.2.1. Isolation Point

IP results' for the 3 groups of sentences are available in graphs of figures 7-9. In graphs black bars represent Type I stimuli, white bars Type II and gray bars Type III.

In Type I and Type III stimuli from Sentence Group 1 (figure 7), the *IP* is located in the First Stressed Vowel (FSV).

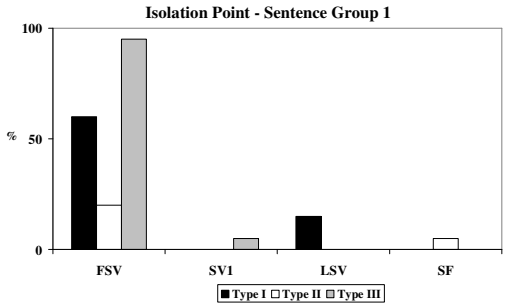


Figure 7. *Percent results for answer 'question' IP in Sentence Group 1 (all subjects included)*

In Type I, Type II and Type III stimuli from Sentence Group 2 (figure 8), the *IP* is also located in the FSV.

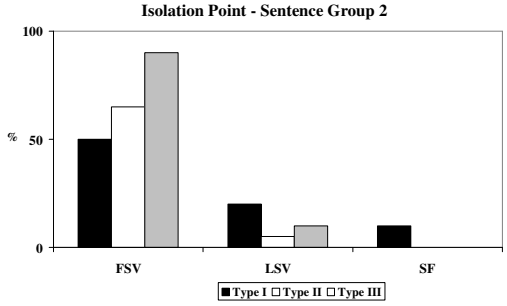


Figure 8. *Percent results for answer 'question' IP in Sentence Group 2 (all subjects included)*

The location for *IP* in Sentence Group 3 (figure 9) is only available for Type III stimuli and it relies on the FSV.

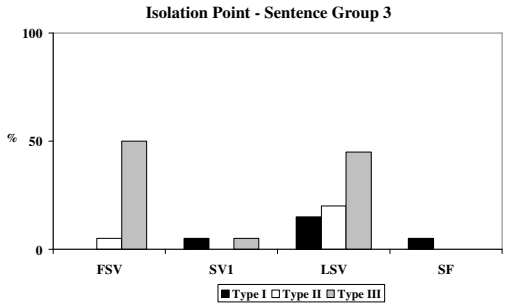


Figure 9. *Percent results for answer 'question' IP in Sentence Group 3 (all subjects included)*

3.2.2. Recognition Point

Like in the analysis of the first hypothesis, the results for *RP* are more diverse than *IP* ones. However, they are clearer because most of the *RP* results available are located in the Last Stressed Vowel (LSV), except for Sentence Group 1 (figure 10) in which they occur in the Stressed Vowel 1 (SV1).

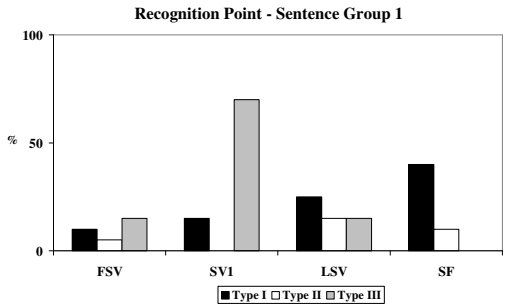


Figure 10. *Percent results for answer 'question' RP in Sentence Group 1 (all subjects included)*

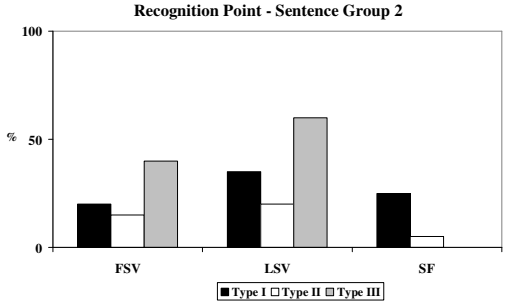


Figure 11. *Percent results for answer 'question' RP in Sentence Group 2 (all subjects included)*

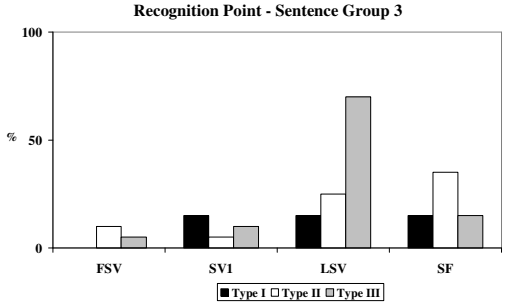


Figure 12. *Percent results for answer 'question' RP in Sentence Group 3 (all subjects included)*

We didn't detect any symmetry between *IP* and *RP* results as can be seen in Table 4.

	Type I		**Type II**		**Type III**	
	IP	***RP***	***IP***	***RP***	***IP***	***RP***
SG1	FSV	LSV	---	---	FSV	SV1
SG2	FSV	LSV	FSV	---	FSV	LSV
SG3	---	---	---	SF	FSV	LSV

Table 4. *Location results for IP and RP 'question' answer. Shadowed cells refer to sentence beginning (all subjects included)*

4. DISCUSSION AND CONCLUSIONS

Statements are identified at the first stressed vowel in the sentence while questions (except the question in pair C) are identified by the last or the penultimate stressed vowel of the sentence.

A detailed acoustic analysis of the fundamental frequency of the question in pair C shows that its intonation contour presents specificities that facilitate its earlier identification when compared with the questions in the other pairs: an initial high and steady F0 plateau that points to a high pitch range, usual in questions. The availability of this early prosodic cue in the sentence allows for sooner identification.

As expected, subjects take longer to be confident on their own answers: recognition points for every sentence occur later than isolation ones, except for the question in pair B.

The fact that listeners are able to identify declarative sentence type by the end of the first stressed vowel of the sentence means that prosodic cues are available in the speech signal at this early location. These prosodic cues can be, for instance, the pitch peak of the sentence that allows subjects to place the boundaries of pitch range that is a global phenomenon. However, question results refer to the need of waiting for sentence end to identify interrogative sentence type. As we know, it is a final local rising movement that usually distinguishes questions from other sentence types. Therefore, both global and local data play an important role in intonation processing.

Both *IP* and *RP* results' reveal that EP listeners start to build their internal representation of intonation very early in the sentence.

Results also showed that the lexical cues power for sentence type processing is stronger than the prosodic ones because sentences starting with *wh*-words, whatever semantic or syntactic function or prosodic information they have tend to be classified as questions at first (see *IP* results in table 4). However, Sentence Group 3 stimuli results do not show the same behavior. We believe that this may be due to the different strength of interrogativity of the words. Probably, the *wh*-word that starts these stimuli is less 'marked' for question interpretation and, at this stage, the prosodic cues are enough to reject the classification as question. So, it seems that the presence of *wh*-words that have a heavy question interpretation allows for a question classification even without a proper question intonation in the sentence. By the contrary, if the lexical strength for interrogativity is poor, like in Sentence Group 3 stimuli, prosodic data became stronger.

The interplay of all cues seems to occur later in the sentence, after more information becomes available.

5. REFERENCES

[1] Viana, M.C., *Para a síntese da entoação do português*. PhD Dissertation, University of Lisbon, 1987.

[2] Frota, S., "Nuclear falls and rises in european portuguese: a phonological analysis of declarative and question intonation", *Probus*, vol. 14 (1), 113-146, 2002.

[3] Mata, A.I., Pereira, I., "Flexibilidade de contornos entoacionais em sequências de natureza interrogativa: percepção e interpretação", *Actas do VII Encontro Nacional da Associação Portuguesa de Linguística*, 211-224, Lisboa, 1991.

[4] Falé, I., *Percepção e Reconhecimento da Informação Entoacional em Português Europeu*. PhD Dissertation, University of Lisbon, 2006.

[5] Hawkins, S., "Contribution of fine phonetic detail to speech understanding", *Proceedings of the 15th ICPhS*, Barcelona, 293-296, 2003.

[6] Grosjean, F., "Spoken word recognition processes and the gating paradigm", *Perception and Psychophysics* 28 (4), 267-283, 1980.
[7] Grosjean, F., Hirt, C., "Using prosody to predict the end of sentences in English and French: normal and brain-damaged subjects", *Language and Cognitive Processes* 11 (1/2), 107-134, 1996.
[8] Grosjean, F., "Gating", *Language and Cognitive Processes* 11 (6), 597-604, 1996.
[9] Schneider, W., Eschman, A., Zuccolotto, A., *E-Prime reference guide*. Psychology Software Tools, Pittsburgh, 2002.
[10] Falé, I., Faria, I.H., "A glimpse of the time course of intonation processing", *Proceedings of the 9th European Conference on Speech Communication and Technology*. Lisboa, 2377-2380, 2005.

MULTIMODAL PERSON TRACKING IN A SMART-ROOM ENVIRONMENT

C. Segura, A. Abad, C. Nadeu, J. Hernando

Technical University of Catalonia, Barcelona, Spain
{csegura,alberto,climent,javier}@gps.tsc.upc.edu

ABSTRACT

Tracking speakers in multiparty conversations constitutes a fundamental task for automatic meeting analysis. In this work, we present the Person Tracking systems developed at UPC for audio, video and audio-video modalities. The proposed systems are designed to deal robustly in both single and multiperson localization task independently on the environmental conditions. Novelties proposed are aimed to enhance the accuracy of the system independently on the application scenario and to reduce the computational complexity. Besides the technology description, experimental results conducted for the CLEAR evaluation workshop are also reported.

1. INTRODUCTION

The automatic analysis of meetings recorded in multi-sensor rooms is an emerging research field. In this domain, localizing and tracking people and their speaking activity play fundamental roles in several applications, like microphone array beamforming or steering of pan-tilt-zoom cameras towards the active speaker. To locate persons with unobtrusive far-field sensors, either video or audio sources can be used, though eventually the most accurate and robust techniques will likely be based on multimodal.

The degree of reliable information provided by person localization systems on the basis of the audio and video signals collected in a smart-room environment with a distributed microphone and video network, depends on a number of factors such as environmental noise, room reverberation, person movements and camera occlusions. These factors, among others, demand an effort on the development of new robust systems capable of dealing with adverse environments.

In the present work, we get an insight on the development and design of robust Person Tracking systems based on audio, video and audio-video modalities. Results obtained in the CLEAR evaluation campaign and comparison among mono and multimodal systems are provided showing the performance of the proposed algorithms.

This work has been partially sponsored by the EC-funded project CHIL [1] (IST-2002-506909) and by the Spanish Government-funded project ACESCA (TIN2005-08852).

2. AUDIO PERSON TRACKING SYSTEM

Conventional acoustic person localization and tracking systems can be split into three basic stages. In the first stage, estimations of such information as Time Difference of Arrival or Direction of Arrival is usually obtained from the combination of the different microphones available. In general, in the second stage the set of relative delays or directions of arrival estimations are used to derive the source position that is in the best accordance with them and with the given geometry. In the third optional stage, a tracking of the possible movements of the sources according to a motion model can be employed.

The SRP-PHAT [2] algorithm (also known as Global Coherence Field [3]) performs and integrates the two first stages of localization in a robust and smart way. In general, the goal of localization techniques based on SRP (Steered Response Power) is to maximize the power of the received sound source signal using a delay-and-sum or a filter-and-sum beamformer. In the simplest case, the output of the delay-and-sum beamformer is the sum of the signals of each microphone with the adequate steering delays for the position that is explored. Thus, a simple localization strategy is to search for the energy peak through all the possible positions in 3D space. Concretely, SRP-PHAT algorithm searches for the maximum of the contribution of the cross-correlations between all the microphone pairs across the space. The main strength of this technique consists on the combination of the simplicity of the steered beamformer approach with the robustness offered by the PHAT weighting.

The proposed system for Audio Person Tracking is based on the SRP-PHAT algorithm with some additional robust modifications. The system design has been aimed to develop a robust system with independence on the acoustic and room conditions, such as the number of sources, their maneuvering modes or the number of microphones.

2.1. Brief Description of the SRP-PHAT Algorithm

As already mentioned above, the SRP-PHAT algorithm searches for the maximum of the contribution of the cross-correlations between all the microphone pairs across the space. The process can be summarized into four basic steps:

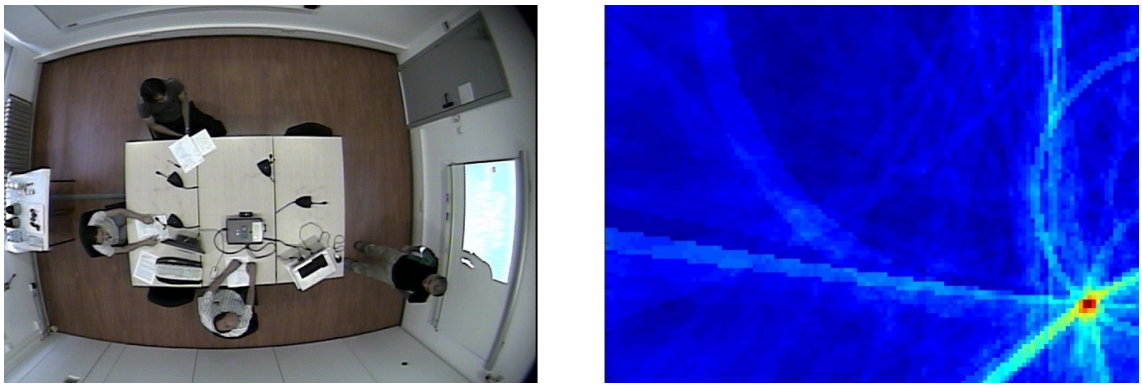


Figure 1. On the left, zenithal camera snapshot. On the right, example of the *Sound Map* obtained with the SRP-PHAT process.

Step 1 The exploration space is firstly split into small regions (typically of 5-10 cm). Then, theoretical delays from each possible exploration region to each microphone pair is pre-computed and stored.

Step 2 Cross-correlations of each microphone pair are estimated for each analysis frame. Concretely, the Generalized Cross Correlation with PHAT weighting [4] is considered. It can be expressed in terms of the inverse Fourier transform of the estimated cross-power spectral density ($\widehat{G}_{x_1x_2}(f)$) as follows,

$$\widehat{R}_{x_1x_2}(\tau) = \int_{-\infty}^{\infty} \frac{\widehat{G}_{x_1x_2}(f)}{|\widehat{G}_{x_1x_2}(f)|} e^{j2\pi f\tau} df \quad (1)$$

Step 3 The contribution of the cross-correlations is accumulated for each exploration region using the delays pre-computed in *Step 1*. In this way, it is obtained a kind of *Sound Map* as the one shown in Figure 1.

Step 4 Finally, the position with the maximum score is selected as the estimated position.

2.2. The Implementation of the Robust Audio Person Tracker

On the basis of the conventional SRP-PHAT, a robust system for Audio Person Tracking is developed. The main novelties introduced and some aspects related to other implementation details are introduced in the following.

2.2.1. Implementation Details

The analysis frame consists of Hanning windowed blocks of 4096 samples, 50% overlapped, obtained at a sample rate of 44.1 kHz. The FFT computation dimension is fixed to 4096 samples.

2.2.2. Adaptive Smoothing Factor for the Cross-Power Spectrum (CPS) Estimations

Smoothing over time of the GCC-PHAT estimations is a simple and efficient way of adding robustness to the system. This smoothing can be done in the time domain (GCC-PHAT) or in the frequency domain (CPS). Considering the smoothed cross-power spectrum $\widehat{G}_{x_1x_2}(k,f)$ in time instant k and the instantaneous estimation $G_{x_1x_2}(k,f)$ our system performs the smoothing in the frequency domain as follows,

$$\widehat{G}_{x_1x_2}(k,f) = \beta\widehat{G}_{x_1x_2}(k-1,f) + (1-\beta)G_{x_1x_2}(k,f) \quad (2)$$

From experimental observation it can be seen that the right selection of this β factor is crucial in the system design. A high smoothing value can greatly enhance the results obtained in an almost static scenario, while it can be dramatically inconvenient in a scenario with many moving speakers.

Hence, an adaptive smoothing factor has been designed. This adaptive factor is obtained based on the velocity estimation provided by a Kalman filter.

2.2.3. Two-Pass SRP Search

It can be seen from experimental observations that most of the information for a rough localization is concentrated in the low-frequency bins of the GCC-PHAT, while high frequency bins are useful in order to obtain a finest estimation given a first coarse estimation. Taking into account this observation a two-pass SRP search has been designed:

Coarse Search This search procedure is performed only in the *x-y* axis (z is assumed to be 1.5 m), with a searching cell dimension of 16 cm and only using the low frequency information of the cross-correlations ($f < 9kHz$). A first coarse estimation is obtained from this search, say $(x_1, y_1, 150)$ cm.

Fine Search A new limited search area around the obtained *coarse* estimation is defined $(x_1 - 50 : x_1 + 50, y_1 - 50 : y_1 + 50, 110 : 190)$ cm. In

this new fine search, dimension of the cell search is fixed to 4 cm for the *x*-*y* axis and to 8 cm for the *z*-axis. In the *fine search* all the frequency information of the cross-correlations is used and a more accurate estimation is obtained.

Moreover, the double SRP searching procedure is adequate to reduce computational load, since the *fine* exploration is only performed across a very limited area.

2.2.4. Confidence Threshold

In SRP-PHAT algorithm the position with the maximum value obtained from the accumulated contributions of all the correlations is selected (*Step 4*). This value is assumed to be well-correlated with the likelihood of the given estimation. Hence, this value is compared to a fixed threshold (depending on the number of microphone-pairs used) to reject/accept the estimation. The threshold has been experimentally fixed to 0.5 for each 6 microphone pairs.

Finally, it is worth noting that although a Kalman filter is used for the estimation of the adaptive CPS smoothing factor, it is not considered for tracking purposes. The reason is that the Kalman filter design and the data association strategies adopted showed a different impact depending on the scenario. In other words, it showed to be too much dependent on the number and the velocities of sources to perform correctly.

3. VIDEO PERSON TRACKING SYSTEM

For this task we propose creating a 3D representation of the room combining the views from a calibrated [5] set of cameras. The scene is discretized into box-shaped regions, voxels, and then each voxel is classified as foreground or background. Indeed, the foreground voxels provide enough information for precisely object detection and tracking. The image at each camera view is segmented forming foreground regions. This foreground regions are then projected and combined in the 3D space forming volumes, where the 3D objects must lie, using the Shape from silhouette technique. The main drawback of the method is that it doesn't always capture the true shape of the object, as concave shape regions are not expressed in the silhouettes. However, this is not a severe problem in a tracking application as the aim is not to reconstruct photorealistic scenes.

After the voxelization process (see figure 2), a connected component analysis *CCA* follows to cluster and label the voxels into meaningful 3D-blobs [6, 7, 8] , from which some representative features are extracted. Finally, there is a template-based matching process aiming to find persistent blob correspondences between consecutive frames.

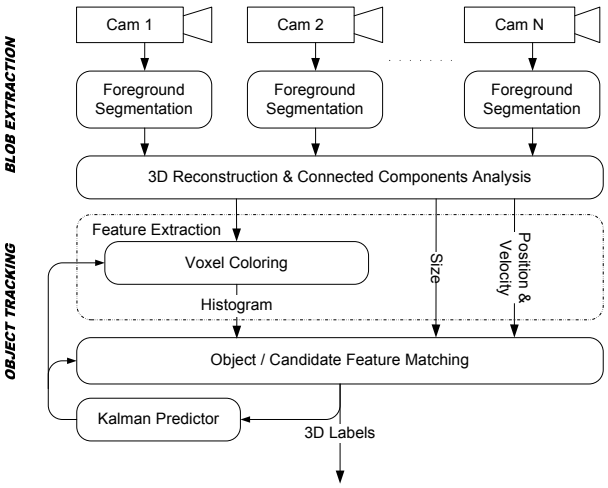


Figure 2. The system block diagram showing the chain of functional modules

4. MULTIMODAL PERSON TRACKING SYSTEM

Multimodal Person Tracking is done based on the audio and video person tracking technologies described in the previous sections. These two technologies may have different nature, for example different frame rate, the video tracking system is able to track several persons, but usually only one person estimate is given by the audio tracking system and only when actively speaking, etc. A multi-modal system aiming on the fusion of information proceeding from these two technologies has to take into account these differences.

We expect to have far more position estimates from the video system than from the audio system since persons in the smart room are visible by the cameras during most of the video frames; in contrary, the audio system can estimate the person's position only if she/he is speaking (so called active speaker). Thus, the presented multimodal approach relies more on the video tracking system and it is extended to incorporate the audio estimates to the corresponding video tracks. This is achieved by first synchronizing the audio and video estimates and then using data association techniques. After that a decentralized Kalman filter is used to provide a global estimate of person's position. The frame rate of the multimodal tracking is the same as that of the video system.

4.1. Audiovisual Fusion

The Kalman filter algorithm provides an efficient computational solution for recursively estimating the position, in situations where the system dynamics can be described by a state-space model. A detailed description of the Kalman filter for tracking can be found in [10, 11].

The decentralized Kalman filter [12] is used for the fusion of audio and video position estimates. As

shown in Figure 3, the system can be divided in two modules associated with the audio and video systems. Each modality computes a local a-posteriori estimate $\hat{\mathbf{x}}_i[k|k]$, $i = 1, 2$ of the person position using a local Kalman filter (KF1 and KF2, respectively), based on the corresponding observations $\mathbf{y}_1[k]$, $\mathbf{y}_2[k]$. These partial estimates are then combined to provide a global state estimate $\hat{\mathbf{x}}[k|k]$ at the fusion center such as:

$$\hat{\mathbf{x}}[x|x] = \mathbf{P}[k|k]\Big(\mathbf{P}^{-1}[k|k-1]\hat{\mathbf{x}}[k|k-1] + \sum_{i=1}^{2}\Big[\mathbf{P}_i^{-1}[k|k]\hat{\mathbf{x}}_i[k|k] - \mathbf{P}_i^{-1}[k|k-1]\hat{\mathbf{x}}_i[k|k-1]\Big]\Big) \quad (3)$$

$$\mathbf{P}^{-1}[k|k] = \mathbf{P}^{-1}[k|k-1] + \sum_{i=1}^{2}\Big[\mathbf{P}_i^{-1}[k|k] - \mathbf{P}_i^{-1}[k|k-1]\Big] \quad (4)$$

The global estimate of the system state is obtained weighting the global and local estate estimate with the global error covariance matrix $\mathbf{P}[k|k]$ and their counterparts $\mathbf{P}_i[k|k]$ at the audio and video systems.

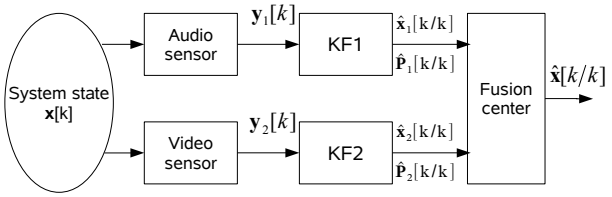


Figure 3. Structure of the decentralized Kalman filter. The fusion center combines the local estimates to compute a global estimate of the system state.

5. EVALUATION

Person Tracking evaluation is run on the data collected by the CHIL consortium for the CLEAR 06 evaluation. Two tasks are considered: single and multiperson tracking, based on non-interactive seminar (collected by ITC and UKA) and highly interactive seminar (collected by IBM, RESIT and UPC) recordings, respectively. Complete description of the data and the evaluation can be found in [13].

5.1. Summary of the Experimental Set-Up

5.1.1. Data description

Room set-ups of the contributing sites present two basic common groups of devices: the *audio* and the *video* sensors.

Audio sensors set-up is composed by 1 (or more) NIST Mark III 64-channel microphone array, 3 (or more) T-shaped 4-channel microphone cluster and various table-top and close-talk microphones.

Video sensors set-up is basically composed by 4 (or more) fixed cameras. In addition to the fixed cameras, some sites are equipped with 1 (or more) PTZ camera.

5.1.2. Evaluation metrics

Three metrics are considered for evaluation and comparison purposes:

Multiple Object Tracking Precision (MOTP) [mm] This is the precision of the tracker when it comes to determining the exact position of a tracked person in the room. It is the total Euclidian distance error for matched *ground truth-hypothesis* pairs over all frames, averaged by the total number of matches made.

Multiple Object Tracking Accuracy (MOTA) [%] This is the accuracy of the tracker when it comes to keeping correct correspondences over time, estimating the number of people, recovering tracks, etc. It is the sum of all errors made by the tracker, false positives, misses, mismatches, over all frames, divided by the total number of ground truth points.

Acoustic Multiple Object Tracking Accuracy (A-MOTA) [%] This is like the *original* MOTA metric in which all mismatch errors are ignored and it is used to measure tracker performance only for the active speaker at each point in time for better comparison with the acoustic person tracking results (where identity mismatches are not evaluated).

5.2. Audio Person Tracking Results

We have decided to use all the *T-clusters* available in the different seminars and only to use the *MarkIII* data of those sites where the *MarkIII* is located in a wall without a *T-cluster* (IBM, RESIT and UPC). In general, only microphone pairs of the same *T-cluster* or *MarkIII* array are considered by the algorithm.

In the experiments where the *MarkIII* is used, 6 microphone pairs are selected for GCC-PHAT computation The pairs selected out of the 64 microphones of *MarkIII* are *1-11*, *11-21*, *21-31*, *31-41*, *41-51* and *51-61*. Hence, an inter-microphone separation of 20 cm for each microphone-pair is considered.

In Table 1 individual results for each data set and average results for both tasks are shown. Notice that task results are not directly the mean of the individual results, since the scores are recomputed jointly. The evaluating system in both tasks is the same and the multi-person task is only evaluated when only one speaker is active. In this way mean performances obtained, as it could be expected, are quite similar. In fact, there is a fail in the multi-person task, but it is more related with

the particular characteristics of each data set, that with the task indeed. For instance, UPC data is particularly noisy and present some challenging situations such as *coffee breaks*. Hence, we can conclude that acoustic tracking performs reasonably well in controlled scenarios with one or few alternative and non-overlapping speakers, while it shows a considerable decrease in difficult noisy scenarios with many moving and overlapping speakers.

Table 1. Audio results for both single and multi-person tracking.

Task	MOTP	Misses	False Positives	A-MOTA
ITC data	108mm	8.56%	1.46%	89.98%
UKA data	148mm	15.09%	10.19%	74.72%
Single Person	145mm	14.53%	9.43%	76.04%
IBM data	180mm	17.85%	10.54%	71.61%
RESIT data	150mm	12.96%	6.23%	80.80%
UPC data	139mm	32.34%	28.76%	38.89%
Multi Person	157mm	20.95%	15.05%	64.00%

5.3. Video Person Tracking Results

Seminar sequences from UPC and RESIT have been evaluated and results are reported in Table 2. Since our algorithm required empty room information, we were constrained to only evaluate UPC and RESIT. By analyzing the results in detail we reached the following conclusions.

The high number of False Positives (FP) is mainly due to the fact our algorithm detected many foreground objects after the 3D reconstruction due to shadows and other lighting artifacts. Moreover, MOTA is related with the FP score thus dropping as FP increases. Further research to avoid such problems include an improvement of the Kalman filtering and association rules. Since our tracking strategy relies on the 3D reconstruction, rooms with a reduced common volume seen by a number of cameras (typically less N-1 cameras) produce less accurate results. Other reconstruction schemes more accommodated to different camera placement scenarios are under research to generate reliable volumes even if a reduced number of cameras is viewing a given part of the room.

Table 2. Video results for the multiperson tracking.

Task	MOTP	Misses	False Pos.	Mism.	MOTA
RESIT data	205mm	26.67%	74.62%	2.18%	-3.47%
UPC data	188mm	16.92%	23.56%	5.85%	53.67%
Multi Person	195mm	21.24%	46.16%	4.22%	28.35%

5.4. Multimodal Person Tracking Results

Only seminar sequences from RESIT and UPC have been evaluated due to the constrains of the Video tracking system mentioned above. For the Multimodal Person Tracking task, two different scorings under two different conditions are defined. For the condition A, the scoring shows the ability to track the active speaker at the time segments that he is speaking, while under the condition B the scoring measures the ability to track all the persons in the room during all the seminar.

The results are reported in Tables 3 and 4 for each condition. It can be seen that the results are very similar to those of the Video Person tracking task. This observation suggests that the multimodal algorithm is mainly influenced by the performance of the video tracking system.

Table 3. Multimodal results for Condition A.

Task	MOTP	Misses	False Pos.	Mism.	A-MOTA
RESIT data	143mm	52.66%	7.14%	3.92%	40.20%
UPC data	101mm	29.48%	25.28%	6.35%	45.24%
Cond. A	118mm	41.18%	16.13%	5.12%	42.70%

Table 4. Multimodal results for Condition B.

Task	MOTP	Misses	False Pos.	Mism.	MOTA
RESIT data	201mm	26.43%	74.47%	2.20%	-3.10%
UPC data	190mm	17.95%	24.61%	5.98%	51.46%
Cond. B	195mm	21.71%	46.71%	4.31%	27.28%

6. CONCLUSIONS

In this paper we have presented the audio, video and audio-video Person Tracking systems developed by the UPC.Novelties proposed in the three systems have been specially designed to add robustness to scenario and environment variabilities. Results obtained in the CLEAR evaluation campaign show that the audio tracker performs reasonably well in situations with few non-overlapping speakers, while it shows a considerable loss of performance in some challenging and noisy situations that must be addressed. Improvement of the Kalman filtering and association rules are also expected to enhance the video system. Finally, the multimodal audio-video system shows a high dependence on the video results caused by the fusion procedure. Thus, future efforts will be devoted to develop new fusion strategies at a higher level.

7. REFERENCES

[1] CHIL Computers In the Human Interaction Loop. Integrated Project of the 6th European Framework Programme (506909). http://chil.server.de/, 2004-2007

[2] DiBiase, J., Silverman, H., Brandstein, M.: Microphone Arrays. Robust Localization in Reverberant Rooms, Chapter 8, Springer, January 2001

[3] Brutti, A., Omologo, M. , Svaizer, P.: Oriented global coherence field for the estimation of the head orientation in smart rooms equipped with distributed microphone arrays. Proceedings of Interspeech 2005, Lisboa September 2005

[4] Knapp, C.H. and Carter, G.C.: The Generalized Correlation Method for Estimation of Time Delay Rooms. IEEE Trans. on Acoustics, Speech, and Signal Processing August 1976

[5] Zhang, Z.: A flexible new technique for camera calibration. Technical report, Microsoft Research, August 2002

[6] Landabaso, J.L., Xu, L-Q., Pardàs, M.: Robust Tracking and Object Classification Towards Automated Video Surveillance. Proceedings of ICIAR **2** 2004 463–470

[7] Xu, L-Q., Landabaso, J.L. Pardàs, M.: Shadow removal with blob-based morphological reconstruction for error correction. Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on, **Vol.2**, Iss., March 18-23, 2005 729–732

[8] Landabaso, J.L., Pardàs, M., Xu, L-Q.: Hierarchical representation of scenes using activity information. Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on, **Vol.2**, Iss., March 18-23, 2005 677–680

[9] Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE trans. on Pattern Analysis and Machine Intelligence, **22(8)**, August 2000

[10] Bar-Shalom, Y., Fortman,T.E.: Tracking and Data association. Academic Press, 1988

[11] Sturim, D. E., Brandstein, M. S., Silverman, H. F.: Tracking Multiple Talkers Using Microphone-Array Measurements. Proceedings of ICASSP 1997, Munich, April 1997

[12] Hashemipour, H. R., Roy, S., Laub, J.: Decentralized structures for parallel Kalman filterin. IEEE Transactions on Automatic Control, 33(1):88-93, 1988

[13] The Spring 2006 CLEAR Evaluation and Workshop. http://www.clear-evaluation.org/

ITERATIVE SPEAKER ADAPTATION USING MLLR

José R. Navarro Cerdán[1], *Carlos Martínez Hinarejos*[2],
Antonio L. Lagarda Arroyo[2], *Luis Rodríguez Ruiz*[1]

[1]Instituto Tecnológico de Informática
[2]Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia, Cno. Vera s/n, 46022, Valencia, Spain
jonacer@iti.upv.es,cmartine@dsic.upv.es,alagarda@dsic.upv.es,lrodrig@iti.upv.es

ABSTRACT

Speech recognition systems are usually speaker-independent, but they are not as good as speaker-dependent systems for specific speakers. An initial speaker-independent system can be adapted to improve recognition accuracy by transforming it into a speaker-dependent system. In this work, a new general acoustic model adaptation technology is presented, using the MLLR algorithm iteratively in a supervised manner. Experiments have been performed on the TT2 Spanish speech corpus. The initial acoustic models were trained from the Albayzin speech database. Their results, which were obtained for 10 speakers, show an improvement in speech recognition accuracy.

1. INTRODUCTION

Speech recognition improvements have contributed to the widespread use of speech recognition systems in several applications [1, 2, 3]. Speech recognition systems rely on acoustic and language models to perform the recognition of input utterances. This work deals with only one part of speech recognition system, acoustic model. They model sequences of feature vectors that describe a specific sound (phonemes, syllabes, etc.). They are usually continuous-density Hidden Markov Models (HMM) [4, 5], in which each state models its output distribution using a mixture of Gaussians. Each Gaussian is defined by a feature mean vector and a covariance matrix.

Parameter estimation of acoustic models is done by means of the well-known Baum-Welch algorithm [6]. A good estimation of these models requires a lot of training data. This makes speaker-independent systems common because obtaining a large amount of training data for systems of this kind is easier than obtaining a large amount of data for speaker-dependent systems. However, for a specific speaker, more accurate results can be achieved by using speaker-dependent acoustic models, provided that sufficient data is available. Unfortunately, obtaining enough speaker-specific data for speaker-dependent acoustic model estimation is very difficult.

The solution consists of obtaining a speaker-dependent acoustic model by adapting a speaker-independent acoustic model to a specific speaker, using only a small amount of specific-speaker data.

2. TYPES OF ADAPTATION TECHNIQUES

Several speaker adaptation techniques have been developed in the last few years [7, 8, 9]. These techniques can be divided into two main groups, depending on what is modified (*the input signal* or *the acoustic model*). The most important speaker adaptation techniques are described in [10, 11].

2.1. Spectral mapping techniques

In these techniques, modifications are made on the input signal. The acoustic signal (or its codification) is altered to adapt the signal to a general acoustic model; therefore, with these techniques, the signal of the new speaker is closer to the signal of the reference speakers. The main techniques are:

- *Dynamic Time Warping (DTW)*[12]: Dynamic time-warping is a dynamic programming algorithm that finds the reference signal alignment that minimizes the distance to input signal.
- *Spectral-Bias*[13]: This method uses the information incorporated in speaker-independent Hidden Markov Models (HMM) and estimates a transformation of the means of the models. Although this method transforms the means of the HMM, it is included in this group because the goal of the method is to improve the match between the reference speakers and the new speaker (the *spectral mapping* idea) rather than to improve the modeling accuracy for the new speaker (the *model mapping* idea).
- *Vocal Tract Length Normalizacion (VTLN)*[14]: Human vocal tract length produces variation in the main components of the source speech signal. Assuming

This work has been partially supported by Spanish CICYT under project TIC2003-08681-C02-02 and by the European Union under grant IST-2001-32091

a different vocal tract length for each individual speaker, source speech signal from a speaker can be transformed into a normalized signal using the VTLN algorithm. This signal can then be used to train the acoustic models. This technique uses a simple transformation function that depends on a parameter (warping factor) and the signal frequency in each instant.

2.2. Model mapping techniques

In these techniques, modifications are made on the acoustic models. In this case, acoustic models are altered in order to make them closer to the source input signal from the speaker. In other words, we approximate general acoustic models to the input signal from the speaker. The main techniques are:

- *Maximum adaptation a posteriori (MAP)*[8]: This is a general probability distribution estimation technique that allows previous knowledge to be introduced in the estimation process. In this case, the knowledge is the parameters of the speaker independent acoustic models.
- *Regression-based Model Prediction, (RMP)*[15]: This method is based on linear regression. The idea consists of using the available adaptation material to make an initial *maximum a posteriori (MAP)* adaptation for the model means. These *MAP* estimates are then used to predict the means of the models which were not present in the adaptation data; this is done via a set of regression coefficients, which are computed using previously trained speaker-dependent models.
- *Maximum Likelihood Linear Regression, (MLLR)*[7]: With this method, the feature means of general acoustic models are adapted to the speaker's voice using a linear regression model that is estimated by means of maximum likelihood. This is the method that we used for our speaker adaptation system and is explained below in Section 3.

In this article, the results obtained with the iterative application of *Maximum Likelihood Linear Regression Model* in a supervised fashion are presented. Our technique is based on making successive speaker adaptations by means of the MLLR algorithm, to improve the speaker's acoustic models reusing the adaptation data. The final goal is to obtain better accuracy in speech recognition for that specific speaker.

3. THE MLLR SPEAKER ADAPTATION TECHNIQUE

This method is based on the application of an adaptation matrix, *W*, on the acoustic model parameters. This matrix *W* is computed by means of maximum likelihood, with the general acoustic model parameters without adaptation and the speaker voice to be adapted as input data. An acoustic model that is completely adapted to the speaker is obtained by applying this method.

To start with the MLLR algorithm description [7, 11], we must consider the case of a continuous density HMM system with Gaussian output distributions. A particular distribution, s, will be characterised by a mean vector, μ_s, and a covariance matrix C_s. Given a parametrized speech frame vector o, the probability density of that vector being generated by distribution s will be $b_s(o)$

$$b_s(o) = \frac{1}{(2\pi)^{n/2}} e^{-1/2(o-\mu_s)' C_s^{-1} (o-\mu_s)}$$

where n is the dimension of the observation vector and $'$ denotes the transpose vector.

The MLLR algorithm can be summarized as follows: for each Gaussian s, compute the new speaker estimated $\hat{\mu}_s$ parameter from the general μ_s parameter. This is obtained using:

$$\hat{\mu}_s = W_s \xi_s$$

where:

- W_s is the adaptation matrix.
- $\xi_s = [\omega, \mu_{s_1}, \ldots, \mu_{s_n}]$ is the extended vector of means, with shift ω.

This computation can also be done for the covariance matrix, but this additional adaptation does not usually provide better results [11]. Thus, the probability density function for the adapted system for the Gaussian s is:

$$b_s(o) = \frac{1}{(2\pi)^{n/2}} e^{-1/2(o-W_s\xi_s)' C_s^{-1} (o-W_s\xi_s)}$$

Since it is usually not possible to estimate W_s for each Gaussian s, *regression classes* are defined as Gaussian sets that share the same adaptation matrix.

The number and optimum composition of *regression classes* cannot be defined analytically. Thus, its selection is usually based on the amount of available adaptation data, the phonetic split between models (decision trees), and the different join critera between models (phonetic features, distance between models, etc).

To estimate the transformation matrix, given a *regression class* $R = \{s_1 k, s_2 k, \ldots, s_R k\}$, W_s is estimated by maximum likelihood:

$$\hat{W}_s = \max_{W_s} \Pr(O_p | \hat{\lambda})$$

where:

- O_p is the sequence of observations.

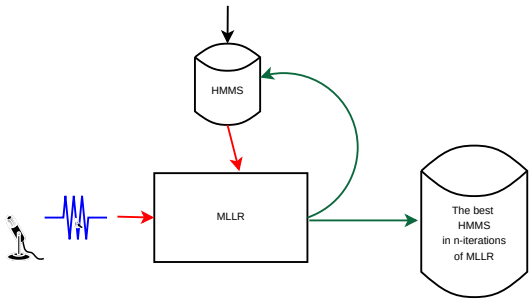


Figure 1. Iterative MLLR architecture.

- $\hat{\lambda}$ is the model obtained applying W_s.
- $\hat{W}_s$ is the optimum estimation of the adaptation methods.

$\hat{W}_s$ is obtained with the optimization of an auxiliar function, Q.

$$Q(\lambda, \hat{\lambda}) = \sum_{\theta \in \Theta} \sum_{k \in \Omega_b} \Pr(O_p, \theta, k|\lambda) \log(\Pr(O_p, \theta, k|\hat{\lambda}))$$

where,

- Θ is the state sequence set.
- Ω_b is the Gaussian set.

The estimation of $\hat{W}_s$ with this formulation is usually complicated and time-consuming. To estimate it, a Viterbi approximation can be used, which corresponds to the following formula:

$$\hat{W}_s = \left(\sum_{t=1}^{T} O^t \xi'_{s_r k} \right) \left(\sum_{t=1}^{T} \xi_{s_r k} \xi'_{s_r k} \right)^{-1} \quad (1)$$

To apply this approach, the observations O^t are initially decoded in a forced way (using a transcription). The decoding process gives the Gaussian with maximum likelihood for each observation, whose mean is $\xi_{s_r k}$. With this data, Equation (1) is applied to compute $\hat{W}_s$.

4. ADAPTATION ARCHITECTURE

This paper provides new results for speaker adaptation using the MLLR algorithm iteratively instead of only once. The results show that successive adaptations of adapted models can significantly improve the results.

Figure 1 shows the architecture of the method. Initially, the input data are simple wave files of sentences, that have been transcribed, obtained from the speaker to be adapted; they are the initial source of information. These wave files are passed to the MLLR algorithm along with the general acoustic model to be adapted. MLLR gives a new acoustic model that is adapted to the speaker.

This classic application of the MLLR algorithm can be improved by readapting the first adapted acoustic model using the same method. In a second iteration of the algorithm, the first adapted acoustic model acts as the new general acoustic model, which is then adapted using the MLLR algorithm (see the feed-back in Figure 1). Thus, the MLLR algorithm uses the same wave files, that were transcribed initially. In the following iterations, the same wave files and their transcriptions (which are supervised) are used to reestimate a new acoustic model substituting the original acoustic model with the new adapted model. If several iterations are done and the sentence accuracy rate (SAR) of a test set of the adapted speaker for each adaptation is obtained, it is possible to determine whether the results have been improved. This will also obtain the best acoustic model.

Our initial set of acoustic models was obtained from the Albayzin Spanish speech corpus [16]. The acoustic models were Hidden Markov Models (HMM) that represented monophones. Their topology is the classical three-state, left-to-right with loops and without skips. The output distribution for each state was modelled by a mixture of 128 Gaussians with diagonal covariance matrixes. The number of components of the Gaussians was 33 (ten cepstrals plus energy, plus first derivative and acceleration).

5. CORPUS DESCRIPTION

The TT2 project [17] is devoted to the construction of Computer Aided Translation (CAT) systems. In this project, text translation is combined with speech input to improve the performance of the human translator. The usual scenario of an interaction between the computer application and the human translator follows these steps:

1. The computer application proposes a translation of the current sentence.
2. The human translator accepts part (a prefix) of the proposed translation.
3. The human translator types in possible corrections.
4. The computer application dynamically changes its proposed translation as the human translator types in the corrections.
5. The human translator returns to step 2 until the current sentence is completely translated.

There are several ways in which the human translator can accept a prefix. In the classical approach, s/he points with the mouse at (or uses the keyboard to move to) the end of the correct part of the sentence. In the case of speech input, s/he can utter any subsentence (one or more words) that is present in the translation, and which is perhaps preceded by the words “accept” and/or “until”. The prefix up to that subsentence will be accepted. In case

Table 1. Some examples of uttered subsentences and selected prefixes for the proposed sentence "adición de fuentes a la lista de recursos".

Uttered sentence	Selected prefix
lista	*adición de fuentes a la lista*
aceptar hasta fuentes	*adición de fuentes*
hasta de	*adición de*
hasta de recursos	*adición de fuentes a la lista de recursos*

of ambiguity, the accepted prefix will be the shortest one. Some examples are presented in Table 1.

A speech corpus was acquired to simulate this scenario when translating Xerox printer manuals to Spanish (Xerox corpus)[17]. This acoustic corpus consisted of a total of 7,489[1] utterances of subsentences derived from the sentences of the task. These subsentences were utterances of 125 complete sentences of the task, which were chosen from the Xerox corpus. Five segmentations into prefixes and suffixes were randomly performed on this set. A random prefix was selected for each suffix generated. The words "aceptar", "hasta", and "aceptar hasta" were added as prefixes to some of these selected subsentences, giving a total number of 625 different sentences to be uttered. A sample of possible segmentations for a sentence is presented in Table 2.

Ten speakers (six male, four female) were recruited. The sentences were divided into five different sets of 125 sentences. One of these sets was chosen as the adaptation set and was common to all the speakers; the other four sets were distributed among the speakers (five speakers shared two of these sets and the other five shared the remaining two sets). The acquisition was performed by each speaker in three different sessions (at different times of the day in order to capture variabilities in speech intonation). Different subsets of the groups were acquired in each session. In each session, the selected speaker uttered a total of 250 utterances (i.e., the selected sentences were repeated twice). Thus, this acquisition gave a total number of 750 utterances per speaker. Of these, 250 sentences were selected to be used for speaker adaptation, and the other 500 sentences were selected to test the system. Both groups of sentences were disjoint.

The acquisition was performed using a high quality microphone, at 16kHz sampling rate and 16 bits per sample. The total duration of the acquired signal was nearly 5 hours, although nearly half of the acquired signal was silence (because of the small length of the uttered sentences). The adaptation material was close to 1.5 hours, given by a total of 2,479 utterances.

6. RESULTS

Figures 2, 3 and 4 show the different results for a sample of three speakers. Each figure represents one speaker;

[1]The original number of sentences was 7,500, but some of them were corrupted.

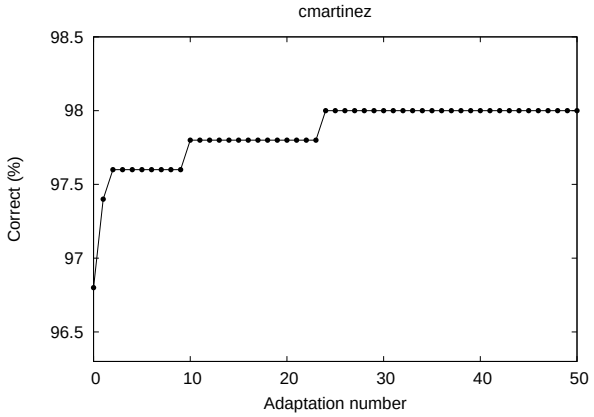


Figure 2. Sentence accuracy rate in each iterative adaptation for *cmartinez* speaker.

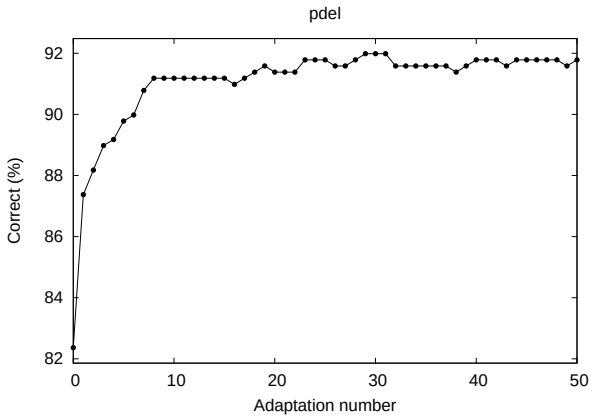


Figure 3. Sentence accuracy rate in each iterative adaptation for *pdel* speaker.

the Y-axis represents the sentence accuracy rate (SAR), and the X-axis represents the number of iterations of the MLLR algorithm (up to a total of 50 iterations). Thus, $x = 0$ means SAR with the general acoustic models; $x = 1$ means SAR with one iteration of the MLLR algorithm; $x = 2$ means SAR with two iterations of the MLLR algorithm, and so on. The *cmartinez* speaker in (Figure 2) shows how iterative adaptations progressively improved SAR results. Two more speakers (i.e., three out of ten) present a similar behaviour.

The *pdel* speaker in Figure 3 shows irregular improvement with the different adaptations (i.e., sometimes one more iteration improved the results and sometimes it made them worse). This irregular behaviour also appeared in other five speakers.

The *nalcacer* speaker in Figure 4 shows that there was no improvement with any adaptation. The best result was obtained with the initial general acoustic models without adaptation. No other speaker presented a similar evolution in the results.

Table 2. Example of prefixes, suffixes, and prefixes of suffixes randomly derived for the sentence "adición de fuentes a la lista de recursos".

Prefix	Suffix	Prefixes of the suffix
adición de	*fuentes a la lista de recursos*	*fuentes a, fuentes a la lista de recursos*
adición de fuentes	*a la lista de recursos*	*a la lista de*
adición de fuentes a	*la lista de recursos*	*la, la lista de*
adición de fuentes a la lista	*de recursos*	*de recursos*
adición de fuentes a la lista de	*recursos*	*recursos*

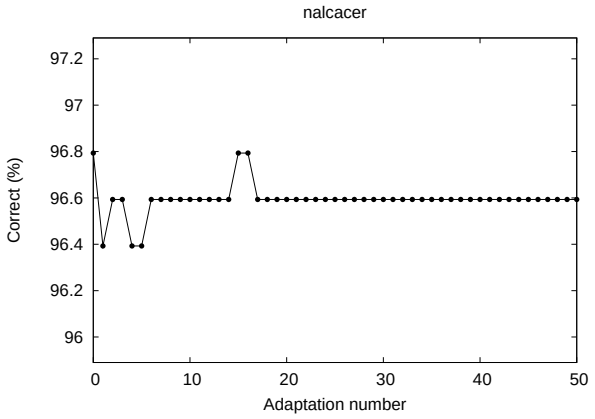


Figure 4. Sentence accuracy rate in each iterative adaptation for *nalcacer* speaker.

In Figure 5 it is shown a comparison among the before-mentioned three speaker prototypes, and the mean behaviour of the ten speakers in the study.

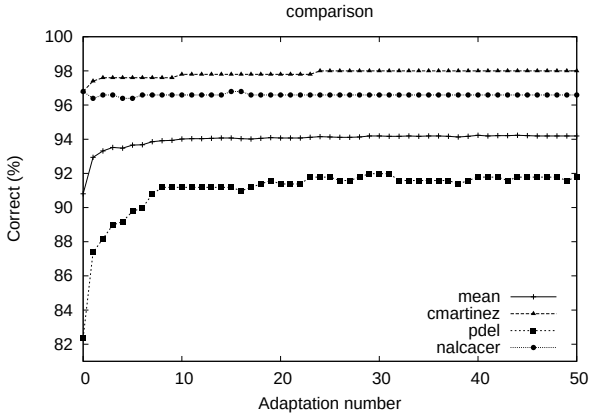


Figure 5. Iterative adaptation comparison of sentence accuracy rate for 3 prototype speakers and the mean of 10 speakers in the complet study.

The first three columns of Table 3 represent the SAR results for the ten speakers: column 1, without adaptation; column 2, with only one adaptation; and column 3, with the best adaptation. The last column represents the number of iterations of the adaptation algorithm that were used to obtain the best result from a total of 50 iterations.

Table 3. Speaker sentence accuracy rate.

Speaker	No adapt.	1 adapt.	Best adapt.	Iteration
alagarda	86.32	90.74	**93.16**	29
ecubel	91.37	94.71	**96.08**	24
cmartinez	96.80	97.40	**98.0**	24
jcivera	82.91	86.64	**90.18**	11
jandreu	94.20	95.80	**96.80**	40
pdel	82.36	87.37	**91.98**	23
evidal	89.16	90.36	**90.56**	3
lrodriguez	96.20	97.20	**98.00**	4
mnacher	91.97	**92.77**	92.77	1
nalcacer	**96.79**	96.39	None	0

Table 4. Sentence accuracy rate means.

Mean before adaptation	90.81
Mean at first adaptation	92.94
Mean with best adaptation	94.43

From these results, it seems clear that in most cases, iterative adaptation provides a significant improvement in recognition accuracy. What is not clear is the optimal number of iterations of MLLR that must be applied; most speakers need more than 20 iterations, but others get optimal results with less than 5 iterations. In only one case did the application of the adaptation make the results worse than those obtained with non-adapted models.

Table 4 shows the mean SAR results: without adaptation; with only one adaptation; and, finally, with the best adaptation for each speaker, from a set of 50 iterative adaptations. These mean results, demonstrate that, in general, iterative adaptation improves recognition accuracy.

7. CONCLUSIONS AND FUTURE WORK

The main conclusion is that, in general, several iterative adaptations seem to improve the speech recognition accuracy. However, with this technique, it is difficult to know what the optimal number of adaptations is because sometimes more adaptations can make the system results worse.

In the future, we plan to formalise this new adaptation technique mathematically. One interesting point is

to obtain an automatic and test-independent way of determining the optimal number of iterations. On the practical side, we plan to use this new technique in some industrial speech projects to make the speech recognition systems more reliable.

8. REFERENCES

[1] J. Chu-Carroll and R. Carpenter, "Vector-based natural language call-routing," *Computational Linguistics*, vol. 25, no. 3, pp. 361–388, 1997.

[2] A. L. Gorin, G. Riccardi, and J. H. Wright, "How may I help you?," *Speech Communication*, vol. 23, pp. 113–127, 1997.

[3] F. Casacuberta, C. Martínez, F. Nevado, and E. Vidal, "Implementation of an automatic voice-driven telephone exchange," *In Proceedings of the IX Spanish Symposium on Pattern Recognition and Image Analysis*, vol. 25, no. 3, pp. 307–314, May 2001, Benicàssim, Spain.

[4] Rabiner L. and Juang B., "Fundamentals of Speech Recognition," *Prentice Hall*, 1993.

[5] Jelinek F., "Statistical Methods for Speech Recognition," *MIT Press*, 1998.

[6] L. E. Baum., "An inequality and associated maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Inequalities*, , no. 3, pp. 1–8, 1972.

[7] C. J. Leggeter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density Hidden Markov Models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[8] J. L. Gauvain and C.H. Lee, "Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, April 1994.

[9] R. Kuhn, P. Nguyen, J. C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, and M.Contolini, "Eigen-voices for speaker adaptation," *In Proceedings of the International Conference on Speech and Language Processing, ICSLP*, vol. 5, pp. 1771–1774, 1998, Sidney.

[10] Carlos Martínez Hinarejos, "Seminario de Técnicas de adaptación al locutor," 2006, DSIC-UPV, Valencia, España.

[11] Heidi Christensen, "Speaker Adaptation of Hidden Markov Models using Maximum Likelihood Linear Regression," *Master Thesis*, 1996, Aalborg University.

[12] Joseph B. Kruskal and Mark Liberman, "The symmetric time-warping problem: from continuous to discrete," in *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, 1983, Massachusetts.

[13] S. J. Cox and J. S. Bridle, "Unsupervised Speaker Adaptation by Probabilistic Spectrum Fitting," *Proceedings ICASSP-89*, pp. 294–297, 1989, Glasgow.

[14] Li Lee and Richard C. Rose, "Speaker normalization using efficient frequency warping procedures," *In Proceedings of the ICASSP-96*, vol. 1, pp. 353–356, 1996, Atlanta, GA.

[15] S. M. Ahadi and P. C. Woodland, "Rapid Speaker Adaptation Using Model Prediction," *In Proceedings of the ICASSP-95*, pp. 684–687, 1995, Michigan.

[16] J.E. Díaz-Verdejo, A. M. Peinado, A. J. Rubio, E. Segarra, N. Prieto, and F. Casacuberta, "Albayzin: a task-oriented Spanish speech corpus," *In Proceedings of First Intern. Conf. on Language Resources and Evaluation (LREC-98)*, vol. 1, pp. 497–501, 1998.

[17] SchlumbergerSema S.A., Instituto Tecnológico de Informática, Rheinish Westfälische Technische Hochschule Aachen Lehrstul für Informatik VI, Recherche Appliquée en Linguistique Informatique, Laboratory University of Montreal, Celer Soluciones, Société Gamma, and Xerox Research Centre Europe., "TT2. TransType2 - Computer Assisted Translation. Project Technical Annex," *Information Society Technologies (IST) Programme*, 2001, IST-2001-32091.

SPEECH 3.0: SPOKEN DIALOG SYSTEMS OF THE THIRD GENERATION

Roberto Pieraccini

CTO, Tell-Eureka Corporation

RESUMEN

About ten years ago we witnessed the birth of the first commercial informational spoken dialog applications over the telephone, such as package tracking and flight status information. Those systems, based on proprietary platforms, implemented simple dialogs that resulted in a few user-machine turns. As the technology progressed and application builders moved into standard platforms, we saw more complex applications of the transactional type, such as stock trading and travel reservations, typically spanning a dozen or so turns.

Today, as complexity is shifting from voice processing to the application space, we are moving into the third generation of dialog systems (Speech 3.0), which are exemplified by problem solving applications. These systems are characterized by highly complex interactions that last for dozens of turns and tens of minutes with a sophisticated interplay between dialog logic, natural language speech understanding, backend functionality, advanced VUI features, and integration with live agents.

After a brief historical perspective on the evolution of the spoken dialog industry, I will describe the issues and the approaches related the development, maintenance, tuning, and business of the third generation of dialog systems. I will show how the level of complexity required by this type of applications needs sophisticated authoring, reporting, and tuning tools, new breed of speech technology, and the ability to manage and account for continuous changes in the environment. I will then conclude by delineating the future trends of spoken dialog systems, both from the research as well as from the industrial point of view.

Biografía: Roberto Pieraccini

Roberto Pieraccini graduated in electrical engineering from the "Universita' degli Studi di Pisa", Pisa, Italy, in 1980. From 1981 to 1990 he was a researcher at CSELT (Torino, Italy). In June 1990 he joined AT&T Bell Laboratories (Murray Hill, NJ) as a Member of Technical Staff, and from 1995 to 1999 he was with AT&T Shannon Laboratories (Florham Park, NJ). From November 1999 to August 2003 he was director of R&D for dialog technology at SpeechWorks International. In 2003 he joined the Human Language Technology department of IBM T.J. Watson Research in Yorktown Heights, where he managed the Conversational Interaction Technology department. In 2005 he accepted the position of Chief Technology Officer at Tell-Eureka Corporation, a company specializing in spoken dialog systems for technical support customer care. During his carrier he was actively involved in research on speech recognition, language modeling, spoken dialog systems and machine learning, and he authored more than 100 publications on these subjects.

AVIVAVOZ: TECNOLOGÍAS PARA LA TRADUCCIÓN DE VOZ

José B. Mariño Acebal

Centro de Investigación TALP
Universidad Politécnica de Cataluña
canton@gps.tsc.upc.edu

RESUMEN

AVIVAVOZ es un proyecto de tres años dirigido a la investigación avanzada en todas las tecnologías clave que intervienen en un sistema de traducción de voz (reconocimiento, traducción y síntesis de voz).
El objetivo del proyecto es lograr avances reales en todos los componentes de un sistema de traducción de voz para alcanzar sistemas de intermediación oral entre personas en las lenguas oficiales del estado español (castellano, catalán, euskera y gallego) entre sí y entre el castellano y el inglés.
El proyecto aborda el avance y la integración de las tres tecnologías implicadas. En reconocimiento de voz se desarrollará un sistema robusto, en un dominio amplio de aplicación (noticias de radiotelevisión y sesiones parlamentarias) y gran vocabulario. En traducción se avanzará en el desarrollo de técnicas estadísticas de traducción incluyendo la incorporación de distintas fuentes de conocimiento lingüístico (detección de eventos, análisis sintáctico y semántico). En síntesis de voz se generarán nuevos modelos acústicos y prosódicos para generar voz expresiva. La interacción e integración entre las tres tecnologías constituye el cuarto de los problemas abordados en este proyecto.
Las tecnologías desarrolladas en el proyecto participarán en campañas de evaluación competitivas internacionales y los avances científicos y tecnológicos se mostrarán mediante un demostrador de doblaje automático de informativos y de discursos parlamentarios.

1. INTRODUCCIÓN[1]

Puede decirse que el interés por la aplicación del computador a la traducción automática surge inmediatamente con el desarrollo de los primeros computadores tras la segunda guerra mundial. Sin embargo, tras muchos esfuerzos de investigación en EEUU la escasa capacidad de los computadores de la época no permite alcanzar resultados de interés práctico y la actividad se desvanece.

Durante los años 70 y 80 el interés en la traducción automática se desplaza a las sociedades multiculturales, como Canadá y Europa. Resultado del esfuerzo realizado son los sistemas Météo y Systran.

A partir de los años centrales de la década de los noventa, la confluencia de la rápida evolución de las prestaciones de los computadores, el desarrollo de Internet, la creciente globalización mundial con el correspondiente incremento de los intercambios interculturales, incentivó de nuevo el interés en la traducción automática. Adicionalmente, las tecnologías del reconocimiento y síntesis de voz habían alcanzado el suficiente grado de madurez para sugerir la viabilidad de la traducción de voz (speech-to-speech translation) mediante un sistema resultante de la concatenación de un reconocedor de voz (convertidor de voz a texto en la lengua origen), un traductor de texto (convertidor de texto entre las lenguas origen y destino) y sintetizador de voz (convertidor de texto a voz en la lengua destino).

El interés social y económico de las potenciales aplicaciones de la nueva tecnología constituyeron un fuerte acicate para atraer la atención de la comunidad investigadora. Entre dichas aplicaciones podemos citar:

- La interpretación simultánea (en congresos, conferencias, etc.)
- La intermediación entre personas (conferencias telefónicas, interacciones personales, etc.)
- Subtitulado de video (y TV).
- Doblaje en diferentes lenguas de audio y video (y TV).
- Interpretación en tele/video conferencias multilingües.
- Indexado automático multilingüe de material multimedia.
- Traducción de mensajes de voz.

Los proyectos C-STAR [1], Verbmobil [2] y EUTRANS [2] supusieron el punto seminal de la tecnología dedicada a la traducción de voz. En particular, el proyecto Verbmobil, financiado íntegramente por el gobierno alemán, tuvo una especial significación al permitir comparar el grado éxito en la tarea de las diversas opciones aplicadas en la traducción. Así, se atrajo la atención sobre la traducción automática estadística (TAE), que por aquel entonces comenzaba su andadura a partir del trabajo de P. Brown [4] y sus colegas en el Watson Research Center de IBM.

[1] Este trabajo ha sido subvencionado por el Gobierno Español mediante el proyecto coordinado TEC2006-13694-C03.

Los proyectos mencionados estaban dedicados a la traducción de elocuciones que se producen en tareas de contenido semántico reducido. Por ejemplo, C-STAR se dedicaba a las necesidades de un turista, Verbmobil se centraba en la concertación de una cita y EUTRANS abordaba las necesidades del huésped de un hotel. El nuevo siglo, con el viento a favor de los avances conseguidos, nos trae proyectos más ambiciosos como TC-STAR [5] y GALE [6], por ejemplo, en los que la tarea a abordar implica vocabularios no limitados y contenidos semánticos no restringidos. En particular, TC-STAR está dedicado a la traducción de las intervenciones en el parlamento europeo entre el inglés y el castellano (y viceversa), y de noticias de radiodifusión de chino a inglés.

En los últimos años, la comunidad dedicada al desarrollo de la tecnología de traducción de voz ha crecido rápidamente, lo que ha motivado el interés de la realización de campañas anuales de evaluación que permitan comparar en un marco común los diversos planteamientos seguidos. A este respecto, cabe destacar las evaluaciones organizadas por el consorcio C-STAR y cuyos resultados son motivo de estudio en el International Workshop on Spoken Language Translation [7], y las campañas realizadas por el National Institute of Standards and Technology (NIST) norteamericano [8].

El proyecto AVIVAVOZ [9] se plantea situar las lenguas oficiales del estado español al nivel tecnológico del estado del arte. En particular, toma como referencia los avances alcanzados en TC-STAR.

2. LAS TECNOLIGÍAS IMPLICADAS

AVIVAVOZ es un proyecto dedicado a la investigación avanzada de las tecnologías centrales para la traducción automática del habla (TAH): reconocimiento de voz, traducción estadística y síntesis de voz.

2.1. Reconocimiento de voz

Es indudable que disponer de un sistema de reconocimiento de habla (RAH) preciso y competente es uno de los puntos clave para abordar un sistema de traducción automática voz a voz. Los sistemas de reconocimiento actuales son capaces de lograr tasas de reconocimiento extremadamente altas en aplicaciones complejas (habla continua y grandes vocabularios), pero sólo con condiciones controladas, es decir, cuando la grabación es limpia y contiene voz previamente segmentada, y además tanto el locutor, como el vocabulario y el tipo de lenguaje son conocidos y están bien modelados. Cuando una o varias de estas condiciones no se cumplen, los reconocedores de voz reducen sus prestaciones de forma importante. Este será, en general, el caso de los sistemas de traducción automática. Por un lado, y dependiendo de la aplicación, cabe esperar unas condiciones de grabación distantes de las ideales, pero además el tipo de habla, aunque no totalmente espontánea, estará bastante alejada del habla planeada o del dictado, tanto en cuanto al cuidado en la pronunciación como en cuanto al vocabulario y al lenguaje.

La aplicación que se plantea es, por tanto, de una dificultad similar a las tareas más complejas que se abordan actualmente en reconocimiento de habla: transcripción automática de programas de noticias, transcripción de conversaciones telefónicas y transcripción de reuniones de trabajo. Aunque los avances en los últimos años han sido importantes, los sistemas de reconocimiento más avanzados como el Byblos BBN en EEUU, el sistema del laboratorio LIMSI-CNRS en Francia, o el CU-HTK de la universidad de Cambridge en Reino Unido, aún no consiguen prestaciones totalmente satisfactorias en estas tareas. En transcripción de programas de noticias, el mejor resultado obtenido en la actualidad está en torno al 10% de WER (“word error rate”), mientras que para transcripción de conversaciones telefónicas la tasa de error se ve incrementada hasta el 15-20%. Estos resultados se logran con cantidades ingentes de material de entrenamiento y en tiempos de ejecución varias veces por encima de tiempo real, lo que dificulta su aplicación en problemas reales y en idiomas minoritarios.

Para lograr estas prestaciones, los problemas que tienen que resolverse (aparte de la recogida de datos) tienen que ver con el tratamiento adecuado de la alta variabilidad acústica, de lenguaje y vocabulario presentes en las interacciones entre humanos. En cuanto al modelado acústico, los problemas aparecen cuando existe desajuste entre el material de reconocimiento y el de entrenamiento. Este desajuste puede estar producido por variación del ruido ambiental o por la variabilidad interlocutor. Respecto al ruido ambiental, los reconocedores incluyen parametrizaciones que incorporan parte de las características del oído humano en cuanto a robustez frente al ruido (parametrización MFCC o PLP). Otros autores proponen la utilización de métodos de combinación de parámetros aplicando algoritmos de fusión de datos. Respecto a la variabilidad interlocutor, la normalización de la longitud del tracto vocal (VTLN) y el entrenamiento adaptativo de locutor son los sistemas más utilizados para evitar su influencia.

En cuanto al modelado lingüístico, las técnicas más empleadas para RAH de grandes vocabularios son los modelos estadísticos tipo N-grama. Existen herramientas de dominio público que permiten entrenar dichos modelos; una de las más conocidas es “SRILM - The SRI Language Modeling Toolkit” de la firma SRI. A partir de esta herramienta se pueden explorar técnicas de selección de material textual apropiado a la tarea a reconocer. Estamos hablando entonces de reducir el desajuste entre entrenamiento y reconocimiento mediante la combinación de múltiples modelos de lenguaje cada uno de ellos especializado en un “tema” diferente. Esta vía de investigación es muy interesante pues va en la línea general del proyecto de

incorporación temprana de las fuentes de conocimiento en contraposición del uso ingente de datos (voz y texto). Redundando en lo anterior, una de las estrategias más frecuentemente adoptadas para hacer viable la etapa de decodificación es el empleo de un esquema multipase. En el primer pase de reconocimiento se emplean modelos acústico y de lenguaje muy generales. La salida de este primer pase se emplea para seleccionar y refinar los nuevos modelos a usar en el segundo pase. Este procedimiento se puede realizar varias veces de forma sucesiva, utilizando en cada pase nuevas o adaptadas fuentes de conocimiento.

Debido a la naturaleza de la aplicación que nos ocupa, es de esperar que el tipo de habla al que deba enfrentarse el reconocedor sea en gran parte de tipo espontáneo. El reconocimiento de habla espontánea presenta una problemática específica y compleja, que conlleva unas prestaciones de los reconocedores mucho más reducidas que en aplicaciones de dictado. La razón principal es que el habla espontánea incluye multitud de "palabras relleno", vacilaciones, repeticiones, palabras cortadas, frases sin terminar y otro tipo de disfluencias. Estos fenómenos son muy difíciles de modelar tanto desde el punto de vista léxico, como lingüístico. Los modelos de lenguaje que se entrenan a partir de textos escritos fallan de forma evidente al ser aplicados a este tipo de tareas, mientras que la construcción de modelos de lenguaje específicos resulta difícil debido a la escasez de material de la que se dispone. La incorporación de información lingüística de mayor alcance que la que actualmente se hace servir es una opción a considerar. Esto aplica tanto al modelado de lenguaje (donde sólo se está usando n-gramas) como al modelado prosódico (donde características a largo plazo deberían ser más beneficiosas que las actuales a corto plazo, aunque ello implique un aumento en la complejidad del algoritmo de búsqueda).

La traducción debe enfrentarse con los errores de reconocimiento. Medidas de confianza adjuntas a las palabras en una lista con las N-mejores hipótesis (o, alternativamente, un grafo) pueden ser una aportación para paliar los efectos de un reconocimiento defectuoso. Por ello, en este proyecto se continuarán los trabajos para el desarrollo de nuevas medidas de confianza basadas en test de hipótesis Bayesiano que se incluirán en las nuevas versiones del motor de reconocimiento. Además, la traducción requiere información adicional a la mera transcripción ortográfica del mensaje oral. Por ejemplo, se necesita información sobre puntuación que puede ser capturada mediante modelos de lenguaje adecuados y/o algoritmos mejorados de segmentación acústica. Ambos problemas serán considerados en este proyecto.

En la literatura dedicada al reconocimiento de voz, la detección de "metadatos" está adquiriendo relevancia. Hasta ahora, los sistemas de reconocimiento proporcionaban solamente transcripción de la voz. Cuando el mensaje debe ser reproducido en otra lengua con una voz similar a la original, se hace necesaria información adicional, como la identidad del hablante o la indicación de cambio de hablante. Esta información debe ser proporcionada al sintetizador de voz. Entre otros parámetros de interés, podemos mencionar: velocidad de articulación, pausas y otras características prosódicas (disfluencias, "palabras relleno", etc.). En este proyecto, se incluyen tareas dedicadas a la extracción de estos "metadatos" de la señal de voz y su utilización en los módulos de traducción y síntesis.

A los problemas anteriores, se añade el interés en el desarrollo de sistemas de reconocimiento multilingües sobre los que nos centraremos en dos aspectos: portabilidad de tecnología entre idiomas y funcionamiento en condiciones de bilingüismo.

2.2. Traducción estadística

En tan solo diez años, los métodos de traducción automática estadística (TAE) han alcanzado prestaciones similares a las de los métodos basados en conocimiento, los cuales han estado evolucionando por más de medio siglo. Sin embargo, a pesar de los progresos actuales, la tecnología de la TA está todavía muy lejos de alcanzar niveles de calidad y prestaciones satisfactorios.

Entre las principales limitaciones que exhiben los sistemas de TA actuales, las más importantes son su limitación a trabajar en dominios restringidos, su incapacidad para manejar vocabularios muy grandes, y sus aún relativamente altas tasas de error. Aunque los expertos opinan que un verdadero avance en la tecnología de la TA sólo será posible mediante la combinación de la fuerza bruta (métodos estadísticos) con fuentes de información lingüística (métodos basados en conocimiento), muchos de los esfuerzos recientes en esta dirección han fracasado en proporcionar mejoras significativas respecto a los sistemas de TA existentes. Por esta razón, hoy por hoy, la TA continúa siendo uno de los dolores de cabeza más interesantes de la Inteligencia Artificial (IA), y un gran reto para académicos e investigadores.

Desde el punto de vista de su aplicación práctica, la TA se puede dividir en dos problemas específicos: la traducción automática del lenguaje escrito (TALE) y la traducción automática del lenguaje hablado (TAH). El problema de la traducción del habla (TAH) presenta, en adición a las complejidades específicas de la TA, dos tipos de problemas adicionales: en primer lugar, aquellos relacionados con la naturaleza del lenguaje oral, como por ejemplo la espontaneidad y la mayor libertad de las estructuras sintácticas; y en segundo lugar, aquellos relacionados con el estado actual del arte en el reconocimiento automático del habla (RAH), como por ejemplo los errores de reconocimiento. Por estas razones, se ha prestado especial atención al problema de la integración de las tecnologías de RAH y TA en los últimos años.

En el caso de la TALE, a diferencia de la TAH, el lenguaje escrito es mucho más controlado en lo relativo

a contenido gramatical de las oraciones. Esto permite explotar una mayor cantidad de información lingüística partiendo desde el nivel léxico hasta los niveles sintáctico y semántico. Recientemente, algunos autores han presentado propuestas para usar información sintáctica para construir modelos de traducción, reportando resultados prometedores en la mejora de la calidad lingüística de las traducciones.

Este proyecto considera el problema de TA en su variante TAH y se concentra en tareas de traducción entre las cuatro lenguas del estado español, así como entre el castellano y el inglés; en función de lo que permitan los recursos bilingües disponibles. Aunque el problema de TA será principalmente abordado desde el paradigma estadístico, también se dedicará un esfuerzo importante a la incorporación, uso y adaptación de las herramientas y recursos lingüísticos disponibles. Así mismo, se prestará especial atención a la integración de los sistemas de RAH y TA mediante el uso de modelos de lenguaje y traducción basados en n-gramas. Adicionalmente, se trabajará en el desarrollo de algoritmos novedosos y métodos para la inclusión de información lingüística con el objeto de mejorar el estado actual del arte en TAE tanto desde el punto de vista de la eficiencia computacional como de las prestaciones en términos de calidad de traducción, tamaño del vocabulario y restricciones de dominio.

2.3. Síntesis de voz

En la última década se ha producido una significativa mejora en la calidad de la voz sintética. Sin embargo, esta tecnología está aún restringida a aplicaciones donde es aceptable una limitada calidad de la voz, o donde el dominio de aplicación es muy específico. La síntesis de voz se utiliza actualmente cuando las comunicaciones entre persona y máquina serían muy difíciles o incluso imposibles sin la ayuda de estos sistemas. El objetivo final de estos sistemas es obtener voces naturales capaces de expresar cualquier estilo, humor, acento u otra característica de los hablantes humanos. Un escenario en el que se utilizan estas aplicaciones es la traducción voz-voz, donde la voz sintética traducida podría reproducir las características del hablante original. Otro aspecto importante a considerar es la lengua utilizada en el proceso de comunicación. Cada vez es más necesario el acceso multilingüe a distintos dispositivos y sistemas. Esto es especialmente relevante en áreas donde hay más de una lengua oficial. Para conseguir estos objetivos, en este proyecto se desarrollarán nuevas técnicas para la generación de prosodia y la producción de voz, en un entorno multilingüe.

Las técnicas de síntesis por concatenación de unidades han permitido la mejora de la calidad acústica de la señal sintética. En estas técnicas, segmentos de señal extraídos de grabaciones previas realizadas con un locutor, se concatenan para producir la señal de salida deseada. La prosodia de la señal de salida debe ser calculada de acuerdo a los modelos prosódicos adecuados. Para generar una transición natural entre los segmentos de voz y conseguir la prosodia deseada, es necesario manipularlos adecuadamente. Para manipular la señal de voz con alta calidad, se han utilizado distintos algoritmos como PSOLA, MBROLA, HNM y modelos sinusoidales. Con estas técnicas, la calidad de la señal sintética de salida disminuye drásticamente si los segmentos manipulados tienen unas características muy alejadas de las de los originales. Sin embargo, son muy apropiadas en sistemas con poca memoria y baja capacidad de cálculo, como los dispositivos portátiles. Obtener mejores modelos de la producción de la voz mejoraría la calidad final de todos estos sistemas.

La solución actual al problema de la distorsión debida a la manipulación de las unidades de voz es grabar tantos segmentos de voz como sea posible, con la esperanza de que a la hora de la síntesis la manipulación se reduzca al mínimo. Para ello se diseña un corpus que contenga todos los sonidos en todos los contextos fonéticos y con todos los contornos prosódicos que puedan ser necesarios en el momento de la síntesis. En muchos casos, no es necesario manipular significativamente la voz original ya que los segmentos seleccionados tienen unas características prosódicas muy parecidas a las buscadas para la señal sintética, obteniéndose una calidad de la voz sintética excelente. Por otro lado, si es necesaria una manipulación significativa, la voz sintética resultante suena muy poco natural y puede llegar a ser incomprensible. En el diseño de un sistema de síntesis basada en corpus hay tres problemas principales. El primero es el diseño del corpus de grabación: cuanto más grande sea, mayor cobertura tendrán las unidades grabadas de modo que se reduce la probabilidad de introducir distorsión por manipulación de las unidades. Sin embargo, muchos aspectos del lenguaje están caracterizados por un gran número de fenómenos raros, lo que implica que es necesaria una base de datos enorme. Se han desarrollado técnicas para obtener bases de datos óptimas, con resultados muy prometedores, pero es necesario profundizar más. El segundo problema es la definición del coste de concatenación de las unidades. En el momento de la concatenación hay que seleccionar las mejores unidades, lo que implica definir una distancia entre las unidades seleccionadas y las unidades objetivo y una distancia entre las características de las unidades. Para cada una de las características utilizadas es necesario definir distancias perceptuales. Finalmente, encontrar la mejor combinación de unidades requiere una búsqueda por toda la base de datos, lo cual implica una gran capacidad de cálculo. Como consecuencia, es necesario utilizar algoritmos de búsqueda rápida.

De los párrafos anteriores se deduce que obtener una nueva voz con alta calidad es una tarea que tiene un coste muy elevado que requiere la selección del locutor y la grabación del corpus. La transformación de la voz del locutor mediante manipulación de sus características es una técnica muy interesante que ya está ofreciendo

resultados prometedores. En las aplicaciones de traducción voz-voz, la reproducción de las características del hablante original (calidad de la voz, estilo de habla, emoción, etc.) es un interesante objetivo.

Uno de los mayores retos que existen en conversión texto a voz es dotar a la voz sintética de la suficiente naturalidad, que se incrementaría al poder expresar emociones. Para producir habla sintética emocional es preciso contar con una buena base de datos de voz emocional que permita el estudio y la caracterización acústica de las emociones. Antes de ser utilizadas, estas bases de datos deben ser evaluadas, ya que la emoción emitida y la percibida no siempre coinciden. La síntesis del habla emocional se realiza principalmente utilizando técnicas basadas en corpus o por regla, aunque hay otras aproximaciones, como usar una representación fonológica intermedia o realizar *morphing* prosódico. Para poder considerar las diferentes emociones en el texto de entrada al sistema, hay que contemplar una interfaz entre la aplicación y el sistema de conversión de texto a voz. El lenguaje de marcado desarrolla esta interface. Actualmente se están definiendo los lenguajes de marcado para cumplir esta función, todos ellos basados en el estándar XML. Entre los lenguajes con aplicación a la síntesis de voz emocional se incluyen el lenguaje de presentación multimodal (MPML, *Multimodal Presentation Markup Language*) y la especificación del lenguaje de marcado para humanos virtuales (VHML, *Virtual Human Markup Language*). Esta última incluye un sublenguaje específico para el marcado de emociones (EML *Emotion Markup Language*).

3. EL CONSORCIO INVESTIGADOR

El proyecto AVIVAVOZ ha sido propuesto por un consorcio formado por tres equipos investigadores. Los grupos participantes son el grupo de Procesado de Voz del "Centro de Investigación de Tecnologías y Aplicaciones de la Tecnología del Lenguaje y el Habla" (TALP) de la Universidad Politécnica de Cataluña, el Grupo de Teoría de la Señal (GTS) del Departamento TSC de la Universidad de Vigo y el Grupo de Investigación Aholab (AHOLab) de la Universidad del País Vasco. Los tres grupos tienen experiencia previa en participación y colaboración conjunta en proyectos coordinados financiados por CICYT y por empresas y en la actualidad participan en una red de excelencia internacional [ECESS] de síntesis de voz.

TALP [10] aporta experiencia en los tres campos: reconocimiento de voz con sistemas competitivos en aplicaciones telefónicas y un sistema de reconocimiento de habla continua, ambos en catalán, castellano e inglés, traducción de habla en catalán, castellano e inglés y síntesis de voz en castellano y catalán. TALP coordinará las actividades de traducción e integración de tecnologías.

GTS [11] aporta experiencia en reconocimiento de habla continua en una tarea compleja como es noticias radiofónicas en gallego y castellano, que complementa perfectamente el sistema de TALP y una sólida experiencia en síntesis de voz en ambos idiomas. GTS coordinará las tareas de reconocimiento de voz.

Finalmente, AHOLab [12] es un grupo experto en síntesis de voz y en particular en prosodia. Su sistema de síntesis trabaja en euskera y castellano. También ha desarrollado un sistema de reconocimiento en euskera. AHOLab coordinará las actividades de síntesis de voz.

4. OBJETIVOS DEL PROYECTO

El objetivo del proyecto es lograr avances reales en todos los componentes de un sistema de traducción de voz para alcanzar sistemas de intermediación oral entre personas en las lenguas oficiales del estado español (castellano, catalán, euskera y gallego) entre sí y entre el castellano y el inglés. En reconocimiento de voz, se desarrollará un sistema para grandes vocabularios y dominios de aplicación amplios (noticias de radiodifusión y discursos parlamentarios). Para la traducción de habla, se construirá un sistema estadístico basado en corpus al que se incorporará conocimiento lingüístico (detección de eventos, análisis sintáctico, etc.). Para la síntesis de voz, se investigarán modelos acústicos y prosódicos nuevos para la generación de voz expresiva. Por último se abordará la integración de las tecnologías anteriores y se desarrollará un demostrador.

4.1. Reconocimiento de voz

El objetivo general es aumentar las prestaciones de los sistemas de reconocimiento automático de habla de los distintos grupos de investigación de este proyecto, y lograr su integración en un sistema de traducción de voz-a-voz. Desgranándolo según actividades:

- Desarrollo de una nueva arquitectura de reconocimiento flexible, con múltiples capas, en la que la decodificación acústico-fonémica se realice de forma independiente de la decodificación de palabras.
- Desarrollo de un módulo de segmentación de audio en trozos lingüísticamente homogéneos.
- Desarrollo de nuevas medidas de confianza basadas en la información proporcionada por la arquitectura anterior.
- Mejora del modelado acústico mediante la aplicación de nuevas técnicas de adaptación al locutor y al canal.
- Mejora del modelado acústico en entornos multilingües para reducir el conjunto total de modelos acústicos.
- Desarrollo de modelos de lenguaje mediante el empleo de algoritmos de adaptación y aprendizaje.
- Tratamiento específico del habla espontánea
- Evaluación del sistema de reconocimiento de grandes vocabularios y habla continua.

4.2. Traducción estadística

El objetivo principal es desarrollar sistemas de traducción automática que incorporen información morfo-sintáctica y semántica. Este objetivo principal se puede subdividir en los siguientes objetivos específicos:

- Preprocesado y edición final de colecciones de datos bilingües de amplio dominio y anotados con información lingüística relevante, para ser usados en el entrenamiento de los sistemas de traducción automática estadística.
- Desarrollo de un sistema base para la traducción automática del habla y su integración eficiente con el sistema de reconocimiento automático.
- Desarrollo de algoritmos y estrategias de búsqueda eficientes que permitan la implementación en tiempo real de los sistemas de traducción automática estadística.
- Desarrollo de sistemas de traducción específicos para las cuatro lenguas oficiales en España.
- Demostración de las prestaciones de los sistemas desarrollados mediante la realización de tareas específicas de traducción y la comparación con otros sistemas de traducción automática.

4.3. Síntesis de voz

El objetivo principal del proyecto en el campo de la síntesis de voz es la obtención de sistemas capaces de generar voces con un alto grado de naturalidad en un entorno multilingüe (inglés, castellano, catalán, euskera y gallego). Estos sistemas deben expresar diferentes estilos de habla, acentos y en general parámetros de calidad de la voz especificados en un texto de entrada. Para lograr este objetivo, se desarrollarán los siguientes objetivos parciales:

- Interpretación adecuada del texto de entrada. Una interpretación incorrecta causada por una entrada anómala (tal como ausencia o incorrecta situación de signos de puntuación, etc.) pueden generar una salida ininteligible. De la misma forma, la transcripción de grafema a alófono debe ser precisa si se desea obtener el mensaje acústico adecuado a la entrada. En este proyecto se investigarán algoritmos de procesado lingüístico y técnicas basadas en datos que mejoren los sistemas actuales.
- Generación de voz expresiva en diferentes estilos de habla, voces e idiomas. Para obtener voz sintética expresiva con un alto grado de calidad es necesario elaborar nuevos modelos prosódicos. En este sentido se investigarán técnicas para inferir la prosodia del texto de entrada, y transferida correctamente a las unidades de síntesis, así como su relación con los parámetros acústicos relevantes.
- Algoritmos de generación manipulación de voz. Se necesitan nuevos modelos de producción de voz que permitan realizar modificaciones de las características prosódicas y espectrales de la señal sin introducir una degradación significativa. Estos modelos pueden ser utilizados para modificar las características de la voz así como para caracterizar al locutor, como es el caso de una tarea de transformación de voz. Los mismos modelos pueden utilizarse también para comprimir la señal de voz y reducir la cantidad de memoria utilizada.
- Estándares, evaluación y sistemas de referencia. Con objeto de evaluar la validez de las nuevas técnicas desarrolladas es necesario definir sistemas de referencia. Asimismo, para permitir el intercambio de módulos y la evaluación comparativa del comportamiento individual de cada módulo, se propone definir interfaces estándares de conexión de módulos.

4.4. Integración de tecnologías

En esta actividad se integran los sistemas de reconocimiento, traducción y síntesis a fin de construir el sistema completo de traducción del habla. El logro de este objetivo general implica el cumplimiento de los sub-objetivos siguientes:

- Desarrollo de recursos bilingües para los pares Castellano – Inglés/Catalán/Euskera/Gallego, incluyendo textos paralelos y herramientas de procesado de lenguaje.
- Realización de una arquitectura web para la comunicación entre los diversos subsistemas.
- Definición y realización de interfaces eficientes entre los distintos subsistemas que permitan mejorar las prestaciones del sistema completo.
- Realización de herramientas para el doblaje automático de video como demostrador de las tecnologías desarrolladas.

5. AGRADECIMIENTOS

Este trabajo se basa en la memoria técnica del proyecto AVIVAVOZ. En consecuencia, debe reconocerse (y agradecerse) a todos los miembros del equipo investigador su contribución.

6. REFERENCIAS

[1] http://www.c-star.org/
[2] http://verbmovil.dfki.de/verbmobil/overview-us.html
[3] http://www.cordis.lu/espirit/src/30268.htm
[4] P. F. Brown et al., “The mathematics of statistical machine translation: parameter estimation”, Computational Linguistics, 19 (2), 1993.
[5] http://www.tc-star.org
[6] http://www.darpa.mil/ipto/programs/gale/index.htm
[7] http://www.slt.atr.jp/IWSLT2006/
[8] http://www.nist.gov/speech/tests/mt/
[9] http://gps-tsc.upc.es/veu/avivavoz
[10] http://gps-tsc.upc.es/veu/
[11] http://www.gts.tsc.uvigo.es/web/index.php
[12] https://bips.bi.ehu.es/aholab/

EDECÁN: sistEma de Diálogo multidominio con adaptación al contExto aCústico y de AplicacióN

Eduardo Lleida[1], Encarna Segarra[2], María Inés Torres[3], Javier Macias[4]

[1] Universidad de Zaragoza (uz@edecan.es)
[2]Universitad Politécnica de Valencia(upv@edecan.es)
[3]Universidad del País Vasco (ehu@edecan.es)
[4]Universidad Politécnica de Madrid (upm@edecan.es)

RESUMEN

El proyecto EDECÁN[1] tiene como objetivo aumentar la robustez de un sistema de diálogo de habla espontánea a través del desarrollo de tecnologías para la adaptación y personalización del mismo a los distintos contextos acústicos y de aplicación en los que pueda encontrarse. El concepto de contexto acústico engloba todos los elementos que influyen, en mayor o menor medida, en la señal que capta/n el/los micrófono/s que conforma/n la entrada del sistema de diálogo. Estos elementos dependen tanto del usuario como del entorno físico que lo rodea. Por otro lado, el contexto de aplicación hace referencia a la estructura semántica de los dominios en los que se desarrolla el diálogo.

Los objetivos de EDECÁN, implican la necesidad de desarrollar estrategias que permitan caracterizar las condiciones de funcionamiento del sistema de diálogo (condiciones acústicas, tipo de habla utilizada, contexto semántico, tipo de usuario ...) y definir e implementar técnicas de adaptación a tales condiciones. La incorporación de técnicas de adaptación y personalización dinamizan el funcionamiento de un sistema de diálogo, lo que obliga a adaptar a esta nueva situación las estrategias de evaluación y medida de usabilidad conocidas. Por otro lado, este proyecto aborda la extensión del sistema de diálogo a nuevos contextos de aplicación, posibilitando al usuario la consecución de múltiples objetivos en el transcurso del diálogo.

EDECÁN es un proyecto coordinado en el que participan investigadores de la Universidad Politécnica de Valencia, Universidad del País Vasco, Universidad Politécnica de Madrid y Universidad de Zaragoza.

1. INTRODUCCIÓN

Los avances recientes en las Tecnologías de la Información y las Comunicaciones, han tenido un gran impacto en el modo en que vivimos, trabajamos e interactuamos con nuestro entorno personal y profesional. Estas tecnologías están permitiendo desarrollar redes distribuidas de sistemas que proporcionan información, comunicación y entretenimiento. En este contexto, una visión futurista de la Sociedad de la Información enfatiza el desarrollo de entornos en los que las personas interactúan de forma transparente con multitud de dispositivos interconectados para desarrollar actividades de la vida diaria. Este planteamiento se materializa, a grandes rasgos, en un individuo rodeado de interfaces inteligentes e intuitivas que se encuentran integradas en elementos de la vida cotidiana, todo esto en un entorno que sea capaz de reconocer y responder a la presencia y necesidades de diferentes individuos, de una forma personalizada, discreta e imperceptible salvo a través de los resultados. El entorno mencionado, bautizado como "ambiente inteligente", está donde el individuo se encuentre y responde a sus necesidades de una forma natural, no limitándose a un único lugar físico sino que comprende a todos ellos, la casa, el coche, el lugar de trabajo, etc. Uno de los retos más apasionantes que pone sobre la mesa esta visión es el desarrollo y la investigación en sistemas de diálogo robustos como interfaz entre las personas y el "ambiente". Un valor añadido de esta investigación es su contribución a la incorporación a la Sociedad de la Información, de la población no habituada al manejo de las nuevas tecnologías al desarrollar una interfaz "intuitiva" entre las personas y los recursos que la Sociedad de la Información pone a su disposición.

Sin embargo, hoy en día, los sistemas de diálogo son más o menos aceptados y útiles cuando trabajan en entornos semánticos muy restringidos, con condiciones acústicas favorables y con usuarios colaboradores. Los portales de voz son un ejemplo ya que éstos están preparados para unas condiciones acústicas particulares (canal telefónico fijo y/o móvil) y aplicaciones muy concretas. Por todo ello, es necesario todavía dedicar un gran esfuerzo de investigación para dotar, a todos los componentes de un sistema de diálogo, de robustez y capacidad de adaptación frente a los diversos contextos y necesidades propias de cada usuario en los que puede encontrarse el sistema.

En un sistema de diálogo se encuentran sistemas de distinta complejidad que deben reconocer las palabras pronunciadas, comprender su significado, gestionar el diálogo (incluyendo la información contextual, manejo de errores y acceso a la aplicación final) y generar finalmente la respuesta oral. Además de estas funciones básicas, el sistema puede incorporar nuevas funcionalidades como el modelado del usuario, la adaptación al entorno, o la inclusión de otras formas de entrada/salida como texto, lectura de labios, gráficos o sistemas táctiles, conformando así un sistema multimodal que robustece, facilita y mejora la interacción de los usuarios, incluso con algún tipo de minusvalía.

[1] Este proyecto ha sido subvencionado por el Gobierno Español mediante el proyecto coordinado TIN2005-08660-C04.
EDECÁN (Del fr. aide de camp). Ayudante de campo. Auxiliar, acompañante.(Diccionario 2001 Real Academia Española)

Desde el punto de vista de la complejidad del diálogo, Allen et al. [1] definen 5 niveles de complejidad según la técnica utilizada para representar los actos de diálogo, tal y como se muestra en la figura 1.

Técnica Empleada	Ejemplo de Tarea	Complejidad de la Tarea	Fenómenos del Diálogo que Gestiona
Diagramas de Estados Finitos	Marcación telefónica	Menos Compleja	Respuesta a preguntas del usuario
Basado en Frames	Información de llegadas y Salidas de Trenes		Respuesta a preguntas del usuario, clarificaciones simples del sistema
Conjunto de Contexto	Agente de Viajes	Más Compleja	Conmutación entre dominios semánticos predeterminados
Modelos basados en planes	Consultor de diseño de cocinas		Estructuras de dominios semánticos generadas dinámicamente, subdiálogos de negociación colaborativa
Modelos basados en agentes	Gestión de apoyo a catástrofes.		Diferentes modalidades.

Figura 1.Niveles de complejidad en un sistema de diálogo [1]

Bajo este contexto, el proyecto EDECÁN propone como objetivo aumentar la robustez del sistema de diálogo a través del desarrollo de tecnologías para la adaptación y personalización del mismo a los distintos contextos acústicos y de aplicación en los que pueda encontrarse. Este objetivo implica la necesidad de desarrollar estrategias que permitan caracterizar las condiciones de funcionamiento del sistema de diálogo (condiciones acústicas, tipo de habla, dominio de aplicación, tipo de usuario, ...), definir e implementar técnicas de adaptación a tales condiciones, extender el sistema de diálogo a nuevos dominios de aplicación, y estudiar la evaluación y usabilidad de los sistemas de diálogo.

Desde el punto de vista del diálogo, EDECÁN es una evolución de dos proyectos anteriores del equipo investigador: DIHANA (TIC2002-04103-C03) y SAITE (FEDER 2FD1997-1062-C02-01). El proyecto DIHANA tuvo como objetivo la profundización en aspectos metodológicos fundamentales sobre la interfaz de audio y modelado acústico, tratamiento del habla espontánea, modelado del lenguaje, comprensión y diálogo, mientras que SAITE se centró en el estudio de estrategias avanzadas de diálogo multiobjetivo. En ambos casos la aplicación de diálogo estaba restringida a un servicio de información de precios y horarios de tren. Dentro de la taxonomía definida en la figura 1, DIHANA y SAITE, como muchos de los sistemas de diálogo construidos hoy en día [1], se pueden clasificar como sistemas de baja complejidad basados en “frames” con iniciativa mixta.

Sin embargo, EDECÁN supone un paso más en dicha taxonomía al aumentar la complejidad a sistemas basados en múltiples dominios, cada dominio representado por una serie de contextos, cada uno de ellos basado a su vez en “frames”. Este aumento en la complejidad del sistema de diálogo conlleva nuevos retos como son la identificación de los cambios de contexto y dominio que el usuario puede realizar (por ejemplo, cuando el usuario vuelve a un contexto anterior y quiere cambiar algún detalle). Además, EDECÁN propone la personalización, a un usuario particular, del sistema de diálogo tanto a nivel del contexto acústico como de aplicación. La personalización conlleva la identificación de usuario (o tipo de usuario) y del entorno acústico, la definición de perfiles de usuario, etc. Por todo ello, EDECÁN tiene también como objetivo evolucionar el diseño de la arquitectura del sistema de diálogo DIHANA para facilitar la adopción e incorporación de las nuevas tecnologías que se desarrollen.

2. LOS SISTEMAS DE DIÁLOGO HABLADO: NUEVOS RETOS RELACIONADOS CON LOS CONTEXTOS ACÚSTICO Y DE APLICACIÓN

Los objetivos en el desarrollo de sistemas de interacción hombre-máquina han evolucionado convirtiéndose en más realistas (a la vez que más ambiciosos) conforme se han desarrollado las tecnologías necesarias. Sin embargo, aunque bajo ciertas condiciones, el modelado de cada una de las áreas que participan en los procesos de interacción oral pueden proporcionar buenos resultados (modelos acústicos, lingüísticos, o de diálogo), tanto el problema de la implantación de estos sistemas en ambientes reales como la adaptación a nuevas tareas son un reto que va a exigir importantes esfuerzos: reconocimiento del habla espontánea con usuarios no “entrenados” y en ambientes ruidosos, modelado del lenguaje y semántico con una amplia cobertura, o la gestión del diálogo dando máxima libertad al usuario.

En los últimos años ha habido importantes avances en el desarrollo de sistemas de diálogo hablado, tanto en la línea de servicios de información (donde la interacción con la máquina requiere una cierta complejidad para conseguir la información deseada), como en servicios donde la interacción es más sencilla pero las condiciones acústicas dificulten mucho la tarea.

2.1. Sistemas de diálogo y contexto de aplicación

Por contexto de aplicación nos referimos a la estructura semántica de los dominios en los que se desarrolla el diálogo. Esta estructura incluye: los conceptos semánticos y las relaciones entre ellos (ontología), el conjunto de reglas (diseñadas o aprendidas) que relacionan conceptos con expresiones del lenguaje natural, y por último, un modelo para la gestión eficiente de la interacción persona-máquina. Una de las características más importante de un sistema de diálogo es que tenga capacidad de adaptarse a nuevas tareas, a diferentes tipos de usuario o de idioma, o a cambios en las condiciones del contexto acústico. Por ello se han dedicado muchos esfuerzos al diseño de arquitecturas que permitan fácilmente la adaptación a nuevos dominios. Tal es el caso de la arquitectura GALAXY [2] del MIT bajo la cual se han desarrollado los sistemas JUPITER (información meteorológica), VOYAGER, (información urbana), u ORION (asistente personal) y que ha servido de plataforma para el proyecto DARPA COMMUNICATOR, http://communicator.sourceforge.net . A nivel nacional, los proyectos BASURDE (TI98-0423-C06) y DIHANA han definido una arquitectura, similar a GALAXY, que permite la integración de nuevos componentes al sistema de diálogo.

Aparte del gestor de diálogo (módulo fundamental en cualquier sistema de diálogo) el módulo de comprensión de lenguaje natural juega también un papel primordial. Aunque tradicionalmente la construcción de este módulo se ha basado en la definición manual de reglas semánticas para la detección

de palabras clave con las que rellenar un marco (frame), también se han desarrollado otras aproximaciones basadas en el uso de modelos estocásticos: el BNN-HUM [3], el AT&T-CHRONUS [4] y el LIMSI-ARISE [5] son algunos ejemplos del uso de modelos ocultos de Markov y N-gramas para modelar estocásticamente el proceso de comprensión. Hay también otras aproximaciones estadísticas basadas en clasificación, traducción, y técnicas de inferencia gramatical. Estas aproximaciones estocásticas entienden el proceso de comprensión como un problema de traducción de la frase de entrada a una determinada representación semántica del significado de dicha frase. Una de las principales ventajas de la aproximación estocástica es que permite aplicar técnicas de aprendizaje automático a partir de datos reales modelando las posibles fuentes de error.

La existencia de varios contextos de aplicación implica un diseño del sistema mucho más complejo, en el que se debe identificar el contexto apropiado para dar respuesta al usuario en cada momento, dando la posibilidad de cambiar de contexto a medida que se vaya desarrollando el diálogo [1]. Un reto importante es detectar los casos en los que el usuario quiere volver a un contexto anterior y modificar algún detalle.

La variedad de dominios exige también el diseño de varios modelos de lenguaje con diferente complejidad así como de potentes mecanismos de integración y coordinación. Por otra parte, las disfluencias acústicas debidas al habla espontánea que aparecen al cambiar de dominio, también pueden ser objeto de tratamiento a nivel del modelo lingüístico. Se puede modelar el lenguaje incluyendo específicamente estos fenómenos como eventos del léxico o bien considerarlos un ruido del canal y tratar de filtrarlos. El tratamiento de las disfluencias sintácticas (reformulaciones, falsos comienzos, etc.) mediante un modelo de lenguaje específico suministra información útil al módulo de comprensión sobre posibles borrados, inserciones o sustituciones que pueden alterar la interpretación de las intervenciones del usuario.

2.2. Sistemas de diálogo y contexto acústico

El contexto acústico lo podemos definir como el conjunto de elementos que influyen, en mayor o menor medida, en la señal captada por el/los micrófono/s que conforma/n la entrada del sistema de diálogo. Estos elementos se dividen en dos grandes grupos: aquellos que dependen del usuario y los que dependen del entorno físico que le rodea.

2.2.1. Adaptación a los condicionantes debidos al usuario

En este caso, los principales condicionantes son la edad, el sexo, lugar de origen, velocidad de locución, estado emocional, espontaneidad, grado de cooperación, etc. Aunque se ha comprobado que un sistema independiente del locutor puede obtener altas tasas de reconocimiento, éstas sólo son alcanzables en condiciones de habla controlada donde la locución del usuario se adapte al tipo de frase predominante en el corpus utilizado para el aprendizaje. Sin embargo, existe un gran grupo de usuarios para los que, por las características particulares de su voz, estos sistemas presentan unas prestaciones inaceptables. Así, por ejemplo, para usuarios con cierto tipo de minusvalías que lleven asociadas un desorden fisiológico del sistema fonador (disartria), para los cuales la utilización de un sistema de diálogo sería de gran ayuda, los sistemas independientes del locutor son totalmente inservibles. Por otra parte, velocidad de locución, estado emocional y espontaneidad, condicionantes muy ligados entre sí, dotan de gran variabilidad a la señal de voz, influyendo negativamente en las prestaciones de los sistemas de reconocimiento del habla.

Las tecnologías de adaptación y personalización al usuario son las respuestas que actualmente se están dando a los problemas derivados de los condicionantes del usuario. La adaptación al usuario se puede realizar mediante la transformación de los vectores de características extraídos de la señal de voz, o con la transformación de los modelos acústicos partiendo del menor número posible de muestras de voz, o mediante la identificación de tipos de hablantes con características similares para los que se entrenan modelos, acústicos y de lenguaje, específicos. El reconocedor realiza una adaptación rápida tras seleccionar el conjunto de modelos a aplicar en cada caso.

La adaptación al estado emocional del hablante es muy importante en entornos más personales del usuario (hogar, automóvil, trabajo, etc) y con habla espontánea (en el vocabulario, el lenguaje o la semántica). Diversas estrategias para mejorar las tasas de reconocimiento son: clasificación y modelado, nuevos parámetros de análisis o técnicas de restauración de la señal de voz. En los sistemas de diálogo actuales se reducen las prestaciones del reconocedor al tratar con habla espontánea. Este hecho se debe al elevado número de “disfluencias” que aparecen, incluso en diálogos usuario-máquina. El tratamiento acústico específico de las disfluencias debidas al habla espontánea es necesario para frenar esta reducción. Por otra parte, en comunidades bi o multilingües es posible adaptar los modelos acústicos y/o lingüísticos a las preferencias del usuario.

Los cambios en la velocidad de locución conlleva la modificación de la duración de los sonidos, la alteración espectral de los mismos, y cambios profundos en los efectos de coarticulación. La modificación de la duración se ha atacado proponiendo diversos modelos de duración, mientras que la alteración espectral y los efectos de coarticulación se puede englobar dentro de las técnicas genéricas de adaptación al locutor y modelado acústico.

El trabajo para adaptar los sistemas de reconocimiento automático del habla a las alteraciones fisiológicas del sistema de generación de la voz es muy complejo. Aún así, se han realizado trabajos que optan por un replanteamiento del modelado acústico, la transformación de la voz alterada a una patrón o la adaptación al locutor. Los resultados obtenidos demuestran que es necesario adaptar y personalizar el sistema a cada hablante específico dada la gran variabilidad en la voz. Especialmente interesante para estos casos es el uso complementario de información multimodal. El trabajo en la adaptación acústica al usuario es un auténtico reto por la escasez de corpus especializados para el estudio y modelado de la variabilidad de voz debida a los condicionantes dinámicos (emoción, desordenes fisiológicos, …), y por la necesidad de desarrollar nuevas tecnologías de adaptación “no estáticas” a este tipo de condicionantes.

La adaptación al usuario involucra también a otros módulos relevantes de un sistema de diálogo como por ejemplo el módulo de comprensión, que utiliza un conjunto de reglas para relacionar expresiones del lenguaje natural con los conceptos semánticos. Cuando se desarrolla un sistema, se plantea un gran abanico de reglas que cubran una gran variedad de expresiones. La personalización de este módulo se centra en seleccionar o particularizar las reglas del sistema a las expresiones que un usuario concreto utiliza con una mayor frecuencia. Esta selección de reglas permite reducir la perplejidad del sistema, mejorando sus prestaciones.

En relación con el gestor de diálogo, que se encarga de dirigir la interacción persona-sistema, la personalización o adaptación al usuario se puede concretar en dos aspectos. La definición de un perfil de usuario con preferencias sobre el uso del sistema de diálogo, puede ser de gran utilidad para reducir el número de turnos de diálogo en interacciones posteriores. Parte de la información necesaria puede obtenerse del perfil de usuario y proponerla sin necesidad de preguntarla (ej: proponer la hora a la que el usuario ha realizado viajes similares con la misma ciudad origen y la misma ciudad destino). El segundo aspecto está relacionado con la variación del ritmo del diálogo según la destreza y/o experiencia del usuario. Usuarios más inexpertos requieren de un ritmo de diálogo más pausado, mientras que un usuario con mayor destreza puede agilizar sensiblemente el proceso.

En los módulos de generación de respuesta y síntesis de voz, la personalización consiste principalmente en adaptar el mensaje al usuario para que asimile la mayor cantidad de información posible. Esta adaptación requiere de una selección inteligente de la expresión a utilizar (más o menos cantidad de información por interacción, diferente tipo de expresión a utilizar: coloquial con tratamiento “de tú” o formal con tratamiento “de usted”), y de la generación acústica posterior. Esta generación acústica puede tener características preferidas por el usuario: voz femenina en lugar de masculina, generación de emociones ante determinadas situaciones, etc. Las voces sintéticas generalmente desarrolladas en la bibliografía están pensadas para servicios por lo general no presenciales (telefónicos) apreciándose más las voces femeninas que las masculinas. La falta de variedad en estas voces sintéticas hace que sean calificadas como monótonas y aburridas en algunas aplicaciones. En un entorno doméstico y personalizado estas voces no resultan adecuadas. A fin de incrementar su amigabilidad, es necesario incorporar variedad emocional y diversidad de locutores, que haga que las voces resulten más agradables.

2.2.2. Adaptación a los condicionantes debidos al entorno físico

Dentro de estos condicionantes podemos encontrar la presencia de ruido aditivo y ruido convolucional sobre la señal de voz. Las prestaciones de un sistema de reconocimiento automático del habla se ven altamente afectadas por el entorno acústico en el cual se encuentra/n el/los micrófono/s que captan la señal de voz. También, un entorno altamente ruidoso provoca la distorsión de la señal de voz por el conocido “efecto Lombard”. En principio, los problemas asociados a la presencia de ambos tipos de ruido pueden paliarse en gran medida con el uso de micrófonos de cercanía unidireccionales (close-talk), situados muy próximos a la boca del locutor. Con ellos, se garantiza una alta relación señal a ruido y una muy baja proporción de ruido convolucional. Sin embargo, la utilización de este tipo de sensores no es posible en la mayor parte de las aplicaciones. En esos casos, el reconocimiento debe realizarse con señales de voz captadas por micrófono/s situado/s a cierta distancia del usuario, donde la relación señal a ruido se degrada considerablemente y empieza a notarse la influencia de la reverberación sobre la señal de voz captada (ruido convolucional). En este caso, distinguimos dos situaciones: una en la que la voz se sitúa en campo directo y el ruido está en campo difuso, y otra, que en los últimos años ha merecido un especial interés, en la que tanto la voz como el ruido se sitúan en campo difuso.

Las técnicas a utilizar para paliar los efectos nocivos de la captura de señales en campo lejano son, entre otras: el empleo de múltiples sensores, procesado basado en la percepción humana biaural, técnicas de procesado de señal para realzar la señal de voz (mejorando la relación señal a ruido de la misma), compensación de los parámetros acústicos de entrada (aproximándolos a los parámetros que se obtendrían con un micrófono de proximidad) o la compensación de los modelos acústicos (obtenidos mediante señal limpia de ruido de fondo y captada en un entorno anecoico asemejándose a los que se obtendrían en el nuevo entorno acústico).

En entornos acústicos especialmente adversos, se está trabajando en la combinación de información procedente de varias fuentes ortogonales para mejorar los sistemas de reconocimiento automático del habla. La información visual es una de esas fuentes no afectadas por el entorno acústico. Se ha demostrado que la incorporación de información visual a los sistemas de reconocimiento automático del habla puede elevar sus prestaciones. Sin embargo, para llevar a cabo una comparativa justa, es preciso contar con datos suficientes tomados en situaciones reales en los que la señal de vídeo también pueda estar afectada por “ruido” visual. Existen tres razones fundamentales por las cuales, la información visual ayuda al reconocimiento del habla: ayuda a localizar al locutor, contiene información segmental del habla que complementa al audio, y puede ayudar a resolver ambigüedades con información adicional sobre el lugar de articulación.

La adaptación al entorno físico también requiere cierto esfuerzo en el módulo de comprensión, el cual está orientado a paliar la pérdida de calidad que sufre el reconocedor de voz cuando se enfrenta a un entorno físico hostil. El principal efecto que se produce en este caso es que ciertas palabras pronunciadas se pierden o se reconocen con error. Para aumentar la robustez de las reglas que relacionan dichas expresiones con los conceptos semánticos hay que dotarlas de mayor flexibilidad para la extracción de los conceptos a pesar de la existencia de errores en la frase reconocida.

En relación con el gestor de diálogo la adaptación al entorno físico se puede concretar en dos puntos: al igual que ocurría con la destreza del usuario, el entorno físico puede hacer que el ritmo de diálogo cambie. El ritmo se ralentiza en condiciones acústicas hostiles para evitar que el usuario se pierda. Por otro lado, el módulo de caracterización acústica puede informar sobre la existencia de determinado tipo de ruido (TV encendida) o situación (micrófono mal colocado) obligando al gestor de diálogo a suspender la interacción e instar al usuario a que resuelva dicha situación (bajar el volumen de la TV o colocarse bien el micrófono) antes de continuar con la interacción.

En relación con el módulo de generación de respuesta, la adaptación se centra en la redacción de las expresiones a pronunciar incluyendo una mayor o menor redundancia de información, según la hostilidad del entorno acústico. En cuanto a la síntesis de voz, la graduación del nivel de volumen y la incorporación de emociones son de gran utilidad para mejorar la percepción.

2.3. Sistemas de diálogo: Evaluación y usabilidad

En USA y en Europa ha habido varias iniciativas para llevar a cabo la evaluación de los sistemas de diálogo y el estudio de su usabilidad. En el contexto americano, hay dos proyectos importantes por su contribución a estos dos aspectos: el proyecto ATIS se centró en la evaluación técnica

y el COMMUNICATOR que ha contribuido principalmente al análisis de la usabilidad de los sistemas. En Europa, varios proyectos han trabajado en la definición de los parámetros de la evaluación de los sistemas de diálogo hablado: el proyecto EAGLES [6] y el DISC [7]. Estos proyectos proponen que la evaluación debe consistir en: una evaluación técnica de los sistemas y sus componentes, una evaluación de la usabilidad de los sistemas y una evaluación de los sistemas sus componentes llevada a cabo por el usuario. En la actualidad, el marco de trabajo más extendido para la evaluación y análisis de usabilidad de sistemas de diálogo es PARADISE [8]. La solución propuesta en PARADISE está muy orientada a sistemas unimodales, ampliándose recientemente a sistemas multimodales: PROMISE [9], pero sin estar completamente desarrollado.

En el proyecto EDECÁN se abordan tres situaciones de evaluación nuevas. En primer lugar un sistema de diálogo multidominio necesita analizar nuevos parámetros de medida que midan aspectos relacionados con el cambio y gestión de varios dominios. En segundo lugar la incorporación de técnicas de adaptación y personalización incorporan cierto dinamismo en el funcionamiento que obligan también a adaptar las estrategias de evaluación y medida de usabilidad conocidas [10]. El tercer aspecto es la evaluación de sistemas con información multimodal, en este caso trabajaremos sobre la propuesta de PROMISE.

3. EL CONSORCIO EDECÁN

El consorcio EDECÁN está constituido por 4 grupos de investigación en Tecnologías del Habla:

EDECÁN-UZ: Grupo de Tecnologías de Comunicaciones, Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza.

EDECÁN- UPV: Grupo de Reconocimiento de Formas, Inteligencia Artificial, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia.

EDECÁN-EHU: Grupo de Reconocimiento de Formas y Tecnologías del Habla. Departamento de Electricidad y Electrónica. Universidad del País Vasco/Euskal Herriko Unibertsitatea .

EDECÁN-UPM: Grupo de Tecnología del Habla, Departamento de Ingeniería Electrónica, Universidad Politécnica de Madrid.

Las diferencias en el origen y especialización de cada uno de los grupos hacen especialmente atractivo el trabajo en conjunto. El grupo EDECÁN-UZ proviene del ámbito de teoría de la señal y las comunicaciones, aportando un conocimiento fundamental para el desarrollo de técnicas de robustez de sistemas de reconocimiento, tanto en técnicas de parametrización de la señal acústica como en el modelado acústico. Los grupos EDECÁN-UPV y EDECÁN-EHU provienen del ámbito del reconocimiento de formas y poseen conocimiento esencial sobre el aprendizaje automático de los modelos. Su presencia permite al consorcio profundizar en el desarrollo de metodologías de aprendizaje automático a partir de muestras, tanto para el tratamiento de fenómenos de habla espontánea como para el diseño de sistemas de modelado de lenguaje, comprensión y diálogo. El grupo EDECÁN-UPV también trabaja en el ámbito del tratamiento del lenguaje natural, en particular en etiquetadores morfosintácticos, desambiguación del sentido de las palabras, reconocimiento de entidades con nombre y su aplicación en recuperación y extracción de información y búsqueda de respuestas. El grupo EDECÁN-UPM tiene una extensa experiencia en el diseño y evaluación de sistemas de diálogo persona-máquina basados en tecnología del habla, con potentes sistemas de reconocimiento, comprensión de habla y conversión de texto a voz de alta calidad.

4. OBJETIVOS DE EDECÁN

EDECÁN es un proyecto de investigación horizontal que necesita de la investigación y desarrollo tanto en tecnologías electrónica y de comunicaciones como en tecnologías informáticas. El proyecto aborda temáticas relacionadas con las técnicas avanzadas de procesado digital de señal y de reconocimiento robusto de voz dentro de los objetivos de robustez frente al contexto acústico. Por otro lado, el proyecto, en su conjunto, aborda el desarrollo de un sistema completo de diálogo multidominio adaptable y personalizable.

Objetivos científicos:

- Optimización y mejora de los componentes de los sistemas de diálogo disponibles:
  - Mejora de la robustez y prestaciones de los sistemas de diálogo disponibles en cada grupo.
  - Diseño e implementación de nuevos componentes (generación de habla y síntesis con emociones e integración básica de información multimodal).
- Búsqueda de mecanismos/técnicas eficientes que mantengan las prestaciones de los sistemas frente a cambios en el contexto acústico (en general: físico, del locutor, etc.). **[Cambio de contexto acústico]**
- Búsqueda de mecanismos/técnicas eficientes que minimicen el esfuerzo invertido para generar un nuevo sistema de diálogo. **[Cambio de contexto semántico]**
- Búsqueda de mecanismos eficientes para que los sistemas exploten la información disponible en un perfil de usuario, con el objetivo de adaptar el comportamiento del sistema a las preferencias / capacidades / características de los usuarios. **[Personalización]**
- Estudio de mecanismos contrastados para llevar a cabo la evaluación de las prestaciones y de la usabilidad. **[Evaluación]**

Objetivos tecnológicos:

- Sobre el corpus
  - Adquisición y trascripción de un corpus específico para la investigación en la adaptación y personalización al contexto acústico y de aplicación.
  - Etiquetado del corpus anterior en términos de los elementos condicionantes del contexto acústico y de aplicación.
- Sobre el prototipo
  - Desarrollo de una plataforma para la integración de las diferentes unidades de los sistemas de diálogo.
  - Construcción de un prototipo de sistema de diálogo que integre los resultados científicos y tecnológicos sobre dos contextos de aplicación definidos por un servicio de información del transporte y un servicio de asistente en el hogar y en el coche.

Por otro lado, dentro de estos objetivos generales del proyecto, cada subproyecto tiene unos objetivos particulares:

El **subproyecto EDECÁN-UZ** se centra en la investigación y desarrollo de tecnologías de adaptación al contexto acústico para los sistemas de reconocimiento

automático de habla. En particular, este subproyecto se plantea como objetivos científicos concretos:

- La investigación en representaciones de la señal de voz robustas para los distintos entornos acústicos
- La identificación de entornos acústicos y la identificación de locutores y tipos de locutores para la personalización y definición de perfiles de usuario y de aplicación
- La adaptación de los modelos acústicos al entorno acústico, locutor, estado emocional, velocidad de locución y alteraciones de la voz.
- El reconocimiento con micrófonos lejanos, especial énfasis se dará a la utilización de agrupaciones de micrófonos y reconocimiento biaural. Se profundizará en el estudio de dos situaciones. Una primera en la que la voz se sitúa en campo directo y el ruido está en campo difuso y una segunda situación, que en los últimos años ha merecido un especial interés, en la que tanto la voz como el ruido se sitúan en campo difuso.
- Nuevos algoritmos de decodificación acústico-fonética que posibiliten la adaptación "on-line" tanto a nivel de modelos acústicos como de parámetros acústicos.

El **subproyecto EDECÁN-UPV** se centra en el área de adaptación de los componentes de comprensión y gestión del diálogo a los cambios de contexto, tanto acústico como de dominio semántico. En particular, este subproyecto se plantea como objetivos científicos concretos:

- El desarrollo de técnicas para mejorar la robustez de las fuentes de conocimiento de los sistemas de reconocimiento, en lo referente a los modelos de lenguaje, modelos de comprensión del habla y en la estrategia y modelos para el gestor de diálogo, frente a los cambios de contexto acústico.
- La búsqueda de técnicas que minimicen el esfuerzo invertido para generar un nuevo sistema de diálogo. Para ello se propone el aprendizaje automático (uso de métodos supervisados y no supervisados) de modelos de lenguaje y de modelos de comprensión del habla, el desarrollo de técnicas de adaptación a nuevas tareas.
- El estudio de metodologías de aprendizaje de la estrategia del diálogo para las diferentes tareas.
- Optimización y mejora de los componentes del sistema de diálogo.

El **subproyecto EDECÁN-EHU** se plantea como objetivos científicos:

- Desarrollo de tratamientos específico para el habla espontánea en los modelos acústicos, en la construcción del léxico y en el modelo de lenguaje.
- Estudio de formulaciones unificadas para la integración ponderada de modelos de estados finitos en el reconocedor.
- Desarrollo de técnicas de adaptación rápida de los modelos acústicos a las características de usuarios desconocidos.
- Desarrollo de modelos de lenguaje específicos para cada dominio y de técnicas de coordinación de los mismos.

Por otra parte se propone incorporar los corpus, modelos y experiencia adquirida en el tratamiento específico del Euskera en reconocimiento automático del habla para poder personalizar los prototipos que se deriven a posibles usuarios vascoparlantes.

El **subproyecto EDECÁN-UPM** se centra en el desarrollo de nuevas tecnologías que permitan mejorar el proceso de desarrollo de Sistemas de Diálogo para su funcionamiento en varios dominios. El trabajo en estas nuevas tecnologías se orienta a tres objetivos principales:

- Adaptación al dominio acústico y de aplicación: adaptación de los modelos acústicos, léxicos y lingüísticos del reconocedor de voz, prestando interés especial al reconocimiento en condiciones de micrófono lejano (localización y enfoque).
- Personalización y adaptación al usuario: adaptación acústica, lingüística y semántica al usuario mediante la incorporación de perfiles de usuario con información de adaptación aplicable en cada uno de los módulos. Una tarea relevante en este punto es el trabajo en identificación de locutor para poder seleccionar el perfil adecuado a aplicar.

Mejora de la gestión de diálogo en un contexto multidominio: aplicación de modelos probabilísticos más potentes (Belief Networks), incorporación de nueva información en la gestión del diálogo (medidas de confianza, información multimodal y perfiles de usuario), desarrollo de herramientas de apoyo al diseño, y mejora de la generación de respuesta (generación de estilos de habla y síntesis con emociones).

AGRADECIMIENTOS

El proyecto EDECÁN no hubiera visto la luz sin el esfuerzo coordinado de todos los miembros que conforman los equipos investigadores de los cuatro subproyectos, a todos ellos, gracias por el trabajo realizado en la definición y redacción del proyecto. El proyecto EDECÁN lleva activo desde el 31 de diciembre de 2005. En diversas ponencias de estas IV Jornadas en Tecnología del Habla se presentan algunos de los trabajos realizados hasta la fecha.

BIBLIOGRAFÍA

[1] J.F. Allen, D.K. Byron, M. Dzikovska, Towards Conversational Human-Computer Interaction, AI Magazine, 2001

[2] Seneff, S. et al. Galaxy-II: a reference arquitecture for conversational systems development. Proc. ICSLP, 931-934. 1998.

[3] Schwartz, R., Miller, S., Stallard, D., Makhoul, J., Language understanding using hidden understanding models. Proc. of ICSLP. Philadelphia, USA. vol. 2, pp. 997–1000. 1996.

[4] Levin, E., Pieraccini, R., Concept-based spontaneous speech understanding system. Eurospeech'95. Madrid, Spain.

[5] Bonneau-Maynard, H., Lefevre, F. Investigating stochastic speech understanding. En Proc. of ASRU, 260-263. ,2001

[6] Expert Advisory Group on Language Engineering Standards. http://www.spectrum.uni-bielefeld/EAGLES/.

[7] Dialogue Engineering Best Practice Methodology. http://www.disc2.dk. 1999.

[8] Walker, M.A., Litman, D.J., Kamm, C.A., Abella. 1998a. PARADISE: A general framework for evaluating spoken dialogue agents. ACL/EACL 97, pp 271-280.

[9] Beringer, N., Kartal, U., Louka, K., Schiel, F., Turk, U., 2002. PROMISE—a procedure for multimodal interactive system evaluation. In: Proc. of the LREC Workshop on Multimodal Resources and Multimodal Systems Evaluation, Las Palmas. pp. 77–80.

[10] Litman, D., Shimei P, 2002. Designing and Evaluating an Adaptive Spoken Dialogue System. User Modeling and User-Adapted Interaction 13, 111-137, 2002.

A VIRTUAL BUTLER CONTROLLED BY SPEECH

A. Uria, A. Ortega, M. I. Torres, A. Miguel*, V. Guijarrubia, L. Buera*,*
J. Garmendia, E. Lleida*, O. Aizpuru, A. Varona, E. Alonso, O. Saz*

Departamento de Electricidad y Electrónica
Facultad de Ciencia y Tecnología.
Universidad del País Vasco
manes@we.lc.ehu.es

* Communication Technologies Group (GTC)
I3A, University of Zaragoza, Spain
lleida@unizar.es

ABSTRACT

The aim of this work was to develop a virtual butler-service to be installed at home to control electrical appliances and to provide information about their conditions. The framework of this project lays on FAGOR Home Appliance. The overall goal was to develop a smart home where anyone, even physically handicapped people, could control the appliance with the voice. A spoken dialog system was developed allowing a spontaneous and speaker independent speech input. A Speech Recognition module is used to convert the speech input signal into text. A speech-understanding model translates the recognized utterance into a sequence of task dependent frames. The Dialog Manager generates the most suitable answer. It also controls the device activation when the required information is ready. The butler will be able to fully control and program a washing machine, a dishwasher machine and an oven, being able to show some recipes. In order to test the Speech Recognition module, a specific corpus was recorded in the FAGOR Home Appliance facilities. Some experiments were carried out with this corpus, obtaining up to a 67,31% of improvement when using speaker and environment adaptation.

1. INTRODUCTION

The framework of this project lays on FAGOR Home Appliances, which is a electrical appliance multinational in Spain: the 44% of FAGOR sales are on the international market, and 70% of these overseas sales are made in countries as competive as France, Germany and the United Kingdom.

The aim of this work is to develop a virtual butler service that would be installed at home to control electrical appliances and provide information about their conditions. The system would allow to ask for the state of each appliance, to program them or to consult a database of recipes while you are cooking. All of these tasks occur in a natural way due to a dialog that is established between the virtual butler and the user.

This work has been partially supported by the national project TIN 2005-08660-C04.

All the queries can be carried out both inside the house or by telephone outdoors. The telephone bestows mobility on the user since the system can be controlled or requested for information from anywhere.

The virtual butler enables interaction with the user. Vocal input is managed by a spoken language system, which aims to provide an effective interface between the user and the system through simple and natural dialogs. Concerning the output, the system replies to the user by generating multimodal presentations which combine spoken output, dynamical graphic displays and actions such as switching on/off or programming the different appliances.

The system accepts spontaneous speech and it is user independent, so anyone can have a conversation with the virtual butler. On the other hand, the system is independent of the platform, therefore it may be developed, deployed, and maintained on multiple operating system platforms.

This paper is organized as follows: the specific corpus for this application is shown in Section 2, the architecture of our system is presented in Section 3. This will be followed by a description of the main modules of our system, which are Speech Recognition (Section 4), Understanding (Section 5) and Dialog (Section 6), ending with the speech recognition results with the recorded corpus (Section 7) and an example to understand the system operation mode (Section 8). Finally, the conclusions and further work are presented (Section 9).

2. CORPUS

In order to complete the preliminary speech recognition results in the domotic scenario, a specific corpus was recorded in the kitchen of the FAGOR Home Appliance facilities. It is composed by 48 speakers with 125 utterances per speaker. 3 tasks were considered: control of appliances in the kitchen (90 utterances per speaker), continuous digits (15 utterances per speaker) and 20 phonetically balanced utterances per speaker. On the other hand, 8 audio channels were recorded: 3 located in the kitchen (freezer, extractor hood and washing machine), 3 placed

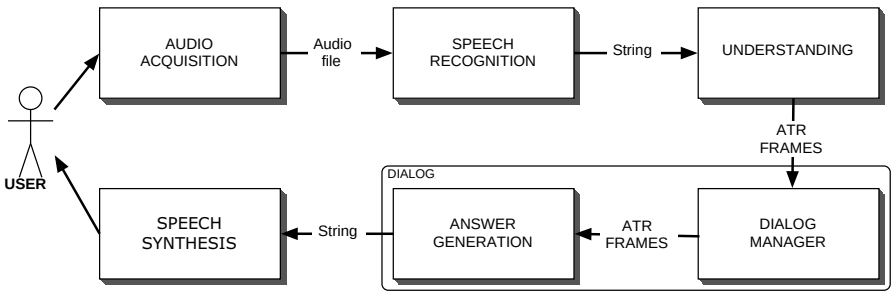


Figure 1. Architecture of the system

on the speaker, a close talk and 2 lapel microphones and finally 2 channels were recorded with a dummy (right and left ears) placed close to the speaker. In all cases, the frequency sample is 16 KHz and the audio signal is coded with 16 bits.

Three acoustic environments were considered in the recording: no appliances on (E0) with 45 dBA of typical Sound Pressure Level, SPL, extractor hood on (E1) with a 60 dBA of typical SPL, and washing machine on (E2) with a 62 dBA of typical SPL. Also, 2 speaker positions were defined: P0, in front of the washing machine, and P1, in front of the extractor hood. 15 utterances for each speaker were recorded for each position and acoustic environment.

3. ARCHITECTURE OF THE SYSTEM

The system is composed of several modules in a serial architecture, as shown in Figure 1. When the user speaks, the audio signal is captured and converted into a electrical signal by an Audio Acquisition module. This signal is decoded into a text string by the Speech Recognition module. The next step is to understand what the user expects, that is, to extract the meaning from the text string (Understanding module). The obtained information is processed by the Dialog Manager and thus an answer is generated by the Answer Generation module. This answer, in text format, is sent to the Speech Synthesizer which allows the system to ask the user for information, generating, in this way, a dialog between the human and the machine. Also other output options can be activated from the string obtained by the Answer Generation module, but in all cases this feedback pretends to work as if the user was speaking to a human butler.

It is worth to stand out that the whole system is task independent, so there is a possibility for extended plug-and-play just adapting the Answer Generation module, which is the only module that involves internal files. Furthermore this system is independent of the platform, so it may be run on multiple operating system platforms. The code of the system is Java and C so it is hardware independent application. In fact, it has been tested on Mac Os X, Linux ...

The main modules in this system, which are Speech Recognition, Understanding and Dialog, are explained in the following sections.

4. SPEECH RECOGNITION MODULE

In our system, the Speech Recognition module has a capital importance, otherwise the system will be wrong from the first stage on. This fact can generate errors in Understanding module and consequently produce mistakes in the dialog. That is, the error is propagated through all the system.

The technology of Automatic Speech Recognition systems, ASR, has evolved dramatically over decades of research. In an ASR system, we can establish two model levels which interact when searching for spoken utterances hypotheses. The first level is the acoustic level, which usually is generally assumed to follow a Hidden Markov Model, HMM, distribution, many authors have contributed to the development of these models [1, 2, 3]. The second level is the language model and referes to how words are concatenated, which can as simple as static rules of grammar, or more complex as an statistically estimated N-gram grammar [4]. This last statistical model has been selected for this work.

Another essential topic is to know how the people speak to the virtual butler. That is the reason to carry out a survey of the conversation style that people would hold on with a system. Around 250 people virtual dialog samples have been considered. A grammar has been generated with the obtained dialog samples so as to create thousands of sentences. A k-Testable in the Strict Sense Language Model (k-TSS LM) language model has been produced by inference. In this work, we make use of k-TSS language model [5] since in contrast to a probabilistic n-gram model, the former one preserves the syntactic constraints.

With regard to the acoustics models, the noises in a kitchen can be modeled and introduced in the recognition system providing a more robust system.

5. UNDERSTANDING MODULE

This module tries to get the meaning from the text string output of the Speech Recognition module looking for key words. In order to extract all this information provided by the string two concepts have been used: *Attributes* and *Frames*.

5.1. Attributes

The *Attributes* are collections of key words that have a similar meaning where each key word belonging to one and only one attribute. An example of the use of *Attributes* is shown in table 1.

Table 1. Example of an attribute

Attr_ElectricalAppliance
washing_machine
dishwasher
oven

5.2. Frames

The *Frames* are files which contain all needed information to determinate and to carry out a task. So, there is a file for each task. Each frame contains all the attributes that the system needs for the corresponding task. Apart from these attributes, each frame also contains some tags which give aid to set the task. Moreover, these frames enclose tables that help the Understanding module.

5.3. Operation mode of the Understanding module

An example is introduced in order to explain the Understanding module. The input of this module is a text string, as shown in the example 1. The Understanding module would process the input and detect the key words. In the example, the key words have been underlined to emphasize.

Example 1 *I want to turn on the oven.*

Once the key words have been detected, the Understanding module looks for the corresponding attributes. For the Example 1 the attributes are shown in the table 2.

Table 2. Attributes associated to Example 1

KEY WORD	ATTRIBUTE
turn on	Atr_Switch_on
oven	Atr_ElectricalAppliance

When the Understanding module has obtained all attributes, it checks the frames looking for the most appropriated task among all the tasks. In the example the task would be to switch on the oven.

Once the task has been determined, the Understating module completes the necessary attributes to carry out the task. In the example the Understanding module obtains another attribute: the function of the oven (grill), thanks to the tables. As a summary, the Understanding module output for the example is shown in table 3.

Table 3. Output for Understanding module to Example 1

FRAME	Switch on the Oven
ATTR : Switch_on	turn on
ATTR : ElectricalAppliance	oven
ATTR : Oven_Function	grill

6. DIALOG MODULE

From the point of view of the complexity of the dialogue, 5 levels of complexity were defined according to the used technique to represent the dialogue acts [6]. Many dialog systems have been developed during the last years covering most of this complexity levels [4, 7].

In our system, the Dialog module has two components: Dialog Manager and Answer Generation, as it can be seen in figure 1.

6.1. Dialog Manager

The operation of the Dialog Manager can be seen in Figure 2. First of all, the Dialog Manager requests a task to the Understanding module. When the task is determined, the Dialog Manager asks for the information of it to the Understanding module. With this information, an answer is generated in the Answer Generation module, which is processed by the Dialog Manager.

The answer of the user can be an affirmation, a negation or it can be a sentence without any affirmation or negation. When the speaker gives an affirmative statement, the Dialog Manager looks for more information about the task. If it obtains some information, the Answer Generation produces a new answer for the user; otherwise the Dialog Manager checks if all needed information is available. If it is not, the Answer Generation module asks about the missing information.

Otherwise, if the user gives a negative statement or the sentence does not include any affirmation or negation, the Dialog Manager looks for a new task, looking for all the needed information about the new task.

6.2. Answer Generation

The Answer Generation module produces an answer with the information provided by the Dialog Manager. The structure of this answer is defined by the task and the

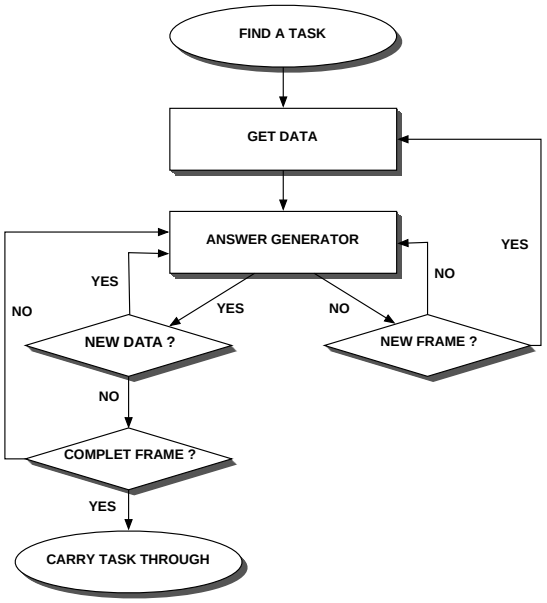


Figure 2. Way of operation of the Dialog Manager module

available information of the system, requesting for confirmation. Therefore, the Answer Generation module is task dependent.

7. SPEECH RECOGNITION RESULTS

Word acoustic models are built from a set of left and right context-dependent and context-independent units. Each unit is modelled by one-state continuous density HMMs with 16 Gaussians. In addition, two silence models for long and interword silences are considered. Each phoneme is modelled by the left contextual unit, the in-contextual unit and right contextual unit. So, for example, the word acoustic model for Spanish digit "dos" ("two") can be obtained by the concatenation of the following units: $/\#< d/\ /d/\ /d > o/\ /d < o/\ /o/\ /o > s/\ /o < s/\ /s/\ /s > \#/$, where $\#$ is the silence unit, $/ < /$ is the left context-dependent unit, $//$ is the context-independent unit and finally $/ > /$ is the right context-dependent unit. The used training corpus is composed by 7,970 utterances of the SpeechDat-Car training corpus [8] and 5,198 of the Albayzin training corpus.

The language model is a stochastic grammar, trained with the CMU Toolkit (v2.0) with 47,657 utterances generated by the virtual dialog samples. So, 363 1-grams, 1,819 2-grams and 5,379 3-grams are considered, giving a perplexity of 4.25.

Since a domotic scenario is usually used by one person, acoustic model adaptation techniques can provide a very good performance in speech recognition. In order to study it, MAP, *Maximum A Posteriori* algorithm is proposed [9], which provides better results than ML, *Maximum Likelihood*, technique when there are not enough data.

The recognition task is carried out only with the control of appliances utterances, splitting all of them into two corpus: adapting and testing, including for both of them signals of all the positions (P0 and P1) and all the acoustic environments (E0, E1 and E2). The Word Error Rate (WER) results for the different experiments are presented in Table 4, where the "baseline" column results have been obtained without any acoustic model adaptation but CMN, *Cepstral Mean Normalization* [10]. The "MAP" column refers to the WER obtained via MAP adaptation. Finally, "Improvement MAP" column presents the WER reduction concerning baseline. It can be observed that an important improvement is achieved with all different microphones.

8. AN EXAMPLE OF A DIALOG

In this Section we will show an example of a dialog. In this case, a user wants to turn on the oven, and the system requests for the needed information. The key words are underlined for clarity and the information that is obtained by the system it is shown in Table 5.

t_1 USER: I want to cook the chicken
BUTLER: Which temperature do you want to set the oven within the grill function?

t_2 USER: No, what I want to select is the solera function
BUTLER: Which temperature do you want to set the oven within the solera function?

t_3 USER: at eighty degrees
BUTLER: Do you want to set the oven at eighty degrees?

t_4 USER: yes, for two hours time
BUTLER: Do you want to set the oven for two hours time?

t_5 USER: It is ok.
BUTLER-> Do you want something else?

...

9. CONCLUDING REMARKS AND FURTHER WORK

We have introduced a virtual butler controlled by speech which allows spontaneous speech with an independent user. The goal is to control the appliances of a kitchen. The input and the output of the system is speech, although some different ways of communication can be used (dynamical graphic displays...).

Some experiments were carried out with a specific corpus recorded in a real kitchen, showing that speaker acoustic model adaptation can improve the results in speech recognition (up to 67.31% of improvement concerning baseline).

In this work the involved language has been the Spanish but it has been considered as a further work, that the

Table 4. Speech recognition results

	Baseline	MAP	Improvement MAP (%)
Close-talk	1,91	1,34	29,69
Left lapel	4,43	1,50	66,13
Right lapel	4,88	1,59	67,31
Left Dummy	72,79	29,22	59,86
Right Dummy	54,56	22,91	58,00
Washing Machine	48,95	23,50	51,99
Extractor hood	62,90	22,98	63,47
Freezer	71,73	29,24	59,23

Table 5. Frames and attributes obtained by the system for the example in Section 8

	FRAME	ATTRIBUTES
t_1	Switch on the oven	cook (atr_SwitchOnOven) chicken (atr_food) grill (atr_OvenFuction)
t_2	Switch on the oven	solera (atr_OvenFuction) no (atr_Negation)
t_3	Switch on the oven	eighty degrees (atr_OvenTemperature)
t_4	Switch on the oven	for two hours (atr_OvenTime) yes (atr_Afirmation)
t_5	Switch on the oven	It is ok (atr_Afirmation)

user could employ another languages such as Basque or English.

Another future work can be the virtual-butler to warn about problems, such as, when it is a power cut or water or gas leak. In this situation the dialog acquires a vital importance, because in these situation it is the virtual butler who takes the initiative.

10. ACKNOWLEDGEMENTS

The authors want to acknowledge FAGOR Home Appliances for the cooperation in this work.

11. BIBLIOGRAPHY

[1] L. E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," in *Inequalities III: Proceedings of the Third Symposium on Inequalities*, Oved Shisha, Ed., University of California, Los Angeles, 1972, pp. 1–8, Academic Press.

[2] F. Jelinek, "Continuous speech recognition by statistical methods," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 532–556, 1976.

[3] L. R. Rabiner, *A Tutorial on HMM and selected Applications in Speech Recognition*, chapter 6.1, pp. 267–295, Morgan Kaufmann, 1988.

[4] J. Glass, T. J. Hazen, and I. L. Hetherington, "Real-time telephone based speech recognition in the jupiter domain," in *Proc. ICASSP*, 1999.

[5] I. Torres and A. Varona, "k-tss language models in a speech recognition systems," *Computer Speech and Language*, vol. 15, no. 2, pp. 127–149, 2001.

[6] M. Dzikovska J.F. Allen, D.K. Byron, "Towards conversational human-computer interaction," *AI Magazine*, 2001.

[7] J. Glass and E. Weinstein, "Speechbuilder: Facilitating spoken dialogue systems development," in *Proc. EUROSPEECH*, 2001.

[8] Asuncion Moreno, Borge Lindberg, Christoph Draxler, Gael Richard, Khalid Choukri, Stephan Euler, and Jeffrey Allen, "Speechdat-car. a large speech database for automotive environments," in *Proceedings of LREC*. Athens, Greece, June 2000, vol. 2, pp. 895–900.

[9] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriory estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 291–298, Apr 1994.

[10] N. Hanai and R. M. Stern, "Robust speech recognition in the automobile," in *in Proc. ICSLP*, 1994.

CONTROL DE UN ROBOT "ROBOSAPIEN" MEDIANTE VOZ

Vicent Tamarit Ballester, Carlos D. Martínez Hinarejos

Departamento de Sistemas Informáticos y Computación - DSIC
Universidad Politécnica de Valencia
Camino de Vera s/n, 46071, Valencia
{vtamarit,cmartine}@dsic.upv.es

RESUMEN

Uno de los campos más interesantes donde aplicar el reconocimiento del habla es, sin duda, aquellos entornos donde el reconocimiento se transforma en órdenes y además la cantidad de órdenes que acepta el sistema es limitada. Es el caso de las tareas que puede plantear la domótica, el manejo de pequeños aparatos eléctricos. En este artículo se presenta una tarea de esta clase: el manejo de un robot de juguete, con forma humanoide (un RoboSapien) a través de la voz. Se aborda, principalmente, la creación del modelo de lenguaje, que define las frases aceptadas por el sistema, así como la semántica de cada una de ellas. También se experimenta la combinación óptima de parámetros de reconocimiento.

1. INTRODUCCIÓN

Este artículo describe la construcción de un sistema de reconocimiento del habla que permita el control a través de computador de un RoboSapien. Para ello se ha diseñado un modelo de lenguaje acorde a la naturaleza del problema, teniendo en cuenta que las órdenes de voz van a ser órdenes que dirijan el comportamiento del robot.

El reconocedor debe ser capaz no sólo de aceptar las órdenes simples, sino de aceptar órdenes complejas que involucren varias órdenes simples, e incluso de posibilitar la definición de nuevas órdenes que derivarán de las 67 básicas.

1.1. El RoboSapien

Este robot humanoide, fabricado por la empresa *Wow Wee Toys* y puesto a la venta en 2004, es, en esencia, un juguete. Sin embargo, además de un divertido juguete, se convirtió al poco de salir al mercado en una buena herramienta de aprendizaje para quien se acerca por primera vez a la robótica o para quien busca una vía barata de experimentación.

El robot se maneja con un mando a distancia por infrarrojos desde el cual se pueden mandar hasta 67 órdenes, que posibilitan un control casi total de los movimientos de los brazos, así como giros, desplazamientos o la programación de sus sensores de contacto y de su sensor de sonido.

Trabajo parcialmente financiado por el proyecto TIC2003-08681-C02-02.

1.2. Manejo desde el ordenador

Para poder interactuar con el robot desde un computador se ha utilizado un circuito de construcción propia basado en el diseño propuesto en [2]. Este circuito permite la emisión de señales infrarrojas, así como su recepción, y es compatible con el software de infrarrojos LIRC (Linux Infrarred Remote Control) [6], que permite configurar dispositivos emisores/receptores bajo los sistemas GNU/Linux.

La configuración de LIRC se basa en la definición del mando a distancia, describiendo sus características técnicas y las diferentes órdenes que pueden enviarse, asociando a cada una de ellas un código que las identifique y un nombre.

2. RECONOCIMIENTO DEL HABLA PARA EL CONTROL DEL "ROBOSAPIEN"

El software de reconocimiento utilizado, sobre el que se construye la parte de reconocimiento y comprensión del sistema es ATROS [4], un acrónimo de "Automatically Trainable Recognizer Of Speech". Para adaptar este software a nuevas tareas hace falta definir tres modelos: el modelo de lenguaje, el léxico y el acústico. El modelo léxico describe la pronunciación de cada palabra como una secuencia de símbolos que representan sonidos, definidos mediante los modelos acústicos. La parte más importante es la construcción del primer modelo, pues es propio de cada tarea y su construcción no es trivial. Para la tarea que nos ocupa se han realizado dos versiones: una muy simple y otra con agrupamiento de estados. Ambas se modelan como transductores de estados finitos (TEF).

2.1. El fichero de órdenes

A la hora de construir los modelos se ha buscado sobre todo flexibilidad. Dada la naturaleza de la tarea, es posible que las frases que activan cada una de las órdenes

```
anda #WALK_ hacia adelante. #FORWARD_
anda #WALK_ hacia atras. #BACKWARD_
corre #WALK_ hacia adelante. #FORWARD_
corre #WALK_ hacia atras. #BACKWARD_
sube #UP_ el brazo #ARM_ derecho. #RIGHT_
sube #UP_ el brazo #ARM_ izquierdo. #LEFT_
```

Figura 1. Extracto del fichero de órdenes a partir del cuál se obtiene el modelo de lenguaje. Cada línea equivale a una orden.

pueda cambiar, o incluso sean diferentes para cada usuario. Pensando en esto se definió el fichero de órdenes, a partir del cual se genera toda la información necesaria para el reconocedor, y se implementaron las herramientas para extraer esa información automáticamente. En la Figura 1 se presenta un extracto del fichero de órdenes.

Cada palabra lleva asociado un significado marcado por "#". De esta forma, al reconocer la frase se emiten esos significados, con los que posteriormente puede construirse la orden final que enviar al robot.

2.2. Órdenes complejas

Se podría hacer una primera distinción de las órdenes, en función de su duración, separándolas entre finitas y continuas. Las primeras son aquellas que producen un resultado en tiempo finito, como saludar o los movimientos de los brazos. Las segundas, en cambio, se siguen ejecutando hasta que no se le da una nueva orden; la orden de correr sería una de este segundo tipo.

Sin embargo, a efectos prácticos se distinguen otros dos tipos de órdenes: simples y complejas. Una orden es simple si está formada por una o varias órdenes del mando y es compleja si la orden no está incluida en la configuración del mando (es decir, está definida por el usuario).

Una orden simple puede ser cualquier orden equivalente a cada uno de los botones del mando original del RoboSapien o a una combinación de éstas. En cambio, las órdenes complejas están orientadas a representar un comportamiento elaborado; es la forma de agrupar varias órdenes que conllevan la consecución de un movimiento concreto. Un ejemplo podría ser la definición de bailes.

Las órdenes complejas han de ser definidas por el usuario. Una posible orden compleja que agrupa los movimientos necesarios para bailar una jota sería:

```
baila una jota. #fJOTA
```

El significado "#fJOTA" indica que se trata de una orden no definida en el control remoto, ya que empieza por "f" minúscula, y está compuesta por las órdenes del fichero de texto "fJOTA". En la Figura 2 aparece el contenido de este fichero de texto. Como se puede ver, es un sencillo fichero de texto en el que pueden especificarse órdenes simples, así como repeticiones de bloques de órdenes que pueden anidarse sin límite. Cada orden debe llevar un retardo asociado. En el caso de que sea una orden finita este

```
UP_ARM_RIGHT 3
UP_ARM_RIGHT 3
UP_ARM_LEFT 3
UP_ARM_LEFT 3
repeat 5
STEP_FORWARD 3
STEP_BACKWARD 3
STEP_LEFT 3
STEP_RIGHT 3
end
```

Figura 2. Contenido del fichero fJOTA

retardo indica el tiempo que, se espera, va a costar de ejecutar la orden; en el caso de que sea una orden continua indica el tiempo que va a estar ejecutándose.

2.3. Inclusión de números en las órdenes

Algo que se ha mantenido igual en ambos modelos ha sido la inclusión de números en el vocabulario. Conforme se ha planteado la construcción del modelo, basado todo en un fichero donde cada línea es una orden, si quisiéramos incluir números deberíamos repetir la frase para cada uno de éstos. Por ejemplo:

```
Da un #1 paso #STEP_ hacia adelante. #FORWARD_
Da dos #2 pasos #STEP_ hacia adelante. #FORWARD_
Da tres #3 pasos #STEP_ hacia adelante. #FORWARD_
. . .
```

De hecho, la generación del modelo final se hace a partir de un conjunto de frases como ésas. Pero para simplificar el manejo al usuario, la herramienta acepta frases más compactas, utilizando una pseudovariable "$N". Antes de generarse el autómata final, un preproceso inicial despliega estas frases.

```
Da $N pasos #STEP_ hacia adelante. #FORWARD_
```

Se podría haber incluido algo similar para otras partes del lenguaje, como las direcciones, pero mientras que los números pueden crecer con facilidad, las direcciones se mantienen siempre en cuatro (adelante, atrás, izquierda, derecha) y no todas las órdenes que las incluyen necesitan esas cuatro. Por ejemplo no tiene sentido que las órdenes de giro incluyan las direcciones "delante", "atrás".

2.4. Modelo simple

En este modelo se construye un TEF para cada orden, independientemente de que comparta palabras con otras. Cada palabra de la frase generará una transición. A partir de todas las frases se construye el modelo final generando un nodo inicial y otro final que engloba al resto. En la Figura 3 se puede ver el TEF resultante para las dos primeras órdenes del fichero mostrado en la Figura 1.

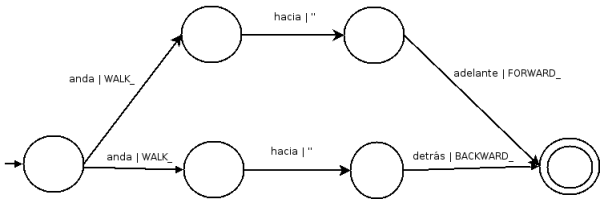


Figura 3. TEF de dos órdenes muy similares generadas siguiendo el planteamiento del modelo simple.

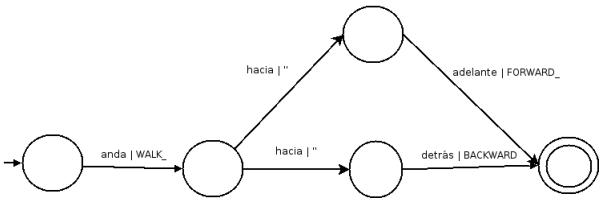


Figura 4. TEF de dos órdenes muy similares generadas siguiendo el planteamiento del modelo con agrupamiento.

Esta construcción es sencilla, pero puede llegar a ser bastante ineficiente, ya que el número de nodos y transiciones crece rápidamente con cada orden. Si suponemos una orden de n palabras, sabemos seguro que eso supondrán n nodos nuevos y $n+1$ transiciones, contando las internas entre las palabras de la orden más la transición del estado inicial al primer nodo de la frase.

2.5. Modelo con agrupamiento

El modelo anterior, por su sencillez, genera modelos con información redundante. Por ello se ha planteado otra forma de construir el modelo de lenguaje que busca obtener TEFs más compactos.

La construcción de este modelo parte también del fichero de órdenes comentado anteriormente y se divide en dos fases: un preproceso sobre las palabras y la construcción del TEF. Primero se realiza el preproceso que lee el fichero y almacena en memoria todas las palabras con su significado, si lo tienen, e información sobre si es estado inicial o final. Tras esto, se recorren todas las palabras y se van marcando como visitadas. Cada palabra nueva se compara con las ya visitadas y si coinciden sus significados, independientemente de que ellas no coincidan, se fusionan; de igual forma, aquellas palabras que no tienen significado pero son iguales también se fusionan. Con esto se pretenden agrupar sinónimos y palabras frecuentes como preposiciones.

Con toda esta información almacenada en memoria se construye el TEF definitivo que cuenta, además, con un estado inicial desde el que se transita con cada una de las palabras marcadas como iniciales, y de un estado final al que van todas las palabras marcadas como finales. El resto de palabras se transforman en transiciones internas del TEF. El resultado final se ilustra en la Figura 4. Aquí se puede apreciar como aquellas transiciones que eran sinónimos se han agrupado (ahora sólo hay un ”anda”) salvo en el caso de que ésto no fuera posible (los ”hacia” no se pueden fusionar porque preceden a una bifurcación necesaria para distinguir ”adelante” y ”detrás”).

A pesar de que existen otros algoritmos de aprendizaje para transductores mucho más potentes, como OMEGA [8] y OSTIA [7], no resultan convenientes para esta tarea debido a que suelen necesitar una muestra muy grande del lenguaje y para este caso se dispone, tan solo, de unas pocas frases.

Tabla 1. Comparación del tamaño de los dos modelos generados

MODELO	NODOS	ARISTAS
Simple	577	731
Agrupamiento	73	192

2.6. Comparación de los modelos generados

Estas dos maneras de crear el modelo de lenguaje producen salidas muy diferentes, con marcadas diferencias en el número de nodos y aristas del modelo resultante, como se puede ver en la Tabla 1. Ambos métodos de generación parten de un fichero con 56 órdenes. En total, el vocabulario consta de 207 palabras sólo para especificar las órdenes, más otras doce para los números, incluidos del uno al diez contando tres variantes para el uno: ”un”, ”uno” y ”una”. Este recuento no contempla las palabras de los significados, expresados con el símbolo ”#” en el fichero de órdenes.

3. MODELO LÉXICO

El modelo léxico consiste en un fichero que incluye todas las palabras del lenguaje reconocido e indican la secuencia de modelos acústicos que la forman. Generalmente se representa como un autómata lineal. Por ejemplo, la palabra ”gira” generaría el modelo de estados finitos de la Figura 5.

```
Name "gira"
State 0 i=1
0 1 "x" p=1
1 2 "i" p=1
2 3 "r" p=1
3 4 "a" p=1
State 4 f=1
```

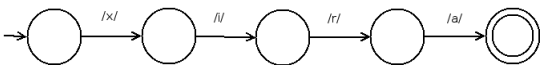


Figura 5. Autómata de estados finitos para la palabra ”gira”. Arriba, su representación en el fichero de configuración.

El modelo léxico se construye automáticamente desde el fichero de órdenes, incorporando los números que

allí no estaban definidios.

4. MODELOS ACÚSTICOS

Para los modelos acústicos se ha recurrido a un conjunto de modelos ya existente, entrenados a partir del corpus Albayzin [3].

Los modelos se representan como Modelos Ocultos de Markov (Hidden Markov Models, HMM) cuya topología es de tres estados, de izquierda a derecha, con bucles y sin saltos. Se utiliza una mixtura de 128 gaussianas (con matriz de covarianza diagonal) de 33 componentes y cada modelo es monofonema. Se utilizó HTK [5] para entrenar los modelos.

5. EXPERIMENTOS

Para poder ajustar correctamente los parámetros del reconocedor y poder evaluar el modelo de lenguaje se ha tomado un corpus de frases con el que se pudo preparar varios experimentos descritos más delante.

En la creación del corpus participaron diez locutores, cinco hombres y cinco mujeres, agrupados en parejas. Cada pareja pronunció diez frases, lo que resulta en un corpus de 50 frases pronunciadas por hombres y mujeres (en total 100 frases). El corpus está formado por 868 palabras y supone un total de cinco minutos y medio de grabación.

Las grabaciones se han realizado a 16KHz, la misma frecuencia a la que está grabado el material de entrenamiento del modelo acústico. No se buscaron en ningún momento condiciones óptimas del entorno para la grabación, a fin de poder disponer de un corpus cercano a las locuciones de un sistema en producción.

La adquisición se realizó utilizando las órdenes que para tal efecto tiene implementadas ATROS, por lo que el programa utilizado para la adquisición es el mismo con el que luego se reconoce.

Los batería de experimentos se llevó a cabo utilizando la versión off-line de ATROS. Los parámetros a variar son los comentados a continuación:

- GrammarBeam: Cada transición del modelo de lenguaje se despliega en un HMM sobre el que se aplica el algoritmo de Viterbi [1]. Si la transición recibe una puntuación mayor que la mejor puntuación más este factor se poda, dejan de explorarse los caminos que desde ella se pudieran generar. Con este factor se controla el número de caminos que hay que explorar. Para valores bajos muchos caminos se podarán, lo que previsiblemente repercutirá en un mayor error de reconocimiento pero en un menor tiempo de cálculo, dado que cada iteración requiere actualizar menos caminos.

- PhonemeBeam: Igual que el parámetro anterior pero a nivel del modelo acústico.

- GrammarScaleFactor: Este parámetro modifica el peso de las probabilidades del modelo de lenguaje frente al modelo acústico. Es útil para equilibrar la influencia del modelo acústico y la gramática. Valores altos dan más importancia a la gramática, mientras que valores bajos hacen que el modelo acústico guíe el reconocimiento.

- WordInsertionPenalty: Penaliza la inserción de nuevas palabras. Cuanto más alto es este factor más se premia a las frases cortas. Este parámetro es necesario porque frase largas, o incluso palabras, pueden decodificarse como palabras más cortas, rompiendo el significado original. Por ejemplo palabras compuestas como ”automóvil”, pueden ser reconocidas como ”auto” y ”móvil”. Con este parámetro se controla este efecto en el reconocimiento.

Se preparon dos baterías de experimentos, la primera para comparar los dos modelos de lenguaje y observar cuál se comporta mejor. La segunda buscaba afinar la primera búsqueda para encontrar los parámetros óptimos.

Los errores que se han tenido en cuenta son los relativos a la semántica de las órdenes, no a la orden en sí; es decir, no se ha tenido en cuenta si ATROS ha reconocido bien la orden, sino si ha reconocido bien el significado de la orden dada. Sobre el significado de las órdenes se han medido, a su vez, dos errores diferentes: el error a nivel de palabra (Word Error Rate), y el error a nivel de frase (Sentence Error Rate). Cada frase reconocida puede contener una o varias órdenes; así, en este caso, el WER se refiere al error de las órdenes a nivel individual, mientras que el SER se refiere al error de todas las órdenes reconocidas en una frase. Para el corpus adquirido la perplejidad es de $2{,}975$ sobre el modelo simple y de $3{,}004$ para el modelo con agrupamiento.

5.1. Comparación de modelos

La primera fue una búsqueda exhaustiva, haciendo un barrido por los cuatro parámetros: GrammarBeam, PhonemeBeam, GrammarScaleFactor (GSF) y WordInsertionPenalty (WIP). Para los dos primeros se han usado los mismos valores, por lo que, en la práctica, la búsqueda ha sido sobre tres parámetros, que en adelante se referirán como: BEAM, GSF y WIP.

En la Tabla 2 se muestran los resultados más destacados para el modelo de lenguaje simple. Se puede ver cómo los resultados mejoran al aumentar el BEAM, pero llegado un punto no merece la pena seguir aumentándolo. El BEAM establece el rango de búsqueda en el modelo de Markov final. Dado que el modelo de lenguaje no es demasiado complejo, al aumentar el parámetro llega un momento que el rango de búsqueda que se establece no elimina caminos óptimos y por ello aumentar este valor no mejora los resultados.

Los resultados para el modelo con agrupamiento están en la Tabla 3. Ocurre lo mismo con este modelo que con

Tabla 2. Tabla con los resultados para el modelo de lenguaje simple. En negrita el mejor resultado.

Error para la semántica (WER/SER)					
			BEAM		
		WIP	200	400	600
GSF	1	0	9.35/24	7.57/21	7.57/21
		1	9.79/24	8.01/21	8.01/21
		5	9.35/21	8.31/19	8.31/19
GSF	5	0	7.72/20	**6.38/17**	6.38/17
		1	7.72/20	6.97/18	6.97/18
		5	8.16/21	7.12/18	7.12/18
GSF	10	0	7.86/21	6.38/18	6.38/18
		1	7.86/21	6.38/18	6.38/18
		5	9.20/21	7.57/18	7.57/18

Tabla 3. Tabla con los resultados para el modelo de lenguaje con agrupamiento. En negrita el mejor resultado.

Error para la semántica (WER/SER)					
			BEAM		
		WIP	200	400	600
GSF	1	0	11.72/28	9.20/26	9.20/26
		1	10.83/27	8.61/25	8.61/25
		5	12.17/31	9.64/28	9.64/28
GSF	5	0	10.68/26	**8.31/24**	8.31/24
		1	10.83/27	8.46/25	8.46/25
		5	13.50/33	10.39/29	10.39/29
GSF	10	0	11.72/30	9.05/27	9.05/27
		1	12.17/31	9.50/28	9.50/28
		5	15.88/37	13.06/33	13.06/33

el anterior: llegado un punto, aumentar el BEAM, no mejora el reconocimiento. Sin embargo, a pesar de mantener el mismo comportamiento, los resultados generales son peores que con el modelo anterior. El culpable de este empeoramiento es la generalización que supone la fusión de estados. Ahora frases como "gira hacia la derecha" y "anda hacia adelante" se podrían reconocer como "gira hacia adelante", debido a que el agrupamiento ha unido todos los "hacia".

El modelo simple presenta, como se ha visto, mejores resultados. Para esta tarea concreta, con pocas órdenes, este modelo sería la elección más lógica. Para modelos mucho más grandes, donde el HMM de reconocimiento puede crecer en exceso generando peores tiempos de reconocimiento, podría plantearse el uso del modelo con agrupamientos, buscando un compromiso entre el tiempo de reconocimiento y el error. Pero para esta tarea la diferencia temporal entre ambos modelos, aunque notable, no es significativa para el reconocimiento. Como se puede ver en la Tabla 4, todos los tiempos, salvo para el BEAM de 1000, están por debajo de los 10 milisegundos. ATROS extrae características de fragmentos de señal de 10 milisegundos; por lo tanto puede considerarse como una tarea en tiempo real para cualquiera de los dos modelos, pues en ningún caso se supera ese tiempo.

Tabla 4. Tiempos de reconocimientos para diferentes BEAM. Se han tomado con GSF de 5 y WIP de 1.

Tiempos para el reconocimiento (segundos/frame)		
BEAM	Simple	Agrupamiento
100	0.002	0.002
200	0.005	0.004
400	0.008	0.006
600	0.009	0.007
1000	0.010	0.007

Tabla 5. Tabla con los resultados de los experimentos para la búsqueda de parámetros óptimos para el modelo normal, con un BEAM de 400. En negrita el mejor resultado.

Error para la semántica (WER/SER)			
	GSF		
WIP	2	5	7
0	7.12/20	**6.38/17**	6.68/18
1	6.53/18	6.97/18	6.68/18
2	6.97/18	7.42/18	7.12/18
3	7.12/18	7.42/18	7.12/18
4	7.86/18	7.42/18	6.82/18
5	7.86/18	7.12/18	6.82/18

5.2. Refinamiento de los parámetros

En las pruebas generales lanzadas se ve como el comportamiento no mejora al aumentar el BEAM, pero sí que hay pequeñas variaciones al modificar el resto de parámetros. Por ello se preparó un segundo bloque de experimentos donde el BEAM se fijó a 400 y se varió el GSF entre 0 y 10 y el WIP entre 0 y 5 en incrementos unitarios. Dado que la diferencia de error es tan sólo del 2 %, los experimentos se repitieron para ambos modelos.

En la Tabla 5 se muestran los resultados más significativos para el modelo simple. La columna central muestra los resultados del mejor GSF, las otras dos, una con un GSF superior y la otra inferior, confirman la tendencia de los errores a minimizarse en el valor óptimo. Para el modelo simple no se aprecia ningún tipo de mejora y no es previsible que una búsqueda aún más minuciosa vaya a proporcionar mejores resultados, por lo que la mejor configuración sigue siendo un BEAM de 400, un GSF de 5 y un WIP igual a 0.

En la Tabla 6 se muestran los resultados para el modelo con agrupamiento. Igual que en el caso anterior se han incluido los valores para tres GSF. Para este modelo, los nuevos experimentos sí que muestran una leve mejora del error.

5.3. Análisis de errores

Partiendo del mejor resultado para el modelo simple obtenido en el apartado 5.1, se ha realizado un estudio para descubrir cuáles son los errores más frecuentes y buscar una posible solución. El análisis se ha realizado sobre la semántica de las órdenes y a nivel de palabra.

Tabla 6. Tabla con los resultados de los experimentos para la búsqueda de parámetros óptimos para el modelo con agrupamiento, con un BEAM de 400. En negrita el mejor resultado.

Error para la semántica (WER/SER)			
	GSF		
WIP	5	7	9
0	8.31/24	**8.01/24**	8.61/26
1	8.46/25	8.16/25	9.20/28
2	9.05/28	9.50/29	10.39/30
3	9.35/28	10.09/29	11.13/30
4	10.09/29	10.39/29	11.72/30
5	10.39/29	10.98/29	12.02/31

Para poder ver los errores de reconocimiento se analizaron los significados de las 100 frases del experimento, una por una, comparándolas con el significado correcto. De esta forma se apreciaron tres problemas en el reconocimiento. El primero de ellos ya se veía con la matriz de confusión: algunos números como el 3 y el 10 o el 2 y el 6, se confunden entre ellos.

Los otros dos problemas son propios de la tarea y del modelo creado. Uno de ellos es la confusión entre palabras con fonética similar. Palabras como ”apágate”, ”ataca” y ”anda” se confunden con facilidad, igual que otras como ”levántate” y ”levanta”, dando lugar a varios errores. El tercer problema es también culpa de la definición del modelo de lenguaje: al crear el conjunto de órdenes se ha asociado varios significados a la misma palabra, por ejemplo ”anda” se utiliza tanto para mandar que dé un paso (STEP_) como para ordenar que ande (WALK_); la diferencia entre ambas órdenes radica en que la segunda orden es lo que en el apartado 2.2 se ha llamado orden continua, que se sigue ejecutando indefinidamente.

Resolver la confusión de los números es complicada, pero los errores causados por los otros dos problema podrían minimizarse actuando sobre el modelo de lenguaje: primero, buscando otras palabras para substituir a las que se confunden y segundo, evitando darle varios significados a la misma palabra.

6. CONCLUSIONES Y TRABAJO FUTURO

Se ha presentado en este artículo la solución a un problema real de reconocimiento del habla, como es el manejo de un robot. En este caso no pasa de ser un pequeño juguete, pero el fundamento puede ser el mismo para robots, no necesariamente humanoides, más complejos, así como también electrodomésticos y cualquier aparato electrónico con un número limitado de órdenes. Nuestra aproximación permite además generar a partir de esas órdenes simples otras más complejas, dotando al sistema, en este caso al RoboSapien, de más versatilidad.

Las posibles ampliaciones recaen sobre todo en el hardware del robot. Se podría mejorar la comunicación con él utilizando tecnologías como Bluetooth o incorporarle visión artificial. A nivel software se puede mejorar el sistema de órdenes complejas, permitiendo crear macros de comportamiento más elaboradas. También se puede revisar la construcción del modelo de lenguaje: partiendo de un gran fichero de órdenes y utilizando alguna modificación de los algoritmos de inferencia gramatical que existen se podrían obtener mejores modelos, con probabilidades más ajustadas. Otra opción más sencilla es construir un nuevo modelo teniendo en cuenta las consideraciones del análisis de errores.

7. BIBLIOGRAFíA

[1] G. Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, March 1973.

[2] Iguanaworks. Emisor/receptor para lirc. `http://iguanaworks.net/ir/`.

[3] J.Diaz, A. Rubio, A. Peinado, E.Segarra, N.Prieto, and F.Casacuberta. Development of task-oriented spanish speech corpora. In *Proceedings of the First International Conference on Language Resources and Evaluation*, pages 497–501, Granada, May 1998. ESCA.

[4] D. Llorens, F. Casacuberta, E. Segarra, J. Sánchez, P. Aibar, and M. Castro. Acoustical and syntactical modeling in the atros system. In *International Conference on Acoustic, Speech and Signal Processing, volume 2*, pages 641–644. IEEE press, March 1999.

[5] HTK Team. HTK speech recognition toolkit, 2004. http://htk.eng.cam.ac.uk/.

[6] LIRC Team. Linux infrarred remote control - lirc. `http://www.lirc.org`.

[7] J. Oncina; P. García; E. Vidal. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):448–458, 1993.

[8] J. M. Vilar. Improve the learning of sub-sequential transducers by using alignments and dictionaries. In *Proceedings of the 5th International Colloquium on Grammatical Inference: Algorithms and Applications*, volume 1891 of *Lecture Notes in Computer Science*, pages 298–311. Springer-Verlag, 2000.

[9] M.A. Jack X.D. Huang, Y. Ariki. *Hidden Markov Models for speech recognition*. Edinburgh University Press, 1990.

GEOVAQA: A VOICE ACTIVATED GEOGRAPHICAL QUESTION ANSWERING SYSTEM

Jordi Luque, Daniel Ferrés, Javier Hernando, José B. Mariño and Horacio Rodríguez

Technical University of Catalonia (UPC)
TALP Research Center
Jordi Girona, 1-3, 08034 Barcelona, Spain
{aluque, javier, canton}@gps.tsc.upc.edu , {dferres,horacio}@lsi.upc.edu

ABSTRACT

In this paper we present GeoVAQA, a Restricted Domain Spoken Question Answering system in the scope of the Spanish geography. The system consists of a web-based application that allows speech input questions about Spanish geography and sends back a concise textual answer. In our system, spoken questions are recognised by an automatic speech recognition (ASR) system. We have used RAMSES, a Spanish recogniser based on semi-continuous HMMs, to perform this task. The transcribed question is the input of GeoTALP-QA, a multilingual Geographical Domain QA system. The retrieval mechanism of the QA system uses Natural Language Processing tools, a Geographical Knowledge Base and the search engine Google to get snippets. A speech question is recorded from web client and transmitted through the Internet to the demo server which transcribes the speech and retrieves a concise answer and a set of relevant snippets.

1. INTRODUCTION

The amount of information on the Internet is growing exponentially and standard word statistic-based search engines, which return a large number of irrelevant matches, are rapidly becoming obsolete. In the same way, the increasing popularity of web-based portable computing devices with limited display capabilities has created a need for advanced information access technologies that interact with the Internet using voice and other modalities (pen, gestures, haptic, and others).

Speech Recognition (SR) is the ability by which a machine identifies spoken words. Being speech the primary means of communication between humans, Automatic Speech Recognition (ASR) systems have becoming a serious alternative to classical textual human-machine interfaces through keyboard.

Question Answering (QA) is the task of, given a query expressed in Natural Language (NL), retrieving its correct answer. QA has become a popular task in the NL Processing (NLP) research community in the framework of different Open Domain QA evaluation contests such as: Text Retrieval Conference (TREC) and Cross-Lingual Evaluation Forum (CLEF). Despite the growing interest in Open Domain QA (ODQA), there are few systems that integrate QA and ASR with a speech input. Speech interfaces to QA systems offer significant potential for finding information with phones and mobile networked devices and represent the next generation capability for universal access by integrating state-of-the-art of QA and ASR systems. Some of them are ODQA systems. [1] presented a system with iterative refinement of QA and ASR for ODQA. They used LVCSR, a publicly available ASR from the Institute of Signal and Information Processing (ISSP)[1], trained on a database of broadcast news. [2] described a speech interface for the AnswerBus[2] QA system using a commercial dictation engine (Dragon Naturally Speaking 6.1). [3] presented QASR, a system using semantic role labelling for QA and the AT&T Watson speech recogniser. [4] presented SPIQA, a spoken interactive ODQA system for Japanese that uses a Weighted Finite-State Transducers (WFST) approach for ASR.

In this paper we describe our experiments in the adaptation and evaluation of an ASR Voice Interface for GeoTALP-QA, a multilingual Geographical Domain Question Answering (GDQA) system [5]. The basic approach of the QA module consists of applying language-dependent processes on both question and passages for getting a language independent semantic representation, and then extracting a set of Semantic Constraints (SC) for each question. Two algorithms, a Semantic Constraint Relaxing and a Frequency-based one, have been applied for answer extraction.

We outline below the organisation of the paper. In section 2, we describe the framework of the demonstration, focusing on the web strategy implementation and the user interface. In section 3, we present the overall architecture of GeoVAQA and describe briefly its main components for ASR and QA. Then, the experiments are presented in section 4. Finally, the last section provides the conclusions.

[1] **ISSP**. http://www.isip.msstate.edu
[2] **AnswerBus**. http://www.answerbus.com

Fig. 1. *Applet user interface*
Demo. *http://gps-tsc.upc.edu/veu/aliado/demos.html*

2. APPLICATION DESCRIPTION

The application consists of a Geographic Restricted Domain QA system that processes oral questions in Spanish in the domain of the Spanish geography, and returns a concise answer and a set of relevant snippets. The user interface consists in a web-based applet allowing speech recording, audio transport through internet access and information about the connection status and the final answer. The applet interface was developed using Java^{TM} 2 Platform Standard Edition Development Kit 5.0 [3].

The speech signal is recorded in the user side at 16 kHz sampling frequency, quantized using 16-bit linear coding and transmitted to the server. In the server side, the ASR system performs speech recognition and returns the speech transcription to the client. At the same time, the ASR output enables question processing and the GeoTALP-QA searches the answer and the candidates. A few seconds and the QA system sends back the answer. The applet shows a concise answer, the candidate with the highest score, and the top-scored candidates. In addition, is possible to explore a brief log of snippets from Google. A screen-shot of the applet is shown in Figure 1.

The implementation strategy allows a low cost computation in the user's terminal, which is only used to record, transmit and display information. The speech recognition and the answer processing occurs in the server side. Both tasks are processed in a Intel Pentium IV 3.0 GHz getting real-time processing in the ASR task and a good response time for the QA system.

[3] `http://java.sun.com/j2se/1.5.0/docs/`

3. ARCHITECTURE SYSTEM DESCRIPTION

In Figure 2 we depict the main components and the information flow in the GeoVAQA system. Note that the question transcribed by the ASR system sometimes incorporates not only redundant information caused by the spontaneity of human speech but also irrelevant information due to recognition errors. Therefore, with this simple integration, recognition errors degrade the performance of the QA system. Some indispensable information to extract answers is deleted or substituted by other words.

3.1. RAMSES-ASR system

We have used the recognition software RAMSES [6] to perform the speech recognition. In order to obtain a good representation of the acoustic model, a multi-condition training with a large amount of data is used to represent a variety of speakers and conditions. For this purpose we employ a database recorded in real world conditions, called SPEECON [7]. We also trained the language models by presenting the application corpus of ALBAYZIN [8], a speech geographic database. The geographical vocabulary, roughly $1,100$ words, is also obtained from the application corpus of ALBAYZIN. Recognition system front-end is based on frequency filtering (FF) features [9] and includes well known robust techniques like spectral subtraction. [10] shows that FF front-end obtain the best word error rates scores in all test using SPEECON database. The back-end part of the system is based on demiphone models [11], a context dependent sub-word unit that models independently the left and the right parts of a phoneme. The demiphone simplifies the recognition system and yields a better performance than the triphone [12]. The Databases and the ASR will be described in more detail below.

3.1.1. SPEECON and ALBAYZIN databases

SPEECON database contains 600 speakers, 550 adults and 50 children. It has been collected in 5 different scenarios: office environments, public places, entertainment, car environments and children area. A detailed description of each scenario can be found in [7]. The database was recorded at 16 kHz sampling frequency and quantizes using 16-bit linear coding. Each session was recorded simultaneously by four microphones: a head-mounted close-talk microphone, a lavalier (a microphone placed just below the chin of the speaker) and two far-field conditions microphones situated at 1 and 2-3 meters from the speaker respectively. In order to obtain the speaker independent demiphone models we used a total of $20,496$ utterances from close-talk and lavalier microphone.

ALBAYZIN [8] is a Spanish spoken database designed for speech recognition purposes. We have used the application database of ALBAYZIN to obtain a

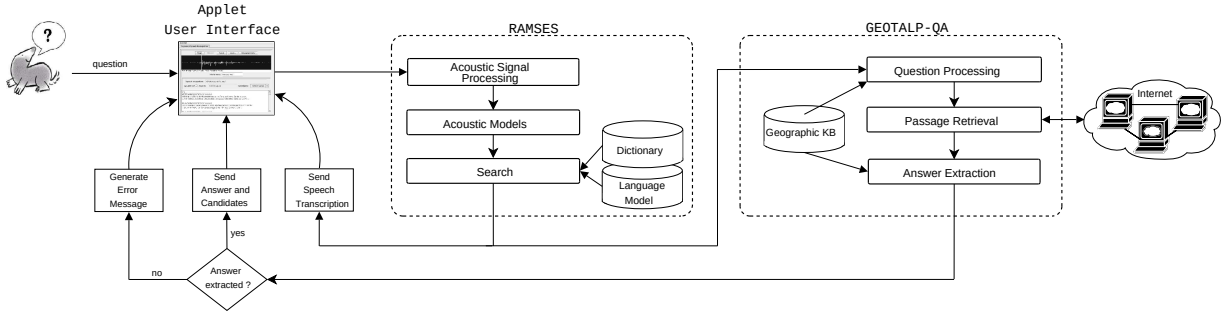


Fig. 2. *GeoVAQA System Architecture*

language model. The corpus is divided in a training set composed by $2,700$ sentences and a testing set composed by a $1,200$ sentences. This subset of the application corpus was used in several experiments in order to choice the best combination of microphones from SPEECON data. Finally, the diphonemes were trained using close-talk and lavalier subsets data. The ALBAYZIN corpus has also been divided in 78 subsets of 50 sentences each one and is uttered by 136 speakers.

3.1.2. Front-end

The front-end used obtains a signal representation by frequency filter (FF) parameters. The signal is pre-emphasized with a zero at $z = 0.97$ and Hamming windowed frames of 30 ms. were taken every 10 ms. Then, Q mel-scale filter-bank energies were computed for each frame and logarithmically compressed.

FF technique consists of a linear transformation of the log Mel-spaced filter-bank energies. That transformation essentially consists of convolving the frequency sequence of those energies with the impulse response of the FIR filter. We employ the impulse response $\{1, 0, -1\}$, which behaves as a smoothed differentiation and has generally shown good performance in comparison to Mel Frequency Cepstral Coefficients (MFCC), especially in noisy environments [9].

A feature vector of dimension 16 is calculated for each frame of the speech signal. Dynamic features are appended to the feature vector, either first derivative of the static feature vector and acceleration are included, the so called delta and delta-delta features. In addition, the delta coefficient of the logarithm of the frame energy is also computed. Thus, finally a vector of 49 features is obtained.

To deal with the background noise which degrades the speech signal we have used a spectral subtraction technique (SS) [13]. This estimate is initialized using the first frames of the utterance, and then recursively updated in those frames marked as non speech by a Voice Activity Detection (VAD) algorithm.

3.1.3. Back-end

The RAMSES recognizer is based on semi-continuous HMMs. We use 815 demiphone models [11] with 2 states per model and using a size of codebooks 512 (64 for the differential energy). Only those demiphone which appears in the training material a minimum of one hundred times are modeled, a total of 749. Those demiphones which frequency appearance is lower, the number of realizations is not enough to obtain a good HMM model, are modeled through a generalized demiphone model independently of context.

The language models were trained on text of the same topic used to build the geographical application corpus of ALBAYZIN. They are X-gram models which encode the syntax, semantics and pragmatics by concentrating on the local dependencies between words.

Four non-speech events: silence, speaker noise, filled pause and stationary noise are modeled each one by a four-state HMM. When decoding, they are combined in parallel, forming four possible alternatives, each with same probability.

3.2. GeoTALP-QA system

GeoTALP-QA has been developed within the framework of ALIADO[4] project. The system architecture uses a common schema with three phases that are performed sequentially without feedback: Question Processing (QP), Passage Retrieval (PR) and Answer Extraction (AE). More details about this architecture can be found in [5] and [14].

Before describing these phases, we introduce some additional geographical knowledge that have been used by our system for dealing with the geographic domain and some language-dependent NLP tools for Spanish.

3.2.1. Geographical Knowledge Base

A Geographical Knowledge Base (KB) of Spanish Geography has been built using the following resources: i) a set of $32,222$ non-ambiguous place names of Spain

[4] **ALIADO**. `http://gps-tsc.upc.es/veu/aliado`

with its geographical subclasses extracted from GEOnet Names Server (GNS[5]), ii) a set of 758 place names with its geographical subclasses from ALBAYZIN, iii) A set of patterns to create geographical NEs aliases (e.g. <toponym>_Mountains) iv) a lexicon containing 462 trigger words (e.g. "poblado" (ville)), v) a set of 7,632 groups of place names that refer to the same location [5], vi) a set of the most common trigger phrases (e.g. "city of *PLACE*", "*PLACE* Airport", etc.) in the geographical domain extracted from GNS gazetteer, vii) geographical subclasses, and viii) sets of words semantically related with each geographical subclass (e.g. "water" related with *sea* and *river*). The Geographical KB is used in an early stage of the QP phase to identify the geographical NEs of the transcribed question. Then the transcribed question is rewrited with the NEs in capital letters to allow that the NERC system could work properly. This KB is also used to obtain the possible geographical subclasses of each place name detected as NE.

3.2.2. Language-Dependent Processing Tools

A set of general purpose NLP tools are used for Spanish (see [15] for a more detailed description of these tools). *FreeLing*[6] is used for tokenization, morphological analysis, POS tagging, lemmatization, and partial parsing. The *ABIONET* NE Recogniser and Classifier (NERC) is used to tag NEs with basic categories (person, organization, location, and others). Finally, EuroWordNet is used to obtain a list of synsets, a list of hypernyms of each synset, and the Top Concept Ontology classes for each token. The same tools are used for the linguistic processing of both the questions and the passages.

3.2.3. Question Processing

The main goal of this subsystem is to detect the Question Type (QT), the Expected Answer Type (EAT), and the question analysis. We represent the lexical, syntactic and semantic information of the question and the passages. A language independent formalism, the *Environment*, represents the semantic content. The *Environment* of a sentence or a question contains the semantic relations that hold between the different components identified in the text. These relations are organised into an ontology of about 100 semantic classes and 25 relations. See below the classes related with the geographical domain.

```
ENTITY
    ENTITY_PROPER_PLACE
        GEOLOGICAL_REGION
            ARCHIPELAGO
            ISLAND
            LAND_FORM
                MOUNTAIN
            SEA_FORM
                CAPE
                GULF
                SEA
            WATER_FORM
                RIVER
        POLITICAL_REGION
            CITY
            CONTINENT
            COUNTY
            COUNTRY
            STATE
ENTITY_QUANTITY
  NUMERIC
MAGNITUDE
       AREA
       LENGTH
       FLOW
       WEIGHT
```

[5] **GNS**. http://earth-info.nga.mil/gns/html

[6] **Freeling**. http://www.lsi.upc.edu/~nlp/freeling

3.2.4. Question Classification

The Question Classification subsystem uses a Prolog DCG Parser with manually built rules to detect a set of 10 Question Types (see Table 1). This parser uses the following features: word form, word position in the question, lemma and part-of-speech (POS). The parser uses external information such: introductory phrases for each Question Type (e.g. "which extension" is a phrase of the QT *What_area*) and data from the Geographical KB (geographical NE subclasses, trigger words for each Geographical subclass, and semantically related words of each subclass).

Question Type	Expected Answer Type
Count_objects	NUMBER
How_many_people	NUMBER
What_area	MEASURE_AREA
What_fbw	MEASURE_FLOW
What_height	MEASURE_HEIGHT
What_length	MEASURE_LENGTH
Where_action	LOCATION_SUBCLASS
Where_location	LOCATION_SUBCLASS
Where_quality	LOCATION_SUBCLASS
Default_class	LOCATION

Table 1. QTs and Expected Answer Types

3.2.5. Passage Retrieval

The PR approach uses the search-engine Google to get snippets. We use a boolean retrieval schema that takes advantage of the phrase search option of Google and the Geographical KB to create the queries.

The algorithm used to build the queries tries to maximize the number of relevant sentences with only one query per question. First, some expansion methods described below can be applied over the keywords. The expansion methods used are: i) join trigger phrases (e.g. "isla Conejera" o "Sierra de los Pirineos"), ii) expand NEs with trigger words (e.g. "Conejera" is expanded to: ("isla Conejera" OR "Conejera")), iii) Noun Phrase expansion using the groups of place names that refer to the same location, iv) Question-based Expansion (i.e. appending keywords or expanding the query depending on the question type). Then, stop-words (including some

trigger words) are removed. Finally, only the Nouns and Verbs are extracted from the keywords list.

3.2.6. Answer Extraction

Answer Extraction can be performed using two different components: a Semantic Relaxing algorithm and Frequency Based one.

Depending on the QT, a subset of useful relations of the question *Environment* has to be selected in order to extract the answer. We define this set of relations as the *Semantic Constraints* (SC) that are supposed to be found in the *Environment* of the answer sentence. These relations are classified as mandatory (MC), or optional (OC). Then, a set of extraction rules are applied following an iterative approach. In the first iteration all the MC have to be satisfied by at least one of the candidate sentences. Then, the iteration proceeds until a threshold is reached by relaxing the MC. The relaxation process of the SC is performed by means of structural or semantic relaxation rules, using the semantic ontology. The extraction process consists on the application of a set of QT dependent extraction rules on the set of sentences that have satisfied the MC. In order to select the answer from the set of candidates, the following scores are computed and accumulated for each candidate sentence: i) the rule score (which uses factors such as the confidence of the rule used, the relevance of the OC satisfied in the matching, and the similarity between NEs occurring in the candidate sentence and the question), ii) the passage score, iii) a semantic score (see [15]) , iv) the extraction rule relaxation level score. The answer to the question is the candidate with the best global score.

Finally, a Frequency Based Extraction component has also been implemented. This extraction algorithm is quite simple. First, all snippets are pre-processed. Then, we make a ranked list of all the tokens satisfying the expected answer type of the question (see the score formulae for each candidate token in Eq. 1). Finally, the top-ranked token is extracted.

$$Score(tk_i) = \sum_{o \in Occurrence(tk_i)} \frac{1}{snippet_rank(o)} \quad (1)$$

4. EXPERIMENTS AND EVALUATION

4.1. ASR Evaluation

A test set of 722 utterances of clean speech about the Spanish geography, composed by roughly $6,600$ words, has been used to evaluate the performance of the ASR system, see section 3.1.1. A total of $20,496$ utterances from both close-talk and lavalier microphones of SPEECON database have been used to enroll the SCHMM models of the acoustic units. A lexicon of $1,086$ words is derived from the test set and the Language Model is also obtained from the test utterances. Acoustic and Language Models information are represented together by a Finite State Machine (FSA). During the testing phase, a Viterbi algorithm is in charge to make the search of the most probable sequence of demiphones. Table 2 shows the performance achieved by the ASR using several configurations of the weights of the probability path computed by Viterbi. These weights normalize the dynamic range of both acoustic and language model probabilities. Positive values of the additive weight S penalize probability transitions, hence the probability of word omissions increases. Higher values of the multiplicative weight M imply to place more confidence in the Language Model. We have fix to -1 the S value to obtain the results shown in Table 2.

	RAMSES performance		
Prob. weigth M	% Sentence	% Word	% Accuracy
9	75.90	93.88	93.82
7	76.87	94.44	94.39
5	75.90	94.32	94.26
3	64.54	90.88	90.74

Table 2. RAMSES-ASR percentage performance over the corpus testing set of ALBAYZIN about the Spanish Geography. Second and third column shows the percentage of the correct recognised sentences and words respectively

4.2. Geographical QA Evaluation

The GeoTALP-QA system has been evaluated with a test set of 62 questions about the Spanish geography (see [5] for more details about this evaluation). These questions were extracted from the geographical sub-corpus of ALBAYZIN [16], which contains utterances of questions about the geography of Spain. In Table 3 are shown the results in accuracy of the GeoTALP-QA evaluation using two different Answer Extractors: Semantic Relaxing and Frequency-Based.

	GeoTALP-QA Accuracy	
Num. Snippets	Semantic Relaxing	Frequency-Based
10	0.1774 (11/62)	0.5645 (35/62)
20	0.2580 (16/62)	0.5967 (37/62)
50	0.3387 (21/62)	0.6290 (39/62)

Table 3. GeoTALP-QA results over a test set of 62 questions about the Spanish Geography

5. CONCLUSIONS AND FUTURE WORK

We have presented GeoVAQA, a Restricted Domain Spoken Question Answering system in the scope of the Spanish geography. The aim of this paper has been to present an application be able to cope with

spoken questions about the Spanish geography domain. The system consists in a web-based application which captures and transmits spoken questions, and presents a concise answer and a set of relevant snippets. In the server side, speech recognizer RAMSES is carrying out the speech transcription which becomes the input of the GeoTALP-QA system. Finally, the transcribed question is processed by the QA system and a concise answer goes back to the text display of the web client.

Both ASR and QA systems have been evaluated separately. The ASR achieves an accuracy around 94% and 76% of correct sentences over a corpus of 722 utterances. Most of the sentences errors were produced by only one substitution or omission. Therefore, if this error occurs to a non critical word, such as a connecting word, it is possible the error does not propagate to the QA system. However, if the word error is produced on a trigger or an introductory question word, it will degrade the performance of the retrieval system. That leads, for instance, to possible feedback strategies allowing QA system interacts with the confidence of the ASR recognition.

The QA system achieved a maximum accuracy of a 62.9% of correct questions over a corpus of 62 questions. This accuracy was obtained using the Frequency-Based algorithm and a set of 50 snippets from Google. On the other hand, the Semantic Relaxing algorithm has offered a lower performance than the Frequency-Based one. The Semantic Relaxing algorithm has achieved a maximum accuracy of 33.87% using the same number of snippets.

As a further work we plan to do an evaluation of the whole system. This evaluation is required to study the ASR degradation process and the drop in accuracy of the global system (ASR and QA) with respect of the results with only QA. An improvement of the results of the QA system is required to achieve a good accuracy during the answering process. The major efforts during the QA phase must deal with the reliability of the Passage Retrieval results (it implies working with the NEs and trigger word ambiguities) and the design of fast and precise Answer Extraction algorithms.

Acknowledgements

This work has been partially supported by the European Commission (CHIL, IST-2004-506909) and the Spanish Research Dept. (ALIADO, TIC2002-04447-C02). Daniel Ferrés is supported by a UPC-Recerca grant from Universitat Politècnica de Catalunya (UPC). TALP Research Center is recognized as a Quality Research Group (2001 SGR 00254) by DURSI, the Research Department of the Catalan Government.

6. REFERENCES

[1] Sanda M. Harabagiu, Dan I. Moldovan, and Joe Picone, "Open-Domain Voice-Activated Question Answering.," in *COLING*, 2002.

[2] Edward Schofi eld and Zhiping Zheng, "A speech interface for open-domain question-answering," in *ACL*, 2003, pp. 177–180.

[3] Svetlana Stenchikova, Dilek Hakkani-Tür, and Gokhan Tur, "QASR: Spoken Question Answering Using Semantic Role Labeling.," in *ASRU*, 2005.

[4] Chiori Hori, Takaaki Hori, Hajime Tsukada, Hideki Isozaki, Yutaka Sasaki, and Eisaku Maeda, "Spoken interactive ODQA system: SPIQA," in *ACL*, 2003, pp. 153–156.

[5] Daniel Ferrés and Horacio Rodríguez, "Experiments Adapting an Open-Domain Question Answering System to the Geographical Domain Using Scope-Based Resources.," in *MLQA, EACL*, 2006, pp. 69–76.

[6] A. Bonafonte, J.B. Mariño, A. Nogueiras, and J.A. Rodríguez, "RAMSES: el sistema de reconocimiento del habla continua y gran vocabulario desarrollado por la UPC," in *VIII Jornadas de Telecom I+D*, 1998.

[7] D.J. Iskra, B. Grosskopf, K. Marasek, H. van den Heuvel, F. Diehl, and A. Kiessling, "SPEECON - Speech Databases for Consumer Devices: Database Specifi cation and Validation," in *LREC*, 2002, pp. 329–333.

[8] A. Moreno, D. Poch, A. Bonafonte, E. Lleida, J. Llisterri, J.B. Mariño, and C. Nadeu, "ALBAYZIN Speech Database: Design of the Phonetic Corpus," in *Eurospeech'93*, 1993, vol. 1, pp. 175–178.

[9] C. Nadeu, D. Macho, and J. Hernando, "Time and Frequency Filtering of Filter-Bank Energies for Robust Speech Recognition," in *Speech Communication*, 2001, vol. 34, pp. 93–114.

[10] P. Pujol, C. Nadeu, D. Macho, and J. Padrell, "Speech Recognition Experiments with the SPEECON Database Using Several Robust Front-Ends," in *ICSLP*, 2004.

[11] J.B. Mariño, A. Nogueiras, P. Pachés, and A. Bonafonte, "The Demiphone: an Effi cient Contextual Subword Unit for Continuous Speech Recognition," in *Speech Communication*, 2000, vol. 32, pp. 187–197.

[12] J.B. Mariño, P. Pachés, and A. Nogueiras, "The Demiphone versus the triphone in a decision-tree state-tying framework," in *Proceedings of the ICSLP*, 1998.

[13] S.F. Boll, "A Spectral Subtraction Algorithm for Suppression of Acoustic Noise in Speech," in *ICASSP*, 1979, pp. 200–203.

[14] D. Ferrés, S. Kanaan, A. Ageno, E. González, H. Rodríguez, M. Surdeanu, and J. Turmo, "Structural and Hierarchical Relaxing of Semantic Constraints.," in *CLEF*, 2004, pp. 557–568.

[15] D. Ferrés, S. Kanaan, E. González, A. Ageno, H. Rodríguez, M. Surdeanu, and J. Turmo, "Structural and Hierarchical Relaxation Over Semantic Constraints," in *TREC*, 2004.

[16] J. Diaz, A. Rubio, A. Peinado, E. Segarra, N. Prieto, and F. Casacuberta, "Development of Task-Oriented Spanish Speech Corpora," in *ICLRE*, 1998, pp. 497–501.

PROTOTIPO DE SISTEMA DE AUTOMATIZACIÓN DE ENCUESTAS TELEFÓNICAS

Alejandro H. Toselli[1], *Elsa Cubel Barea*[1], *Alberto Sanchis Navarro*[2]

[1]Institut Tecnològic d'Informàtica
[2]Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València
{ahector,ecubel,asanchis}@iti.upv.es

RESUMEN

En este trabajo se presenta un prototipo de sistema de automatización de encuestas telefónicas (ADET). Se describe las características relevantes del mismo, la complejidad de la tarea que implica una encuesta real y se dan las prestaciones del mismo para una encuesta dada. El prototipo es completamente funcional y accesible al público desde internet: **http://prhltdemos.iti.es/∼adet**.

1. INTRODUCCIÓN

En este trabajo se presenta un prototipo de automatización de encuesta telefónicas (ADET), desarrollado por el *Instituto Tecnológico de Informática* (ITI - www.iti.es) en colaboración con la empresa ODEC (empresa de servicios y aplicaciones informáticas de captura y proceso de datos - www.odec.es).

El objetivo concreto del prototipo es evaluar la viabilidad para el desarrollo de futuros sistemas basados en tecnología de reconocimientos espontáneo del habla, donde una de sus potenciales aplicaciones serían los sistemas de automatización de encuestas telefónicas.

En líneas generales, el prototipo ADET se ha desarrollado bajo el esquema cliente-servidor (ver figura 1), interactuando ambas partes a través de un canal de internet y de una conexión telefónica. La parte del cliente involucra un ordenador donde se ejecuta una interfaz de visualización de preguntas y resultados reconocidos (respuestas) y un teléfono. El servidor incorpora un motor de reconocimiento del habla y un módem con el cual establece la comunicación telefónica. El prototipo es completamente operativo y accesible desde **http://prhltdemos.iti.es/∼adet**.

Desde el punto de vista funcional, el prototipo se especializa, por un lado, en el reconocimiento propio de las respuestas de voz telefónica (para cada pregunta que se va realizando), obteniendo y almacenando de cada una de ellas la transcripción ortográfica de la información relevante. Por otro lado (el prototipo) se encarga también de realizar las preguntas de la encuesta, mediante la reproducción de ficheros de audio previamente grabados. No se descarta que en un futuro, esta funcionalidad esté implementada en cambio por un sintetizador de voz, que reproducirá las preguntas basándose en información textual.

El desarrollo del prototipo ha estado supeditado a tres directivas a saber:

a) La puesta a punto del prototipo para la realización de una encuesta específica, se debe realizar de forma automática con una mínima supervisión. Esta característica permite que el sistema pueda adaptarse fácilmente a cualquier tipo de encuesta.

b) El encuestado puede emitir la respuesta de un modo natural y el prototipo identifica cuál es la palabra o palabras relevantes que indican la opción elegida entre las posibles respuestas.

c) El prototipo debe incorporar la capacidad de detectar errores de reconocimiento. De este modo, cuando el sistema identifica mal la respuesta escogida, se repite la pregunta.

Sin embargo, por la gran complejidad que puede entrañar una encuesta real, solo se consideran en este caso las que involucran respuestas del tipo *multiple-choise*.

El presente trabajo se ha estructurado en cinco secciones principales. La sección 2 incluye un análisis de la complejidad que supone la automatización de encuestas telefónicas, basándonos en ejemplos de encuestas reales[1]. En la sección 3 se incluye una descripción detallada del prototipo desarrollado que permite la automatización de las encuestas telefónicas. La sección 4 muestra los resultados experimentales obtenidos para validación del prototipo. Por último, en la sección 5 se destacan las conclusiones principales de este informe.

2. ANÁLISIS DE LA COMPLEJIDAD DE LA TAREA

Una encuesta es llevada a cabo mediante la formulación, por parte del encuestador, de una serie de preguntas, y la contestación correspondiente a cada una de ellas por

Este trabajo ha sido financiado parcialmente por *Conselleria d'Empresa, Universitat i Ciència - General itat Valenciana* ID:GV06/252.

[1]Muestras proporcionadas por ODEC, S.A. (`www.odec.es`)

el encuestado. En la gran mayoría de los casos, al encuestado se le ofrece un conjunto de posibles respuestas de las que debe elegir sólo una.

Con el fin de estudiar la complejidad que supone la automatización de encuestas telefónicas, la compañía ODEC proporcionó un total de 75 ficheros de audio. Cada uno de estos ficheros contiene la grabación de una encuesta telefónica. El guión de la encuesta grabada en todas las muestras es el mismo, variando la persona encuestada y, aunque en menor frecuencia, los encuestadores/as.

En las grabaciones se observan una serie de comportamientos que aumentan enormemente la complejidad de la automatización de este tipo de encuestas. El hecho de que la encuesta sea realizada por una persona condiciona totalmente el comportamiento de la persona encuestada, siendo éste muy diferente a cómo se comportaría si la encuesta fuese realizada por una máquina. Concretamente se observan comportamientos como:

- Conversaciones fuera del ámbito de la encuesta entre la persona encuestada y encuestador.
- Formulación de opiniones personales de las personas encuestadas ante ciertas preguntas sin responder entre las opciones debidas. En estos casos el encuestador intenta reconducir a la persona encuestada para que se limite a contestar entre las posibles opciones.
- En muchas ocasiones la persona encuestada responde a la pregunta haciendo alusión a respuestas anteriores. Lógicamente el nivel de comprensión de un humano no es comparable al de una máquina.
- El encuestado no sólo emite la respuesta concreta, sino que puede incluir antes y después de la opción elegida cualquier tipo de frase.

Esta serie de dificultades, propiciadas por el marco en el que se desarrolla la encuesta, impiden la aplicación satisfactoria de un sistema de reconocimiento automático del habla que permita la transcripción automática de las respuestas de la encuesta. La automatización de encuestas telefónicas requiere pues que el procedimiento de realización de la encuesta sea distinto.

La transcripción automática de las respuestas de la encuesta puede lograrse, con un alto grado de acierto (ver sección 4), siempre y cuando se cumplan una serie de requisitos. Hay que tener en cuenta que el prototipo desarrollado (ver sección 3) sea capaz de realizar la encuesta, simplemente utilizando la información de cuáles son las posibles respuestas para cada pregunta. Éste es un aspecto interesante, ya que el prototipo no necesita de supervisión externa, aunque limita sus prestaciones o, más bien, el modo en que se debe interactuar con él.

Para que la transcripción automática de las respuestas sea posible es necesario que la encuesta se limite a ser una sucesión de preguntas-respuestas. El sistema hace la pregunta (ofreciendo las posibles respuestas) y el encuestado se limita a contestar una de las posibles opciones. La respuesta emitida no tiene por qué seguir exactamente la sintaxis de las posibles contestaciones, sino que existe un margen para la omisión o inserción de palabras. El sistema es capaz de identificar cuales son las palabras relevantes y, en consecuencia, transcribir la opción escogida. Éste sería el comportamiento lógico si la encuesta fuese realizada por una máquina, ya que se elimina la posibilidad de que el encuestado incurra en los comportamientos indicados anteriormente.

Para hacer frente a lo anterior, los modelos de lenguaje y de léxico que se han utilizado con el motor de reconocimiento del habla se generan en forma automática a partir de las posibles respuestas de cada pregunta, sin ningún tipo de supervisión. Los modelos léxico (palabras) son directamente representados por autómatas de estados finito que modelan la concatenación secuencial de sus caracteres constituyentes. Para los modelos de lenguaje se probaron diferentes tipos y configuraciones, incluyendo aquellos modelos (gramáticas) que solo aceptan la frase misma de la respuesta, pasando por modelos que buscan palabras claves dentro de la frase reconocida (word spotting), hasta llegar a los modelos n-gramas [5, 6]. Todos ellos son representados también por autómatas de estados finitos (ver figura 3).

3. EL PROTOTIPO ADET

El esquema cliente-servidor del prototipo ADET es mostrado esquemáticamente en la figura 1. El cliente se encuentra conformado por una estación de trabajo y un teléfono, que interactúan a través de una conexión de internet y una conexión telefónica (respectivamente), con un servidor donde se lleva a cabo el proceso de reconocimiento de las repuestas enviadas por teléfono. En la estación de trabajo del cliente se ejecuta una interfaz, desarrollada completamente en JAVA (ver figura 2), donde el operario, entre otras cosas, puede:

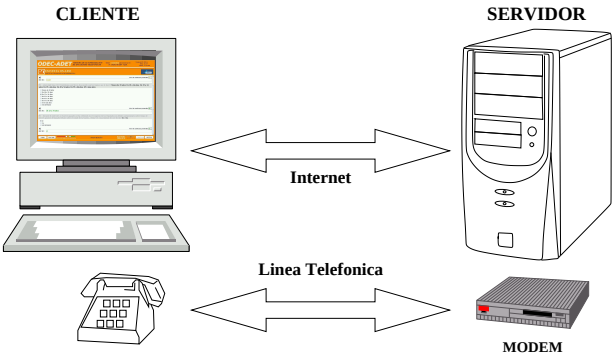


Figura 1. Esquema cliente-servidor del prototipo ADET.

1. Visualizar la secuencia de preguntas a medida que se van formulando (guión de la encuesta) y que se

pueden oír simultáneamente a través del auricular del teléfono.

2. Visualizar las posibles clases de respuestas esperadas, de estilo *multiple-choise*, y de las cuales se espera una contestación aproximada a algunas de ellas.

3. Visualizar de la respuesta reconocida (enviada por el servidor) conjuntamente con una medida de confianza [1, 2] sobre la certeza de ese reconocimiento.

4. Interactuar con el sistema por medio de una serie de controles, como por ejemplo cancelar, repetir, o finalizar la encuesta, fijar niveles de confianza del reconocimiento, etc.

Figura 2. Imagen de la interfaz gráfica del prototipo ADET.

La parte del servidor se compone principalmente de un PC y de un módem externo con el cual se establece la conexión telefónica con el cliente y por donde se reciben sus respuestas de voz. En la PC del servidor, principalmente se ejecuta un motor de reconocimiento del habla continuo operativo, denominado "ATROS" [3], desarrollado completamente en el *Instituto Tecnológico de Informática* (ITI - www.iti.es). Entre las características más sobresalientes del mismo se encuentran:

a - su flexibilidad de configuración mediante parámetros, que permiten adaptarlo a los requerimientos de memoria y tiempos de respuesta en la adquisición y reconocimiento de la señal acústica telefónica.

b - su capacidad para cargar en memoria diferentes modelos de lenguaje y poder cambiar entre ellos en función de las respuestas diferentes que deban reconocerse para cada pregunta de la encuesta.

c - compatibilidad con el formato de los modelos acústicos de Markov (HMM) utilizado por la herramienta de reconocimiento del habla HTK [4]. El entrenamiento de los **modelos acústicos** HMM para el reconocimiento de voz telefónica a 8KHZ fue realizado con este software (HTK).

Además del motor de reconocimiento del habla, en el servidor se ejecutan programas encargados de:

1 - Facilitar la interacción entre módem y motor de reconocimiento del habla, mediante un protocolo de comunicación módem-motor-de-reconocimiento y códigos-mensajes de error (previamente definidos) para evaluación del estado respectivo de ambos.

2 - La gestión y ejecución del flujo de una encuesta dada, así como la consulta del estado de evolución de la misma.

Para el intercambio de información/datos entre los procesos y programas que se ejecutan en el cliente y el servidor, se ha desarrollado e implementado un protocolo de comunicación entre ambos a través del canal de internet (TCP/IP, utilizando sockets). Así también se implementaron procesos destinados a mantener la sincronización de la transferencia de información del canal de internet y la del canal telefónico.

Como se ha mencionado anteriormente, una de las características importantes del prototipo ADET es la incorporación de las llamadas *medidas de confianza*. En cualquier problema de reconocimiento de formas, ya sea reconocimiento automático del habla, traducción automática, o reconocimiento de texto manuscrito continuo, los sistemas automáticos no son infalibles y cometen errores. Añadir a estos sistemas la capacidad de detectar cuándo se ha cometido un error es de gran interés, ya que permite al propio sistema reaccionar ante sus propios errores. Dentro de nuestro contexto, las mismas darían la opción al sistema de poder rechazar la respuesta si esta no supera un determinado umbral de confianza pre-establecido, y volver a realizar la pregunta. Este es el objetivo que persigue la estimación de medidas de confianza. La estimación de medidas de confianza se ha aplicado extensamente en el área de reconocimiento automático del habla [1].

4. EVALUACIÓN

Para evaluar el prototipo se ha utilizado una encuesta facilitada por ODEC, la cual consiste básicamente en una serie de preguntas y posibles respuestas asociadas (*multiple-choise*) sobre la situación política en España. Las preguntas fueron clasificadas según el posible conjunto de respuestas. Es decir, todas aquellas preguntas cuyo posible conjunto de respuestas coincide, pertenecen a la misma clase o tipo. Para la encuesta dada, los tipos identificados, son mostrados en la tabla 1. El número total de preguntas evaluadas es de 66.

Como muestras de voz para la evaluación se han utilizado dos corpus:

Corpus ODEC: De los 75 ficheros de audio proporcionados por ODEC se seleccionaron 6 encuestas para evaluación. El criterio para seleccionarlas fue comprobar que en ellas se seguía, más o menos, el paradigma pregunta/respuesta. Con las encuestas se-

Tipo	Nº de Preg.	Respuestas
1	15	Sí No No sabe
2	10	Mucho Bastante Poco Nada No sabe No contesta
3	13	Muy bien Bastante bien Bastante mal Muy mal Regular No recuerda No sabe No contesta
4	13	Ha mejorado Ha empeorado Sigue igual No conoce No sabe No contesta
5	13	Con la mayoría de las cosas que dijo Con bastantes de las cosas que dijo Con pocas de las cosas que dijo Con nada o casi nada de lo que dijo No sabe No contesta
6	2	Situación económica / asuntos económicos La política antiterrorista Desarrollo del Estado autonómico / Reforma de los estatutos de autonomía Modelo de financiación autonómica Estabilidad del gobierno Inmigración Temas sociales (sanidad / educación / etc.) Otros Ninguno en especial No sabe No contesta

Tabla 1. Agrupación en 6 clases diferentes de las respuestas más comunes de la encuesta facilitada por ODEC, sobre la temática política en España.

leccionadas se procedió a la extracción manual de los segmentos de voz correspondientes a las contestaciones de las preguntas de evaluación. Para cada muestra de audio extraída fue necesario transcribir cuál era la respuesta escogida.

Corpus ITI: Utilizando el prototipo se procedió a que personal del ITI realizase la encuesta. De todas las encuestas adquiridas se seleccionaron al azar 6, correspondientes a personas distintas. Para cada muestra de audio fue necesario también transcribir cuál era la respuesta escogida.

En la evaluación se experimentaron con distintos modelos del reconocedor de habla. Todos estos modelos fueron generados siempre de forma automática a partir de las posibles respuestas de cada pregunta. Como se mencionó en la sección anterior, se probaron diferentes tipos y configuraciones de modelos de lenguaje, representados por autómatas de estados finitos. Los mismos se caracterizan por el grado de restricción en la aceptación de una respuesta dada, y van desde aceptar las respuestas esperadas tal cual están escritas, hasta especializarse en buscar las palabras claves que caracterizan a cada una de las mismas. La figura 3 muestra un ejemplo de ello. Hay que

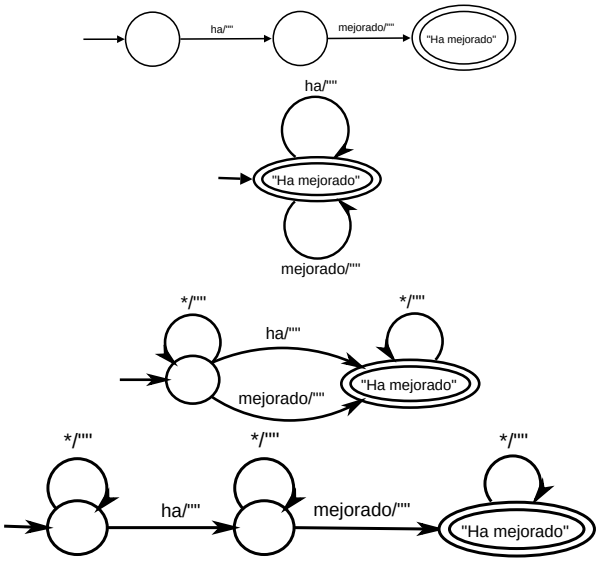


Figura 3. Modelos de lenguaje representados por autómatas de estado finito (los cuales son también transductores) que modelan la frase “ha mejorado”. De arriba hacia abajo: el primero acepta solo la frase “ha mejorado”; el segundo acepta una secuencia de las palabras “ha” y/o “mejorado” en cualquier orden y número de repeticiones; el tercero acepta una secuencia de fonemas cualquiera (representado por el *) + “ha” o “mejorado” + secuencia cualquiera de fonemas; finalmente el último modelo acepta una secuencia de fonemas cualquiera + “ha” + secuencia de fonemas cualquiera + “mejorado” + secuencia de fonemas cualquiera.

destacar que cada uno de estos autómatas que modelan

el conjunto de respuestas a cada pregunta son en realidad transductores, porque cada uno de sus respectivos estados finales emite una de las posibles cadenas de respuesta (ver ejemplo en figura 4).

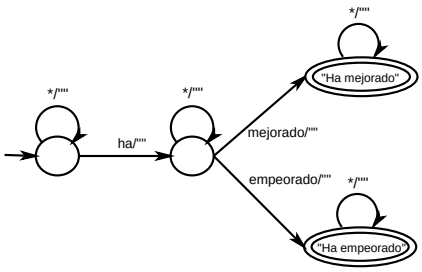


Figura 4. Ejemplo de un transductor que modela las dos posibles respuestas: “Ha mejorado” y “Ha empeorado” aceptando como entrada cadenas *aproximadas* a algunas de ellas, como por ejemplo: la frase “crea que ha mejorado mucho” sería interpretada como “Ha mejorado” y la frase “pues ha empeorado bastante” como “Ha empeorado”.

Como medida de las prestaciones del prototipo se ha calculado la tasa de aciertos de reconocimiento de respuesta. Esta tasa mide el tanto por cien de frases pronunciadas, en las que el prototipo identificó correctamente cuál era la respuesta escogida por el entrevistado. Hay que resaltar que el encuestado puede emitir la frase sin seguir exactamente la sintaxis de las respuestas posibles, y el prototipo, como resultado del reconocimiento de la frase, obtiene, gracias a la utilización de transductores, la transcripción de la respuesta seleccionada.

En las siguientes tablas 2 y 3 se muestra el acierto del prototipo ADET (en tanto por cien (%) y número absoluto (Abs)) en asignar la respuesta correcta en función de lo que dice el encuestado. Como ya se ha indicado no siempre el encuestado contesta exactamente siguiendo la sintaxis de la respuesta. Lógicamente esto es más frecuente en el corpus ODEC.

En la tabla 2 se muestran los mejores resultados, para cada corpus de voz, con la restricción de que los modelos utilizados por el reconocedor se generan automáticamente siguiendo la misma metodología. Éste sería el caso de un sistema totalmente automático exento de supervisión.

Estos resultados pueden ser mejorados añadiendo una mínima supervisión. Para algún tipo de pregunta los modelos que son más flexibles obtienen mejores resultados, mientras que para otros tipos ocurre al contrario. En la tabla 3 se muestran los mejores resultados utilizando en cada pregunta el modelo que se comporta mejor.

Como se observa, una mínima evaluación de los modelos generados automáticamente permitiría escoger los modelos apropiados para cada pregunta, repercutiendo en una mejora de las prestaciones del sistema.

5. CONCLUSIONES

El estudio de los ejemplos de encuestas proporcionadas por ODEC permite concluir que la automatización de encuestas telefónicas no es posible en un entorno en el que la persona encuestada no tiene ninguna restricción a la hora de expresarse. Los sistemas automáticos de reconocimiento del habla obtienen buenos resultados siempre y cuando el vocabulario con el que deben tratar esté limitado. Justamente en esta tarea estos sistemas pueden ser muy útiles, ya que el vocabulario viene restringido por el posible conjunto de respuestas de cada pregunta. Lo único necesario es que la encuesta se realice bajo unas condiciones determinadas, de tal forma que la persona encuestada se limite a responder con cierta flexibilidad alguna de las posibles respuestas. Como se ha visto en los resultados presentados, bajo estas condiciones, el prototipo desarrollado es capaz de transcribir correctamente entre el 83 y el 87 % de las respuestas (corpus ITI). En el caso del corpus de ODEC, en el que las respuestas pueden incluir una mayor espontaneidad, dentro de un margen razonable, el acierto del prototipo también es aceptable (entre un 69 y 78 %).

El prototipo desarrollado en este proyecto es capaz de realizar encuestas telefónicas generando los modelos necesarios de forma totalmente autónoma y automática. Este aspecto proporciona la posibilidad de que sea utilizado sin necesidad de tener conocimientos en reconocimiento automático del habla. Podría ser idóneo en aplicaciones ya incipientes, como son encuestas telefónicas automáticas que se realizan, por ejemplo, para obtener datos sobre la satisfacción de clientes ante un servicio dado.

6. BIBLIOGRAFÍA

[1] A. Sanchis, *Estimación y aplicación de medidas de confianza en reconocimiento automático del habla*, Ph.D. thesis, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, May 2004.

[2] A. Sanchis, A. Juan, y E. Vidal, “Improving utterance verification using a smoothed naive bayes model,” in *IEEE International Conference on Acoustic, Speech and Signal Processing*. April 2003, vol. 1, pp. 592–595, IEEE Press.

[3] M. J. Castro, D. Llorens, J.A. S’anchez, F. Casacuberta, P. Aibar, y E. Segarra, “A fast version of the atros system,” in *European Conference on Speech Communication and Technology. EUROSPEECH’99*, Budapest, September 1999, pp. 1299–1302.

[4] S. Young, J. Odell, D. Ollason, V. Valtchev, y P. Woodland, *The HTK Book: Hidden Markov Models Toolkit V2.1*, Cambridge Research Laboratory Ltd, Mar. 1997.

	Corpus ITI			Corpus ODEC		
Tipo	Nº Muestras	Acierto (%)	Abs	NºMuestras	Acierto (%)	Abs
1	90	94.4	85	69	76.8	53
2	60	81.7	49	49	75.5	37
3	78	80.8	63	38	55.2	21
4	78	71.8	56	24	70.8	17
5	78	89.7	70	32	62.5	20
6	12	75.0	9	15	60.0	9
TOTAL	396	83.8	331	227	69.2	157

Tabla 2. Resultados de evaluación del prototipo ADET para el caso de un sistema totalmente automático exento de supervisión.

	Corpus ITI			Corpus ODEC		
Tipo	NºMuestras	Acierto (%)	Abs	NºMuestras	Acierto (%)	Abs
1	90	98.9	89	69	95.7	66
2	60	88.3	53	49	87.8	43
3	78	87.2	68	38	55.2	21
4	78	73.1	57	24	70.8	17
5	78	89.7	70	32	65.6	21
6	12	75.0	9	15	60.0	9
TOTAL	396	87.4	346	227	78.0	177

Tabla 3. Resultados de evaluación del prototipo ADET utilizando en cada pregunta el modelo que se comporta mejor.

[5] F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, 1998.

[6] Slava M. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. ASSP-35, pp. 400–401, Mar. 1987.

VOCALIZA: AN APPLICATION FOR COMPUTER-AIDED SPEECH THERAPY IN SPANISH LANGUAGE

Carlos Vaquero, Oscar Saz, Eduardo Lleida

Communications Technology Group (GTC)
I3A, University of Zaragoza
{cvaquero, oskarsaz, lleida}@unizar.es

José Manuel Marcos, César Canalís

Colegio Público de Educación
Especial Alborada, Zaragoza
cpeealborada@gmail.com

ABSTRACT

This paper introduces Vocaliza, a software application for computer-aided speech therapy in Spanish language. The objective of this application is to help the daily work of the speech therapists that train the linguistic skills of Spanish speakers with different language pathologies. The application works at three levels of language: phonological, semantic and syntactic. For this purpose, the application makes use of four speech technologies: Automatic Speech Recognition, speech synthesis, speaker adaptation and utterance verification. Vocaliza is a flexible and modular application, which will be growing in functionality during the next years. The evaluation given by a group of speech therapists to the work of Vocaliza is encouraging, as long as they consider the application useful and easy to use, which were the main objectives in the development of the application.

1. INTRODUCTION

Recently, the demand for computer-aided speech therapy software has increased as computer technologies were getting more reliable and affordable to speech therapists and people suffering speech pathologies. The most popular of these systems has been SpeechViewer by IBM, but the non existence of a version for the Spanish language and its lack of modularity made it very uncomfortable for speech therapists in Spain to use on a regular basis. In terms of research work, during the last decade many European projects related to speech technology and speech therapy such as Orto Logo-Paedia [1], SPECO [2], ISAEUS [3] and HARP [4] have been developed, some of them resulting in the development on a software for speech therapy at the end of the research process. However, there are no versions available of these softwares in Spanish language, so the applications developed in these projects can not be used by speakers and speech therapists to train communication skills in this language. Moreover, in these projects, the work of speech therapy is carried out only at phonological level, when most of speech therapists would be interested in the training of other characteristics of the spoken language.

This work has been supported by the national project TIN 2005-08660-C04-01

This paper is organized as follows: Section 2 describes the objectives that are set to the development of a computer-aided speech therapy software in Spanish language. In section 3, there is a wide description of the way of operation of the software while section 4 shows the application graphical user interface. Section 5 explains how speech technologies are included in the software for speech therapy. Then, in section 6, the evaluation of the software given by a group of speech therapists is explained, to finally extract the conclusions to this work in Section 7.

2. OBJECTIVES AND REQUIREMENTS

The objective of this work was the development of a free-to-distribute software application for speech therapy in Spanish language. For this purpose, suggestions and requirements of specialists in speech therapy were listened prior to the start of the work. The requirements set for the application can be separated from four points of view:

- In terms of the linguistic level that the application should train, there was a special interest in the fact that the application helped for the training in several levels of language: from purely phonological level to syntactic and semantic levels.
- Regarding application usability, the application should provide enough flexibility for speech therapists to work on a wide range of pathologies, while methods used to treat these pathologies should be amusing to attract users (mainly children) with pathological speech.
- The application should have a modular way of dealing with the users, this is, the information from every user should be easy to move between different computers in which the software is installed, without any loss of the user information.
- The application should be easy to use, as speech therapists and patients of speech pathology may not be used to work with computers.

All these requirements were taken into account for the final development of the application, whose given name was Vocaliza.

For this kind of work, the collaboration of experts in speech therapy and pedagogy is strongly necessary. In our case, we have counted with the assistance of the staff of the Public School for Special Education "Alborada", located in Zaragoza, Spain. Their knowledge in different fields of work with disabled children was essential for reaching the objectives of this work.

3. WORKING WITH VOCALIZA

Vocaliza operation can be summed up in a block diagram as shown in Figure 1. The application is divided into three main blocks with different functionalities, in which the exchange of information between blocks is represented by different arrows. The upper block represents the user of the application, who will manage and use all the options in the application. Each arrow stands for a different type of information:

- **Solid black arrow** represents the configuration information, as Vocaliza enables users to set up different aspects of the application in order to adapt it to every speech pathology.
- **Solid grey arrow** is the audio signal. The users of Vocaliza must talk to the application to train their communication skills, so that Vocaliza will analyze and evaluate the evolution in the speech of the user.
- **Dotted grey arrow** represents the information of the acoustic models. Vocaliza uses Automatic Speech Recognition (ASR) to enable users to train their speech, and it provides flexibility allowing the user to select which model the application will use, making possible to work with both speaker dependent and speaker independent systems.
- **Dotted black arrow** stands for word, riddle and sentence information. Vocaliza uses isolated words to train phonological language aspects, riddles to train the semantic level of language and sentences to train the syntactic level. Every word, riddle and sentence belonging to the application must have related information such as an image or text.

3.1. Games: Speech training

The games block provides different speech training methods, enabling to train every one of the three language levels that were set in the requirements of the software. The best way to train speech is speaking, so the speech training in Vocaliza is based on making the user speak. It is not difficult for an adult person to understand that speaking is a great way to improve his/her communication skills, but a child could find any speech therapy hard, tiring and boring. This is the reason for Vocaliza to use games to train user communication skills, making the application interesting to children with pathological speech.

Vocaliza features four different games to train user speech. Three games are designed to train specific levels of language, whereas the fourth game provides additional functionality. All the games follow the same scheme, the application shows some visual information to the user while playing an audio recording or synthesized speech. The user must utter the word or sentence related to the audio-visual information in order to complete the game successfully. Therefore, every game needs ASR to work properly.

The four games included in Vocaliza are:

- **Pronunciation game**: Pronunciation game is designed to train phonological language level, working with isolated words. To complete this game, the user must utter correctly the word shown on the computer screen. Then, the application will evaluate user pronunciation, displaying a score on the screen. This score will depend mainly on the improvement shown by the user in his/her pronunciation.
- **Riddle game**: Riddle game is designed to train semantic aspect of language. This game works with riddles, which are composed by a question and three possible answers. Every answer is an isolated word. The user must choose the right answer and pronounce it correctly to complete the game successfully.
- **Sentence game**: Sentence game is designed to train syntactic level of language. It works with sentences and the user must pronounce all the words in the sequence on the correct order to complete the game.
- **Evocation game**: Evocation game is designed to enable users to pronounce freely every word they might want to train. Unlike all the other games in Vocaliza, evocation game does not provide any visual or audio information to the user but a visual feedback of the word uttered by the user. As the pronunciation game does, evocation game works with isolated words and trains mainly phonological level of language.

As shown in Figure 1, the users of Vocaliza can configure the games. The application allows users to decide if they want to listen to the audio information or if the application must display any kind of text related to every word, riddle or sentence that the users may face. This enables speech therapist to adapt Vocaliza to every user requirements, depending on their communication and cognitive skills

In addition, games take information about the words, riddles and sentences that the user wants to train, as well as the acoustic model for that user (speaker dependent or speaker independent), providing flexibility to treat a wide range of speech pathologies.

3.2. Training: Speaker Adaptation

Speaker adaptation allows users to train speaker dependent acoustic models. All the games of the application

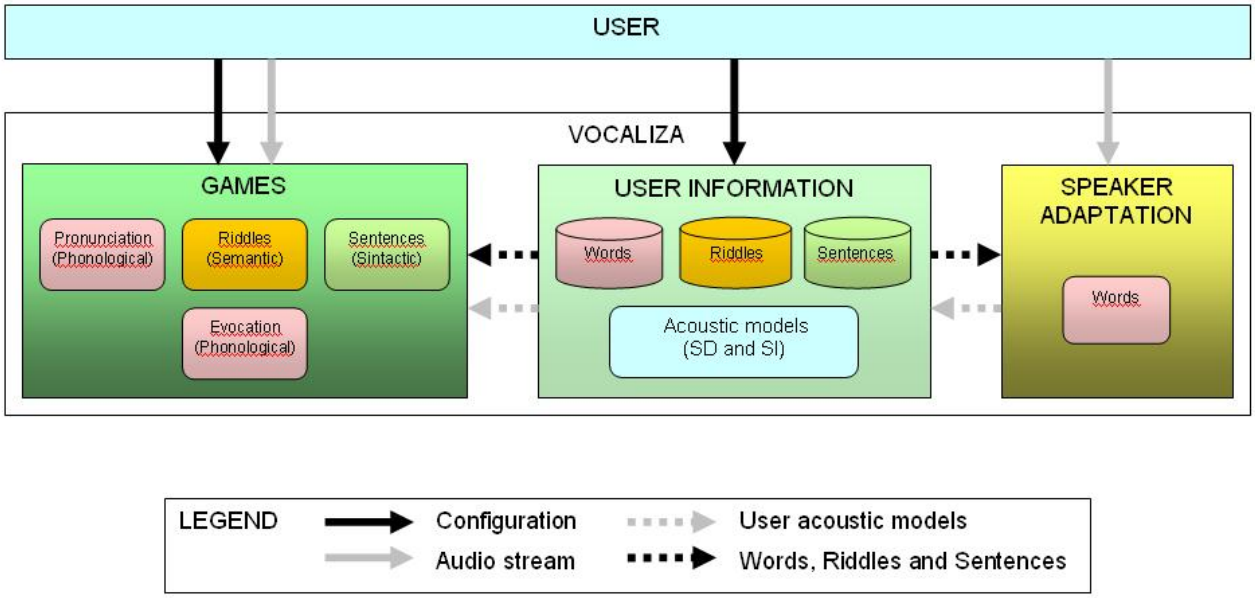


Figure 1. *Vocaliza: Block Diagram.*

are based on ASR systems, and the use of speaker adaptation is necessary to improve the performance of the ASR system when the user suffers a severe pathology. In the pronunciation game, the speaker dependent model is also necessary to give out the evaluation to the speaker speech, enabling the user to track his/her evolution and communication skills improvement.

Speaker adaptation in Vocaliza works with isolated words. The user selects which words to pronounce and how many times to utter each word, and starts recording his/her utterances (Vocaliza provides recording functionality). With this recorded data, the application will train an acoustic model adapted to the user.

As Figure 1 shows, speaker adaptation block take words stored in user information block and speech utterances of these words from the user, to train an adapted acoustic model which will be stored in the user information block.

3.3. User Information

The user information block stores all information regarding user configuration, including all words, riddles and sentences the user may face, the default configuration for the games, and every acoustic model available to the user.

Every person training his/her speech with Vocaliza should have his/her own user inside the application, in order to keep and store all the setup information related to his/her speech training. This provides great flexibility and modularity, allowing the user to add new words, riddles and sentences to train himself, as well as to select the acoustic model that the ASR system will use, being able to choose between speaker dependent or speaker independent acoustic models.

Figure 1 shows that this block is fully managed by the user, and it provides all the user information and configuration for the games. In addition, it brings word information to speaker adaptation block and stores every speaker dependent acoustic model.

All information associated with a user in Vocaliza is stored in different files which are kept in the same folder in the computer. Since users might want to train their speech in different places, and thus, in different computers, user information has been organized so that carrying it is as easy as copy the folder that contains all the information of the user from one computer to another.

4. GRAPHICAL USER INTERFACE

The Graphical User Interface (GUI) in Vocaliza was designed keeping in mind that any user with speech pathologies should be able to use the application without any help, even if the user is not used to handle computer applications. Moreover, Vocaliza GUI should provide an easy interface to manage all setup options, since speech therapists may not be used to work with computers.

Therefore, Vocaliza main window layout was designed as shown in Figure 2. This window has two important and distinct regions: the main menu, containing all configuration options and managing functionality, and the image panel, which provides easy access to every game in Vocaliza. This layout enables children with speech pathologies to play games easily, without adult supervision, using only the image panel, whereas speech therapist can access all configuration options fast and without trouble using the main menu. If the user selects a game, simply clicking on one of the four game images placed at the bottom half of the main window, the corresponding game window will open, showing all images and text related to the selected game.

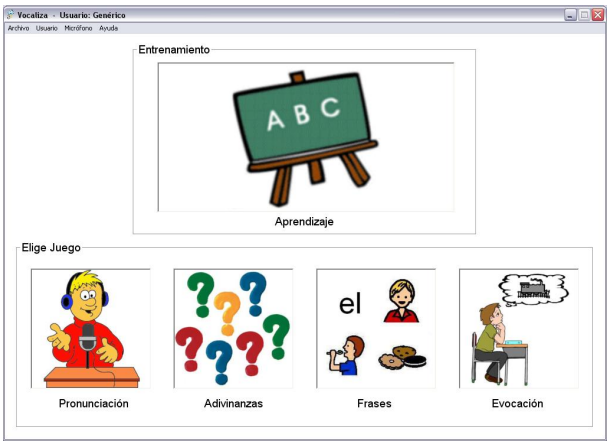


Figure 2. *Vocaliza: Main Window.*

Figure 3. *Vocaliza: Riddle Game.*

Games GUI is designed in order to avoid any user interaction different from speaking at the microphone or listening to the application speech, so user will not need to know how to use a keyboard or a mouse to play games. Only for closing the game window the use of another input mode different from speech is required. Figure 3 shows a screenshot of the riddle game.

Speaker adaptation process is designed to be as easy as possible too, but some knowledge about managing windows applications may be required, so this process is intended to require adult supervision. Speaker adaptation process can be started by clicking on the blackboard image at the top half of the main window, and it takes three steps:

- First of all, the user will select the word to pronounce and will record an utterance of the selected word, using the training window, as Figure 4 shows.
- Then, the user will check if the utterance was recorded with enough quality to use it to estimate an adapted model. For this purpose, the signal window enables the user to see and listen the recorded signal. The signal window is shown in Figure 5. These first two steps may be repeated several times, to get as many utterances of speech from the user as desired.

Figure 4. *Vocaliza: Training Window.*

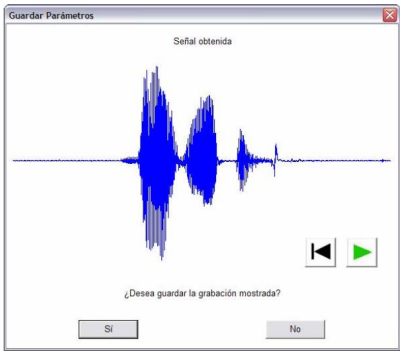


Figure 5. *Vocaliza: Signal Window.*

- Finally, the user will select which utterances will be used to estimate the adapted model, using the utterance selection window, as shown in Figure 6, and the speaker adaptation process will start.

User data and user configuration can be fully managed through user window, which is shown in Figure 7. The access to this window is done through the main menu and allows the user to add or remove words, riddles or sentences, as well as to configure all aspects of games behaviour and to select the acoustic model that ASR will use.

5. SPEECH TECHNOLOGIES IN VOCALIZA

Most of Vocaliza functionality is provided by speech technologies. Vocaliza games use an ASR system to decide if the user has completed the game successfully, speech synthesis to show how a word must be pronounced, speaker adaptation to estimate acoustic models adapted to the user, and utterance verification to evaluate user pronunciation.

5.1. Automatic Speech Recognition

Automatic Speech Recognition is the core in the Vocaliza application. Every game needs ASR to decode the user utterance, and to decide which word sequence has been pronounced so that the application will be able to let the user know if the game has been completed successfully.

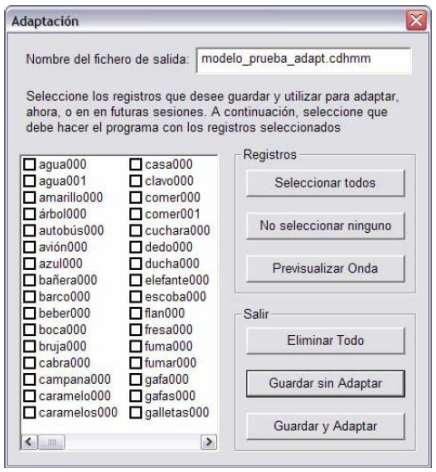


Figure 6. *Vocaliza: Utterance Selection Window.*

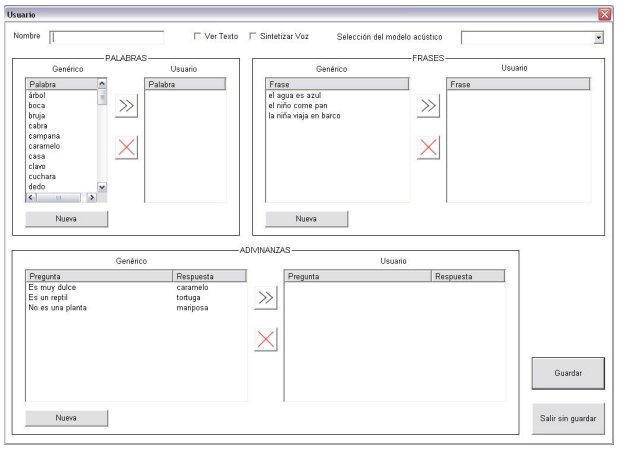


Figure 7. *Vocaliza: User Window.*

Speech signals are adquired with a sampling frequency of 16 kHz and a bit depth of 16 bits. Signals are windowed with a Hamming window of 25 ms. length, with an overlap of 15 ms, and the features used for the ASR are 37 MFCC (Mel Frequency Cepstral Coefficient), consisting on 12 static parameters, 12 delta parameters, 12 delta-delta parameters and the delta-log-energy. The units used by the system are a set of 822 context-dependent units plus a silence model and an interword model for a total set of 824 units. Every unit is modelled with 1 state per model and a 16-Gaussian mixture for every state.

5.2. Speech synthesis

Speech synthesis provides a method to show the user how a word or sentence should be pronounced. It is not the only method available in Vocaliza but is the easiest method to use: as soon as a new word, sentence or riddle is added to the application, Vocaliza is able to synthesize a correct Spanish utterance of the corresponding word, sentence or question.

However, speech synthesis may be a very strict method to teach the user how to pronounce a word or a sentence, thus, to provide flexibility, Vocaliza allows speech therapists to record word utterances, which the application will use instead of speech synthesis, in order to show different utterances depending on user age, speech pathology and other requirements of the user. For instance, slow utterances could be shown to train very young children speech.

5.3. Speaker Adaptation

Speaker adaptation enables Vocaliza to estimate speaker dependent acoustic models adapted to each user. Vocaliza uses Maximum A Posteriori (MAP) estimation [5] which, given a speaker independent acoustic model and a set of user utterances, can estimate a speaker dependent acoustic model, adapted to the user.

MAP is a well known and reliable estimation method which does not require a great number of utterances to retrieve a reliable acoustic model adapted to the user, not like other estimation methods such as Maximum Likelihood (ML) training. Not needing a great amount of data from the user is a very interesting feature since Vocaliza will estimate acoustic models from a set of utterances recorded by the user, which in most cases will consist of a small number of utterances due to two factors: speech therapists can not spend long time recording every user speech, and users with pathological speech will find very hard and tiring to record a great amount of speech. Moreover, accuracy in speaker dependent ASR based on MAP estimation methods tends to be equal to accuracy in speaker dependent ASR based on ML estimation methods when the number of utterances is high, so the use of MAP will not involve any loss in the ASR performance.

At first, there is only one acoustic model for all users in Vocaliza, which MAP will use as starting point to estimate all adapted acoustic models. As every user records his/her utterances and launches speaker adaptation, he/she will get better adapted acoustic models.

5.4. Utterance Verification

Utterance Verification (UV) is a technique embedded in Vocaliza to provide a mechanism to evaluate how the speech of the user improves. This technique tries to quantify how much has the user improve his/her communication skills.

Classically, UV has been proposed as a mechanism to verify accuracy of recognition hypotheses, which assigns measures of confidence to each hypothesized segment obtained through Viterbi segmentation of the utterance. Typically, a measure of confidence is assigned to every recognized word, and each hypothesized word is accepted or rejected depending on its corresponding measure of confidence.

Vocaliza uses a Likelihood Ratio (LR) based UV [6] procedure to assign a measure of confidence to each hy-

pothesized word in an utterance. This procedure gives the confidence measure as the ratio of the target hypothesis acoustic model likelihood with respect to an alternate hypothesis acoustic model likelihood. Choosing suitable acoustic models as target and alternate hypothesis can provide a measure of confidence which, in addition, roughly quantifies user speech improvement.

In order to obtain a confidence measure to quantify user speech improvement, Vocaliza uses a speaker independent acoustic model, which is assumed to model correct speech, as target hypothesis, and a speaker dependent acoustic model, which is assumed to be adapted to pathological speech, as alternate hypothesis. Assuming that the only significant variation that user speech will undergo during his/her treatment will be an improvement regarding his/her pathological speech, the measure of confidence obtained for every word in every utterance will increase as user improves his/her communication skill. Therefore, this measure of confidence involves a relative evaluation method to quantify user speech improvement, which will need a initial confidence measure value to work correctly.

6. EVALUATION OF VOCALIZA

To evaluate the usefulness of Vocaliza as well as all the features that the application was intended to provide (usability, modularity and flexibility), the first released Vocaliza version was handed to teachers and speech therapists in "Alborada" School. In addition to the application, two questionnaires were given to the staff of the School. The first one was aimed to the teachers and speech therapists in order to retrieve their opinion about the usefulness, modularity and flexibility of the application, whereas the second one aimed to collect the opinion of children and pupils from "Alborada" School about usability and attractiveness of Vocaliza.

All the questionnaires retrieved show the great interest that the staff of "Alborada" School has for Vocaliza. Both teachers and pupils consider Vocaliza a useful and entertaining application. Teachers from the School not only think Vocaliza is useful as a application for computer-aided speech therapy but as a educational application as well. They find the riddles game very useful for the training of cognitive skills of children with speech or without speech pathologies. Furthermore, the way of displaying the images in the sentences game is very similar to some systems of augmentative and alternative communication (AAC), which makes Vocaliza also very interesting as AAC system.

On the other hand, pupils from "Alborada" find the application amusing, but they admit having troubles to use the application on their own. However it is important to remark that those pupils are not used to work with computers at all and they also suffer cognitive disorders associated to their speech pathology.

7. CONCLUSIONS

As a result of this work, we have a totally functional application which aims to help the work of the speech therapists in three levels of the language: phonological, semantic and syntactic. The software is ready to be distributed at this moment, and is free to use for every speech therapist who may require it.

The evaluations by a group of speech therapists shows that Vocaliza is a useful tool for the training of children with speech disorders at several levels of the language. The speech therapists also evaluate positively the easiness of use of the software, altogether with the usefulness of the speech technologies implemented in Vocaliza for their work. In this way, all the requirements set at the beginning of the work have been completely fulfilled.

All these reviews and evaluations are very encouraging for us to keep working in this direction as we plan to keep improving the functionality of Vocaliza.

8. BIBLIOGRAPHY

[1] Protopapas A. Öster A-M, House D. and Hatzis A., "Presentation of a new eu project for speech therapy: Olp (ortho-logo-paedia)," TMH-QPSR vol. 44, Fonetik, 2002.

[2] Öster A. Kacic Z. Barczikay Z. Vicsi K., Roach P. and Sinka I., "Speco — a multimedia multilingual teaching and training system for speech handicapped children," Tech. Rep., Eurospeech, 6th Conference on Speech Communication and Technology, Interspeech, 1999.

[3] García Gómez et al., "Isaeus — speech training for deaf and hearing-impaired people," Tech. Rep., Eurospeech, 6th Conference on Speech Communication and Technology, Interspeech, 1999.

[4] "Harp – an autonomous speech rehabilitation system for hearing impaired people," Final report, HARP (TIDE project 1060), May 1996.

[5] J.L. Gauvain and C.H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994.

[6] E. Lleida and R.C. Rose, "Utterance verification in continuous speech recognition: Decoding and training procedures," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 2, pp. 126–139, 2000.

MULTIMODAL PERSON IDENTIFICATION IN A SMART ROOM

J.Luque, R.Morros, J.Anguita, M.Farrus, D.Macho, F.Marqués, C.Martínez, V.Vilaplana, J. Hernando

Technical University of Catalonia (UPC)
Jordi Girona, 1-3 D5, 08034 Barcelona, Spain
aluque@gps.tsc.upc.edu

ABSTRACT

In this paper we present a person identification system based on a combination of acoustic features and 2D face images. We address the modality integration issue on the example of a smart room environment. In order to improve the results of the individual modalities, the audio and video classifiers are integrated after a set of normalization and fusion techniques. First we introduce the monomodal acoustic and video identification approaches and then we present the use of combined input speech and face images for person identification. The various sensory modalities, speech and faces, are processed both individually and jointly. The result obtained in the CLEAR'06 Evaluation Campaign shows that the performance of the multimodal approach results in improved performance in the identification of the participants.

1. INTRODUCTION

Person identification consists in determining the identity of a person from a data segment, such as a speech, video segment, etc. Currently, there is a high interest in developing person identification applications in the framework of smart room environments. In a smart room, the typical situation is to have one or more cameras and several microphones. Perceptually aware interfaces can gather relevant information to recognize, model and interpret human activity, behaviour and actions. Such applications face an assortment of problems such a mismatched training and testing conditions or the limited amount of training data. In this work we present the audio, video and multimodal person identification techniques and the obtained results in the CLEAR'06 Evaluation Campaign inside the CHIL (Computers in the human interaction loop) project [1]. The CLEAR'06 Person Identification evaluation is a closed-set task, that is, all the possible speakers are known. Matched training and testing conditions and far-field data acquisition are assumed, as well as no a priori knowledge about room environment.

For acoustic speaker identification, the speech signals are parameterized using the Frequency Filtering (FF) [2] over the filter-bank energies, which is both computationally efficient and robust against noise. Next, in order to model the probability distribution of the parameters generated by each speaker, Gaussian Mixture Models (GMM) [3] with diagonal covariance are used.

In the case of visual identification, an appearance-based technique is used due to the low quality of the images. Face images of the same individual are gathered into groups. Frontal images within a group are jointly compared to the models for identification. These models are composed of several images representative of the individual. The joint recognition enhances the performance of a face recognition algorithm applied on single images. Individual decisions are based on a PCA [4] approach given that the variability of the users' appearance is assumed to be low and so are the lighting variations.

Multimodal recognition involves the combination of two or more human traits like voice, face, fingerprints, iris, hand geometry, etc. to achieve better performance than using monomodal recognition [5], [6]. In this work, a multimodal score fusion technique, Matcher Weighting with equalized scores, has been used. This technique has obtained an improvement for the correct identification rate on the closed-set $15/30$ seconds training and $1/5/10/20$ seconds testing conditions on the CLEAR'06 Evaluation task.

This paper is organized as follows: In sections 2 and 3 an overview of the audio and video algorithms and techniques is given. Section 4 presents the technique for multimodal fusion. Section 5 describes the evaluation scenario and the experimental results. Finally, section 6 is devoted to provide conclusions.

2. ACOUSTIC PERSON IDENTIFICATION

The speaker identification (SI) task consists in determining the identity of the speaker of a speech segment. In this task it is usually assumed that all the possible speakers are known. For this evaluation, recordings from 26 speakers using one microphone of an array have been provided. The first stage of current speaker recognition systems is a segmentation of the speech signal into regular segments. The speech signal is divided into frames of 30 ms at a rate of 10 ms. From each segment a vector of parameters

that characterizes the segment is calculated. In this work we have used the Frequency Filtering (FF) parameters [2]. These parameters are calculated as the widely used Mel-Frequency Cepstral Coefficients (MFFC) [7] but replacing the final Discrete Cosine Transform of the logarithmic filter-bank energies of the speech frame with the following filter:

$$H(z) = z - z^{-1} \tag{1}$$

These features have several interesting characteristics: they are uncorrelated, computationally simpler than MFCCs, have frequency meaning and they have generally shown an equal or better performance than MFCCs in both speech and speaker recognition. In order to capture the temporal evolution of the parameters the first and second time derivatives of the features are generally appended to the $\mathbf{o}_{FF}$, basic static feature vector. The so called delta coefficients [8] are computed using the following regression formula,

$$\Delta\mathbf{o}_t(i) = \frac{\sum_{\theta=1}^{\Theta} \theta\big(\mathbf{o}_{t+\theta}(i) - \mathbf{o}_{t-\theta}(i)\big)}{2\sum_{\theta=1}^{\Theta}\theta^2} \tag{2}$$

where $\Delta\mathbf{o}_t(i)$ is the delta coefficient i at time t computed in terms of the corresponding static coefficients $\mathbf{o}_{t-\Theta}$ to $\mathbf{o}_{t+\Theta}$. The same formula is applied to the delta coefficients with another window size to obtain acceleration coefficients. For each speaker that the system has to recognize, a model of the probability density function of the parameter vectors is estimated. These models are known as Gaussian Mixture Models (GMM) [3], which is a weighted sum of Gaussian distributions:

$$\lambda_j = \sum_{m=1}^{M} w_m N\big(\mathbf{o}, \mu_m, \mathbf{\Sigma}_m\big) \tag{3}$$

where λ_j represents the model of the j^{th} speaker, $\mathbf{o}$ is the vector of parameters being modeled, M is the number of Gaussian mixtures, w_m is the weight of the Gaussian m, and N is a Gaussian function of mean vector μ_m and covariance matrix $\mathbf{\Sigma_m}$. The parameters of the models are estimated from speech samples of the speakers using the well-known Baum-Welch algorithm.

Given a collection of training vectors, maximum likelihood model parameters are estimated using the iterative expectation-maximization (EM) algorithm. Twenty iterations are computed to estimate the client model. In the testing phase of a speaker identification system a set of parameter $\mathbf{O} = \{\mathbf{o}_i\}$ is computed from the testing speech signal. Next, the likelihood that each client model performs from the vector $\mathbf{O}$ is calculated and the speaker showing the largest likelihood is chosen,

$$s = \arg\max_j \big\{L\big(\mathbf{O}|\lambda_j\big)\big\} \tag{4}$$

where s is the recognized speaker and $L\big(\mathbf{O}|\lambda_j\big)$ is the likelihood that the vector $\mathbf{O}$ was generated by the speaker of the model λ_j.

3. VIDEO PERSON IDENTIFICATION

In this section, the Visual Person ID task is presented. We have developed for this task a technique for face recognition in smart environments. The technique takes advantage of the continuous monitoring of the scenario and combines the information of several images to perform the speaker recognition. Recognition is stand-alone, taking detection and tracking for granted. That is, the system is semi-automatic. Appearance based face recognition techniques are used given that the scenario does not ensure high quality images. As the visual identification evaluation is a close-set identification task, models for all individuals in the database are created off-line using two sets of video segments: the first one consists on one segment of 15 s per each individual in the database, while the second one consists on one segment of 30 s per individual.

The proposed system works with groups of face images of the same individual. For each test segment, face images of the same individual are gathered into a group. Then, for each group, the system compares such images with the model of the person.

We first describe the procedure for combining the information provided by a face recognition algorithm when it is applied to a group of images of the same person in order to, globally, improve the recognition results. Let $\{C\}_j = \{C_1, C_2, ..., C_S\}$ be the different models or classes stored in the (local or global) model database. S is the number of individual models. Let $\{x\}_i = \{\underline{x}_1, \underline{x}_2, ..., \underline{x}_M\}$ be a group of M probe images of the same person. Each model C_j contains N_j images, $\{y\}_n^j = \{\underline{y}_j^1, \underline{y}_j^2, ..., \underline{y}_{N_j}^j\}$ where N_j may be different for every class. We fix a decision threshold R_d so that $\underline{x}_i$ and $\underline{y}_n^j$ represent the same person if $d(\underline{x}_i, \underline{y}_n^j) < R_d$. If, for a given $\underline{x}_i$ the decision function is applied to every $\underline{y}_n^j \in C_j$, we can define the δ value of $\underline{x}_i$ relative to a class C_j, δ_{ij} as

$$\delta_{ij} = \#\big\{\underline{y}_n^i \in C_j \;;\; d(\underline{x}_i, \underline{y}_n^j) < R_d\big\} \tag{5}$$

That is, δ_{ij} counts the number of times that the face recognition algorithm matches $\underline{x}_i$ with an element of C_j. With this information, the δ-Table is built:

	$\mathbf{C_1}$	$\mathbf{C_2}$	...	$\mathbf{C_s}$
$\mathbf{X_1}$	δ_{11}	δ_{12}	...	δ_{1S}
$\mathbf{X_2}$	δ_{21}	δ_{22}	...	δ_{1S}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{X_M}$	δ_{M1}	δ_{M2}	...	δ_{MS}

Table 1. δ-Table

Relying on the table 1, the proposed technique allows to determine the class C_j which better represents the set of probe images $\{x_i\}$. Further information about this technique can be found in [9].

In this work, a PCA based approach [4] has been used. This way, the decision function is the Euclidean distance between the projections of x_i and y_n^j on the subspace spanned by the first eigenvectors of the training data covariance matrix:

$$d(\underline{x}_i, \underline{y}_n^j) = ||W^T \underline{x}_i - W^T \underline{y}_n^j|| \quad (6)$$

The XM2VTS database [10] has been used as training data for estimating the projection matrix and the first 400 eigenvectors have been preserved. Due to the images being recorded continuously using the corner cameras, face images can not be ensured to be all frontal. Mixing frontal and non-frontal faces in the same models can be quite a problem for face recognition systems. To avoid this situation, eye coordinates are used to determine the face pose for each image. Only frontal faces are used for identification. Note that, in our system, models per each person have been automatically generated, without human intervention. All images for a given individual in the training intervals are candidates to form part of the model. Candidate face bounding boxes are projected on the subspace spanned by the first eigenvectors of the training data covariance matrix WT. The resulting vector is added to the model only if different enough from the vectors already present in the model.

4. MULTIMODAL PERSON IDENTIFICATION

In a multimodal biometric system that uses several characteristics, fusion is possible at three different levels: feature extraction level, matching score level or decision level. Fusion at the feature extraction level combines different biometric features in the recognition process, while decision level fusion performs logical operations upon the monomodal system decisions to reach a final resolution. Score level fusion matches the individual scores of different recognition systems to obtain a multimodal score. Fusion at the matching score level is usually preferred by most of the systems.

Matching score level fusion is a two-step process: normalization and fusion itself [11], [12], [13], [14]. Since monomodal scores are usually non-homogeneous, the normalization process transforms the different scores of each monomodal system into a comparable range of values. One conventional affine normalization technique is z-score, that transforms the scores into a distribution with zero mean and unitary variance [12], [14].

After normalization, the converted scores are combined in the fusion process in order to obtain a single multimodal score. Product and sum are the most straightforward fusion methods. Other fusion methods are min-score and max-score that choose the minimum and the maximum of the monomodal scores as the multimodal score.

4.1. Normalization and Fusion Techniques

Scores must be normalized before being fused. One of the most conventional normalization methods is z-score (ZS), which normalizes the global mean and variance of the scores of a monomodal biometric. Denoting a raw matching score as *a* from the set *A* of all the original monomodal biometric scores, the z-score normalized biometric x_{zs} is calculated according to Eq. 7,

$$x_{zs} = \frac{a - \mu(A)}{\theta(A)} \quad (7)$$

where $\mu(A)$ is the statistical mean of A and $\theta(A)$ is the standard deviation.

Histogram equalization (HE) is a non linear transformation whose purpose is to equalize the variances of two monomodal biometrics in order to reduce the non linear effects typically introduced by speech systems. The HE technique matches the histogram obtained from the speaker verification scores and the histogram obtained from the face identification scores, both evaluated over the training data. The designed equalization takes as a reference the histogram of the scores with the best accuracy, which can be expected to have lower separate variances, in order to obtain a bigger variance reduction.

In Matcher Weighting (MW) fusion each monomodal score is weighted by a factor proportional to the recognition rate, so that the weights for more accurate matchers are higher than those of less accurate matchers. When using the Identification Error Rates (IER) the weighting factor for every biometric is proportional to the inverse of its IER. Denoting w^m and e^m the weigthing factor and the IER for the mth biometric x^m and M the number of biometrics, the fused score u is expressed as

$$u = \sum_{m=1}^{M} w^m x^m \quad (8)$$

where

$$w^m = \frac{\frac{1}{e^m}}{\sum_{m=1}^{M} \frac{1}{e^m}} \quad (9)$$

Before carrying out the fusion process, histogram equalization is applied over all the previously obtained monomodal scores. Since the best recognition results have been achieved in the acoustic recognition experiments, the histogram of the voice scores has been taken as a reference in the histogram equalization. After the equalization process, the weighting factors for both acoustic and face scores are calculated by using the corresponding Identification Error Rates, as in Eq. 9. Z-score normalization is also applied, and final fused scores are obtained by using Eq. 8.

5. EXPERIMENTS AND DISCUSSION

5.1. Experimental set-up

A set of audiovisual recordings of seminars and of highly-interactive small working-group seminars have been used. These recordings were collected by the CHIL consortium for the CLEAR 06 Evaluation. The recordings were done according to the "CHIL Room Setup" specification [1]. A complete description of the different recordings can be found in [15]. The figure 1 depicts a brief description of the UPC smart-room sensors and space conditions.

Data segments are short video sequences and matching far-field audio recordings taken from the above seminars. In order to evaluate how the duration of the training signals affects the performance of the system two training durations have been considered: 15 and 30 seconds. Test segments of different durations ($1, 2, 5, 10$ and 20 seconds) have been used during the algorithm development and testing phases. A total of 26 personal identities have been used in the recognition experiments.

	Number of segments	
Segment Duration	Development	Evaluation
1 sec	390	613
2 sec	182	0
5 sec	78	411
20 sec	26	178

Table 2. Number of segments for each test condition

Each seminar has one audio signal from the microphone number 4 of the Mark III array. Each audio signal has been divided into segments which contain information of a unique speaker. These segments have been merged to form the final testing segments of $1, 5, 10$ and 20 seconds (see Table 2) and training segments of 15 and 30 seconds. Video is recorded in compressed JPEG format, with different frame-rates and resolutions for the various recordings. Far-field conditions have been used for both modalities, i.e. corner cameras for video and Mark III microphone array for audio. In the audio task only one array microphone has been considered for both development and testing phases. In the video task, we have four fixed position cameras that are continuously monitoring the scene. All frames in the 1/5/10/20 seconds segments and all synchronous camera views can be used and the information can be fused to find the identity of the concerned person. To find the faces to be identified, a set of labels is available with the position of the bounding box for each person's face in the scene. These labels are provided each 1s. The face bounding boxes are linearly interpolated to estimate their position in intermediate frames. To help this process, an extra set of labels is provided, giving the position of both eyes of each individual each 200 ms. The metric used to benchmark the quality of the algorithms is the percentage of correctly recognized people from test segments.

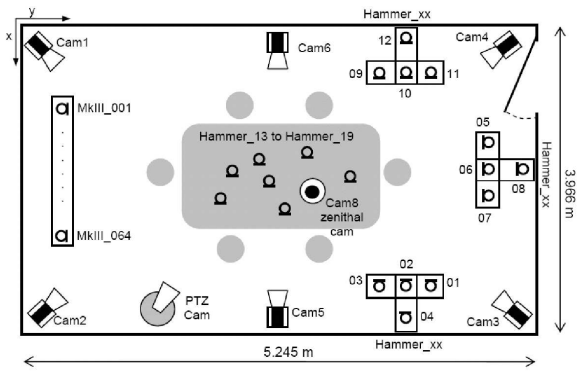


Fig. 1. *The UPC smart room setup*

5.2. Results

In this section we summarize the results for the evaluation of different modalities and the result improvement with the multimodal technique. Table 3 shows the correct identification rate for both audio and video modalities and the fusion identification rate obtained depending on the length of the used test files. Some improvements have been performed on the system since the CLEAR Evaluation, leading to better results than the ones presented in that. Related to acoustic identification task, it can be seen that the results, in general, are better as the segments length increases. Table 3 reports that for the different test segment lengths the recognition rate increases when more data is used to test the speaker models. Overall, using the 30 seconds training segments, an improvement of up to 6% in the recognition rate is obtained with respect to the case where 15 seconds segments are used. For the face identification evaluation, in general, these results show a low performance of the system. Results for the training set B (using a segment of 30 sec. to generate the models) show only a slight increase of performance with respect to training set A. It can also be seen that the results improve slowly as the segments length increases. The reasons for this low performance are manifold: First of all, the system uses only frontal faces to generate the models and for recognition. However, most of the face views found in the recordings are non frontal. Another reason for the low percentage of correctly identified persons is the low quality of the images. The need to cover all the space in the room with four cameras results in small images, were the person's faces are tiny. In the worst cases, face sizes are only 13x13 pixels.

The determination of the weighting factors for the multimodal fusion has been done by using the training signals of 30 seconds as a development set. The first 15 seconds have been used for training and the other 15 seconds for testing. The recognition results obtained in the evaluation for multimodal identification can also be seen in Table 3. Fusion results of both systems are

		Train A			Train B		
Duration	Segments	Speech	Video	Fusion	Speech	Video	Fusion
1	613	75.0 %	20.2 %	77.3 %	84.0 %	19.6 %	87.8 %
5	411	89.3 %	21.4 %	92.0 %	97.1 %	22.9 %	97.3 %
10	289	88.2 %	22.5 %	93.4 %	97.6 %	25.6 %	98.6 %
20	178	92.1 %	23.6 %	97.7 %	98.8 %	27.0 %	100.0 %

Table 3. Percentage of correct identification for both audio and video unimodal modalities and multimodal fusion. The first column shows the duration of test segments in seconds. The second one shows the number of tested segments. Train A and B are the training sets of 15 seconds and 30 respectively

also shown for the different lengths. Fusion correct identification rates are higher than the monomodal rates. The obtained fusion results outperform those obtained with both monomodal systems.

6. CONCLUSIONS

In this paper we have described two techniques for visual and acoustic person identification in smart room environments. A Gaussian Mixture Model of the distribution of the Frequency Filtering coefficients has been used to perform speaker recognition. For video, an approach based on joint identification over groups of images of a same individual using a PCA approach has been followed.

For the acoustic identification task, the results show that the presented approach is well adapted to the conditions of the experiments. For the visual identification task, the low quality of the images results in a low performance of the system. In this case, results suggest that identification should be performed combining more features other than frontal face bounding-boxes. To improve the obtained results, a multimodal score fusion technique has been used. Matcher Weighting with histogram equalized scores is applied to the scores of the two monomodal tasks. The results show that this technique can provide an improvement of the recognition rate in all train/test conditions.

Acknowledgements

This work has been partially sponsored by the EC-funded project CHIL (IST-2002 – 506909) and by the Spanish Government-funded project ACESCA (TIN2005 – 08852).

7. REFERENCES

[1] R. Stiefelhagen J. Casas, "Multi-camera/multi-microphone system design for continuous room monitoring,," in *CHIL Consortium Deliverable D4.1*, 2005.

[2] C. Nadeu, D. Macho, and J. Hernando, "Time and Frequency Filtering of Filter-Bank Energies for Robust Speech Recognition," in *Speech Communication*, 2001, vol. 34, pp. 93–114.

[3] D. A. Reynolds, "Robust text-independent speaker identifi cation using Gaussian mixture speaker models," in *IEEE Transactions ASSP*, 1995, vol. 3 N^o 1, pp. pp. 72–83.

[4] L.Sirovich M. Kirby, "Application of the Karhunen-Loeve procedure for the characterization of human faces," in *IEEE Trans. PAMI*, 1990, vol. 12, no. 1, pp. pp. 103–108.

[5] R.M. Bolle et al., "Guide to Biometrics," in *Springer*, 2004.

[6] R.Brunelli and D. Falavigna, "Person Identifi cation Using Multiple Cues," in *IEEE on PAMI*, 1995, vol. 17, No. 10, pp. 955–966.

[7] Mermelstein P. Davis S. B., "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," in *IEEE Transactions ASSP*, 1980, vol. Vol. 28, pp. pp. 357–366.

[8] Furui S., "Speaker independent isolated word recognition using dynamic features of speech spectrum," in *IEEE Transactions ASSP*, 1986, vol. No. 34, pp. pp.52–59.

[9] J.Cruz V.Vilaplana, C.Martinez and F.Marques, "Face Recognition Using Groups of Images in Smart Room Scenarios," in *ICIP*, 2006.

[10] K. Messer et al., "XM2VTSDB: The extended M2VTS Database," in *AVBPA*, 1999.

[11] N.A Fox et al., "Person Identifi cation Using Automatic Integration of Speech, Lip and Face Experts," in *ACM SIGMM WBMA*, 2003, pp. pp. 25–32.

[12] M. Indovina et al., "Multimodal Biometric Authentication Methods: A COTS Approach," in *MMUA*, 2003, pp. pp. 99–106.

[13] T. Chen S. Lucey, "Improved Audio-visual Speaker Recognition via the Use of a Hybrid Combination Strategy," in *The 4th International Conference on Audio- and Video- Based Biometric Person Authentication*, 2003.

[14] T. Tan Wang Yuan, Wangm Yunhong, "Combining Fingerprint and Voiceprint Biometrics for Identity Verifi cation: an Experimental Comparison," in *ICBA*, 2004, pp. pp. 663–670.

[15] Djamel Mostefa et al., "CLEAR Evaluation Plan v1.1," in `http://isl.ira.uka.de/~nickel/clear/downloads/chil-clear-v1.1-2006-02-2%1.pdf`, 2006.

EFFECTIVENESS OF SHORT-TERM PROSODIC FEATURES FOR SPEAKER VERIFICATION

Iker Luengo, Eva Navas, Inmaculada Hernáez, Jon Sanchez, Ibon Saratxaga, Iñaki Sainz

Aholab – Escuela Técnica Superior de Ingeniería.
Universidad del País Vasco – Euskal Herriko Unibertsitatea
Urkijo zum. z/g 48013 Bilbo
ikerl@bips.bi.ehu.es, eva.navas@ehu.es, inma.hernaez@ehu.es, jon.sanchez@ehu.es,
ibon.saratxaga@ehu.es, inaki@bips.bi.ehu.es

ABSTRACT

In this work a traditional MFCC based speaker verification system is combined with a prosody based one to determine whether simple short-term prosodic information is useful for improving current state-of-the-art ASV. The traditional speaker verification system based in spectral information has an EER of 3.85% when using 1024 mixtures. The prosody based system uses short-term intonation and energy information and achieves an EER of 23.93% with 128 mixtures. After applying LDA and fusing those scores, a final EER of 3.84% is achieved. This result does not show a significant improvement when compared with the result of the traditional speaker verification system.

1. INTRODUCTION

The use of long-distance transactions has become very popular in the last few years: shopping through the Internet, web based bank transactions, restricted access to secure areas in remote computers... All these systems need some kind of authentication procedure, in order to verify the users' identity. Most of them use password-type authentication, but passwords may be forgotten or stolen.

Nowadays biometric authentication is the best alternative. Biometric characteristics can't be lost nor forgotten, and are very difficult to imitate. This kind of authentication can already be seen in different applications: Laptops with fingerprint controlled access and hand geometry based access to certain buildings are some examples.

The growing interest on biometrics based automatic identification systems is reflected in the increase of biometric verification system contests, as the Fingerprint Verification Competition [1] or the NIST Speaker Recognition Evaluations [2], where new algorithms and methods are being proposed in order to improve current results. Automatic Speaker Verification (ASV) is not an exception. Most of the current implementations use some kind of spectral envelope features to parameterize the voice (MFCC, LPCC...), achieving a great performance [3] [4] [5]. But recent researches are trying to include prosodic information into the system, in order to reduce error rates.

Speech prosody refers to the intonation, energy and rate of the speech. It is well known that these features are characteristics of each person, so that they carry information about the speaker. Furthermore, prosody is uncorrelated with the spectral envelope shape. Therefore, supposedly adding prosodic features to the already used spectral features may lead to an improvement in the system's performance.

Most of the works in this area focus on the use of some kind of long-term prosodic information [6] [7] [8] and fusing the results with a state-of-the-art system. Others try to use per frame sampled short-term prosodic values like intonation and power curves [9] [10]. This last approach is very appealing, as the new features could be easily combined with the traditional cepstral coefficients.

This work focuses on determining whether simple short-term prosodic information is useful for improving current state-of-the-art ASV. In order to see this, a new ASV system is presented, which uses both spectral and prosodic features. The paper is organized as follows: First the constructed verification system is explained. Then, a description of the database used during the experiments is given, followed by a description of the experiments and their results. Finally, these results are commented.

2. DESCRIPTION OF THE VERIFICATION SYSTEM

2.1. Baseline system

The baseline system consists of a traditional GMM-UBM system [11] with Mel Frequency Cepstral Coefficients (MFCC) parameterization. More precisely,

This work has been partially funded by the Spanish Ministry of Science and Technology (TIC2003-08382-C05-03).

an 18 MFCC feature vector was extracted every 10 milliseconds. These vectors were augmented with the first and second order derivatives. Cepstral Mean Subtraction (CMS) was applied in order to reduce the channel effects [12].

Speaker models were trained by Maximum A Posteriori (MAP) adaptation of the previously trained Universal Background Model (UBM) [3]. Only the means were adapted, leaving the variances and weights unaltered. Last, as usual in this kind of systems, UBM and HNorm score normalization [13] were performed during the tests.

2.2. Prosodic system

In the prosody based system, energy and intonation related information was used. Separate models were created for prosody modelling in voiced and unvoiced regions, in order to handle the discontinuities in the intonation curve. The signal's power was estimated every 10 milliseconds using Hamming windowing of 30 milliseconds length. F0 was also estimated every 10 milliseconds, using a method based on cepstrum transformation and Viterbi algorithm. This method not only gives the estimated value of F0, but also decides whether a frame is voiced or not. Once the power and intonation curves were estimated, their first and second order derivatives were calculated, in order to take into account the dynamics of the features.

The voiced and unvoiced frames were separated to get two feature vector streams. Voiced frames were parameterized with five features (instantaneous F0, its first and second derivative, and the first and second derivative of the power), whereas unvoiced frames were parameterized with only two (the first and second derivatives of the power). The instantaneous power was discarded in both cases, as this value is more related to the channel gain than to the speaker's identity.

Using these two vector streams, two different models were trained for each speaker, using a traditional GMM-UBM scheme. First, two UBM models were developed, one for voiced and another one for unvoiced frames. Then, the speaker models were created from these by means of MAP adaptation.

During the test phase, and for each recording, two different scores were calculated, one for the voiced and another one for the unvoiced frame streams. UBM score normalization was applied to these scores before fusing them with the product rule. That is, the final score for the prosodic information is the product of the scores of the voiced and the unvoiced stream scores.

2.3. System fusion

In order to combine the results of both classifiers, a late fusion scheme was chosen: first the scores using both traditional and prosody based systems are calculated, and then, a final score is obtained combining these scores.

For this purpose Linear Discriminant Algorithm (LDA) [14] was selected. LDA is capable of finding the linear combination of scores that best separates user and impostor scores, but it must be trained over a validation set of speakers, before applying it to the final tests.

3. DATABASE DESCRIPTION

The AHUMADA database [15] was used for the experiments carried out in this work. This database consists of voice recordings of 103 male Spanish speakers, and was specifically recorded for the development of automatic speaker recognition systems. In fact, it was used during the NIST speaker recognition evaluation campaigns of 2000 and 2001 [2], together with an extension aimed to include feminine speakers.

Although the complete database contains both high quality microphone recordings and telephone speech recordings, only the later ones have been used for the experiments. This allows capturing the effects that channel distortion has on the system's behaviour. These telephonic recordings were carried out along three different sessions, and in each of them, the speakers used a different telephone handset.

- The first session (named T1), was recorded through an internal-routing call with a flat frequency response, so there is no spectral distortion in the signals.
- In the second session (T2), the recording was done through the speaker's home telephone, so the handset type remains unknown.
- In the third session (T3) each of the speakers used one of the nine handsets that were available in the recording laboratory. So that, for this session and for each speaker, the handset type that was used (carbon button or electret) is known.

Among the items that were recorded in each session, two were selected for the experiments: The read common text (the same text for all speakers and sessions) and the read specific text (a different text for each of the speakers and sessions). From now on, these items will be named C (for Common) and S (for Specific) respectively. All these recordings were sampled at 8 kHz, 16 bit per sample. The mean length of the selected items was about 65 seconds.

To sum up, three telephonic sessions of 103 male speakers, with two items per session were available for the experiments. This sub-corpus was divided as follows: 51 speakers were reserved to train the UBM models, 26 were used for the validation tests, and the rest were kept as users of the system. Every user was treated as an impostor for the rest of the users, which is equivalent to having 25 impostors. The speakers were randomly designated to one of the groups, in order not to bias the result.

# mix.	*2*	*4*	*8*	*16*	*32*	*64*	*128*	*256*	*512*	*1024*
Baseline	29.13	20.29	17.18	13.25	10.36	8.12	6.75	5.71	5.21	3.85
Voiced	35.47	33.21	29.88	28.96	27.78	25.32	23.93	24.03	24.74	23.93
Unvoiced	46.53	49.45	47.86	45.86	44.44	43.16	42.74	43.18	43.59	44.44
V+UV	35.27	32.48	29.49	28.77	27.33	25.64	23.93	24.89	24.53	24.14

Table 1. *%EER values for the baseline and prosody based systems.*

Speaker model training was done using tasks C and S from session T1. As this session was recorded without spectral distortion, the resulting models will be channel independent. Tasks C and S from session T3 were used for the development tests, because knowing which handset type was used makes it possible to determine, the HNorm coefficients for handset normalization. Finally, task S from session T2 was reserved for the final tests. In this way, these final tests were carried out with a handset that was unknown for the system, as it was not used before, neither in the training nor in the development phase. Furthermore, using only task S, also the contents of the recordings were new for the system.

For the training, the whole recordings were used. This gives about 130 seconds of training material for each speaker. For the tests, nine speech segments of 10 seconds each were extracted from each item, allowing a 50% overlap between two consecutive segments. So, for each speaker 9 validation test segments were available. This means that each user was tested against 25x9=225 impostor segments.

All this makes the design of these experiments very realistic, as in an environment in which the users make the recordings for training and development in the laboratory, but they try to gain access from their own home or office, using their own phone.

4. RESULTS

In a GMM system, the number of Gaussian mixtures is critical for the system's accuracy. In order to estimate the best number of mixtures for the models, different order GMMs were developed, from 2 up to 1024 mixtures, both for MFCC and prosody based systems. The final mixture number was selected in order to minimize the Equal Error Rate (EER) between False Rejection (FR) and False Acceptance (FA) rates. Table 1 shows the ERR curve for the trained systems. As expected, the baseline system achieves a better performance than prosody alone. It can also be seen that energy information of unvoiced frames does not help to the final result of the prosody based verification system.

According to the achieved EER values, 1024 and 128 mixture GMMs were selected for the baseline and prosody based systems respectively. After applying LDA and fusing the scores, a final EER of 3.84% was achieved. Figure 1 shows the DET curves [16] of the systems before and after applying the fusion. The DET curves for the baseline and the combined systems are very similar. In fact, both systems have the same EER.

5. CONCLUSIONS

As a result of these experiments, it can be seen that adding simple intonation and energy related short-term features does not improve the state-of-the-art results. Due to the great improvement achieved in the last few years in ASV (like various handset and likelihood normalization techniques), current spectral envelope based systems achieve a performance much greater than the prosody based ones (3.85% against 23.93% EER in these experiments).

This does not mean that prosody is not useful for ASV, as current researches have achieved some improvement using it [9] [10], but that prosody based ASV is far behind spectra based one. For example, just as HNorm came out to deal with handset variability in traditional systems, there is a need to solve the inter-session variability in prosody. There are still very interesting research areas to explore in this field.

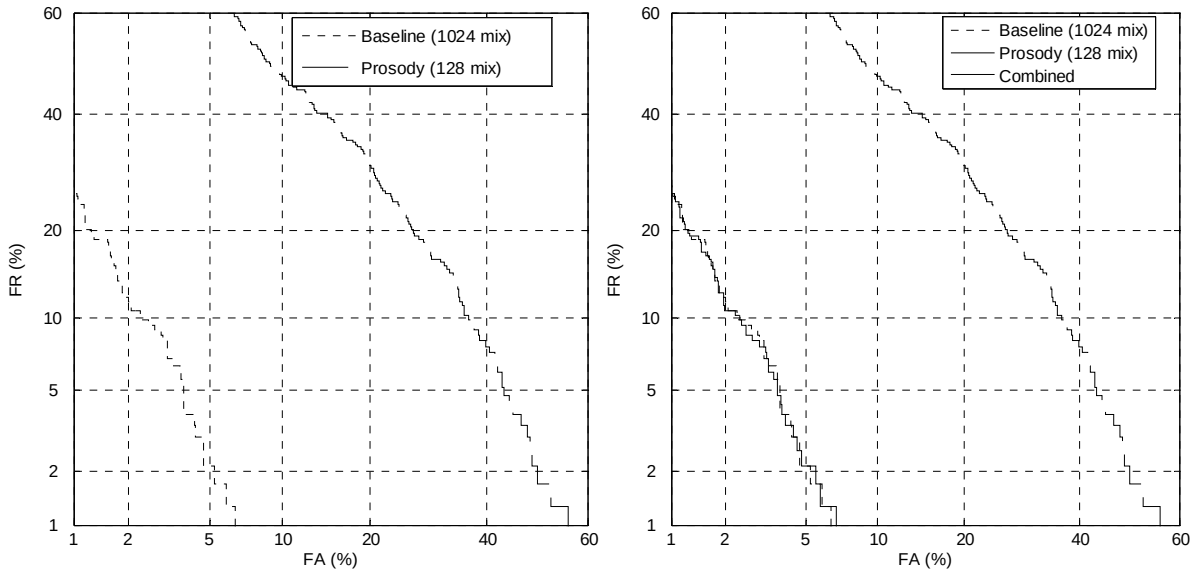


Figure 1. *DET curves for the baseline and prosody based systems (left) and the final combined system (right).*

6. REFERENCES

[1] FVC2006 web site: http://bias.csr.unibo.it/fvc2006/

[2] NIST Speaker Recognition Evaluations' web site: http://www.nist.gov/speech/tests/spk/

[3] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models," Digital Signal Processing, vol. 10, pp. 19-41, 2000.

[4] M. A. Przybocki and A. F. Martin, "NIST Speaker Recognition Evaluation Chronicles," presented at Odyssey, Toledo, España, 2004.

[5] A. F. Martin and M. A. Przybocki, "The NIST Speaker Recognition Evaluations: 1996-2001," presented at Speaker Odyssey 2001, Creta, Grecia, 2001.

[6] M. J. Carey, E. S. Parris, H. Lloyd-Thomas, and S. Bennett, "Robust Prosodic Features for Speaker Identification," presented at ICSLP, Philadelphia, EEUU, 1996.

[7] A. G. Adami, R. Michaescu, D. A. Reynolds, and J. J. Godfrey, "Modeling Prosodic Dynamics for Speaker Recognition," presented at ICASSP, Hong Kong, 2003.

[8] B. Peskin, J. Navratil, J. Abramson, D. Jones, D. Klusacek, D. A. Reynolds, and B. Xiang, "Using Prosodic and Conversational Features for High Performance Speaker Recognition," presented at ICASSP, Hong Kong, 2003.

[9] D. A. Reynolds, W. Andrews, J. P. Campbell, J. Navratil, B. Peskin, A. G. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Michaescu, J. J. Godfrey, J. Douglas, and B. Xiang, "SuperSID project final report," SuperSID project (http://www.clsp.jhu.edu/ws2002/groups/supersid/) 2002.

[10] M. Arcienega and A. Drygajlo, "A Bayesian Network Approach for Combining Pitch and Spectral Envelope Features for Speaker Verification," presented at COST275 workshop, Roma, Italia, 2002.

[11] F. Bimbot, J. F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, and D. A. Reynolds, "A Tutorial on Text-Independent Speaker Verification," EURASIP Journal on Applied Signal Processing, vol. 4, pp. 430-451, 2004.

[12] C. Barras and J. L. Gauvain, "Feature And Score Normalization for Speaker Verification of Cellular Data," presented at ICASSP, Hong Kong, 2003.

[13] R. Auckenthaler, M. J. Carey, and H. Lloyd-Thomas, "Score Normalization for Text-Independent Speaker Verification Systems," Digital Signal Processing, vol. 10, pp. 42-54, 2000.

[14] J. B. Kennedy and A. M. Neville, *Basic Statistical Methods for Engineers and Scientists*, Harper & Row, New York, 1986.

[15] J. Ortega-García, J. González-Rodríguez, and V. Marrero-Aguiar, "AHUMADA: A Large Speech Corpus in Spanish for Speaker Characterization and Identification," Speech Communication, vol. 31, pp. 255-264, 2000.

[16] A. F. Martin, G. R. Doddington, T. Kamm, M. Ordowski, and M. A. Przybocki, "The DET Curve in Assessment of Detection Task Performance," presented at Eurospeech, Rhodes, Grecia, 1997.

DESCRIPCIÓN BIOMÉTRICA DE LA VOZ A PARTIR DE LA ESTRUCTURA BIOMECÁNICA DE LA CUERDA VOCAL

P. Gómez, R. Fernández, A. Álvarez, L. M. Mazaira, V. Rodellar, J. I. Godino

Grupo de Informática Aplicada al Procesado de Señal e Imagen, Universidad Politécnica de Madrid, Campus de Montegancedo, s/n 28660 Boadilla del Monte, Madrid

e-mail: pedro@pino.datsi.fi.upm.es

RESUMEN

La parametrización del comportamiento dinámico de la cuerda vocal puede ser útil en diferentes campos del procesado de la voz, como en síntesis, detección de patologías del aparato fonador, o en identificación y verificación del locutor. En el presente trabajo trata de la determinación de los parámetros biomecánicos de la cuerda vocal en forma no invasiva, generando estimaciones de las masas, elasticidades y pérdidas asociadas al modelo de cuerda vocal compuesto por el cuerpo y la cubierta elástica de la misma (lamina propria). Se estimaciones de los valores efectivos de dichos parámetros generados para una base de datos de individuos normofónicos. Estos resultados muestran una discrepancia de un orden de magnitud respecto a los valores clásicamente utilizados en modelado directo de la cuerda vocal. Ésta se explicaría mediante el examen del patrón de vibración longitudinal de la cuerda. La parametrización presentada produce descriptores biométricos aplicables en logopedia, rehabilitación y autentificación.

1. INTRODUCCIÓN[1]

La parametrización de la onda glótica resulta de gran utilidad en múltiples campos, como en detección e identificación de la patología del aparato fonador [9][3], en el modelado de la cuerda vocal [1], en la síntesis de la voz [11], en la descripción biométrica e identificación del locutor [12], etc. La parametrización directa de la señal de voz íntegra contiene información propia de la biometría del locutor (estructura y dimensiones del tracto vocal, biomecánica de las cuerdas vocales, etc.), pero está sujeta a variaciones fuertemente condicionadas por el proceso articulatorio. Esto significa que rasgos propios de la biometría y de la fonética acústica aparecerán intermezclados. En el escenario propio del reconocimiento de voz es conveniente independizar los mismos (reconocimiento independiente del locutor). En el contexto de caracterización biométrica es posible que también los rasgos fonéticos aporten información sobre la personalidad del locutor, pero un tratamiento individualizado de ambos niveles podría generar resultados más precisos si los mismos se integran de forma adecuada, siguiendo las tendencias actuales en identificación y verificación del locutor [2]. Parece procedente realizar una caracterización de la biometría del sistema de producción de onda glótica donde se haya separado la influencia del tracto vocal, en la creencia de que la parametrización por separado de los rasgos biométricos de tracto vocal y de pulso glótico pueden describir mejor las características de un locutor dado. Por dicha razón se ha desarrollado una metodología para determinar la biomecánica de la cuerda vocal a partir de la onda glótica obtenida por eliminación de la influencia del tracto vocal. En trabajos previos [8][4] se ha demostrado que la biomecánica de las cuerdas vocales (masas, resortes y pérdidas) es directamente responsable de la distribución espectral de potencia de los correlatos glóticos dinámicos que pueden derivarse de la traza de voz, como es la excitación glótica (*fuente glótica*). Bajo ciertas condiciones es posible obtener estimaciones directas de estos parámetros suponiendo que las distribuciones espectrales de los correlatos glóticos están directamente relacionadas con ciertas funciones de la dinámica de la cuerda vocal [6]. El desbalance entre cuerdas, y sus desviaciones con respecto a las estimaciones obtenidas a partir de una muestra de la población general de locutores puede estar directamente relacionado con la patología de la glotis [7]. En el presente trabajo se proporcionan descripciones y estimaciones de los parámetros biomecánicos de las estructuras de cuerpo-cubierta de la cuerda vocal. Se aportan resultados obtenidos a partir de una base de datos que incluye individuos adultos y adolescentes de ambos sexos. Las estimaciones generadas se comparan con los valores utilizados en modelado clásico de la cuerda vocal [14]. Las discrepancias entre los valores teóricos y los obtenidos por este método de estimación ponen en evidencia la necesidad de revisar la forma en que la

[1]Este trabajo ha sido subvencionado por los proyectos TIC2002-2273, TIC2003-08756 y TEC2006-12887-C02-00 del Plan Nacional de I+D+I, Ministerio de Educación y Ciencia.

cuerda vocal es considerada en el modelado clásico de la misma.

2. MODELADO DE LA CUERDA VOCAL

El presente estudio se basa en el conocido modelo de *3 masas* de Story y Titze [14] que se muestra en la Figura 1.a, donde el cuerpo de la cuerda (formado por los músculos responsables de tensar la misma) aparece representado por las masas M_{bl} y M_{br} (para las cuerdas izquierda y derecha, respectivamente), y por los resortes K_{bl} y K_{br} que describen el comportamiento elástico de los tejidos del cuerpo de la cuerda, ligando la misma a las paredes del cartílago tiro-aritenoide.

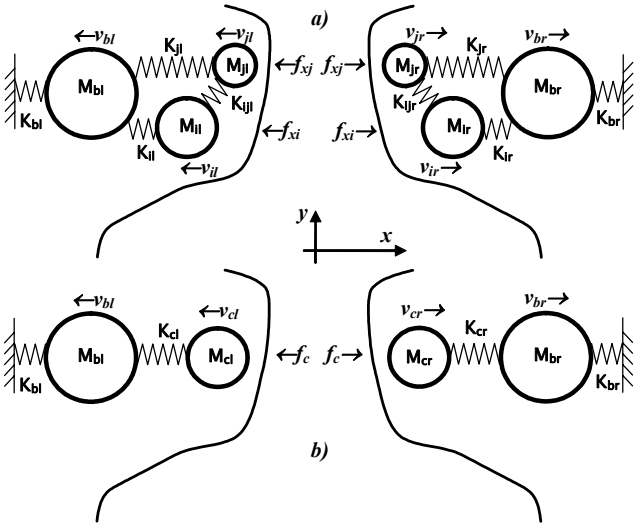


Figura 1. *a) Modelo de **3 masas** de las cuerdas vocales, donde se muestran los parámetros biomecánicos y las variables dinámicas involucradas en el mismo. b) Modelo de **2 masas** derivado del anterior donde las masas de la cubierta se han considerado rígidamente ligadas.*

La cubierta de la cuerda (compuesta por una estructura epitelial en capas) se representa por un par de masas por cuerda dado por $M_{il,r}$ y $M_{jl,r}$ ligadas a las masas $M_{bl,r}$ del cuerpo de la cuerda por medio de resortes $K_{il,r}$ y $K_{jl,r}$, interacopladas por los resortes $K_{ijl,r}$. Las masas y resortes son parámetros discretos del modelo asociados a estructuras de tipo puntual. En una simplificación dinámica solo se permite que el sistema se mueva a lo largo del eje x, siendo $f_{xi,j}$ las fuerzas que actúan sobre cada masa (externas, debido a la presión transglótica), siendo $v_{il,r}$, $v_{jl,r}$ y $v_{bl,r}$ las velocidades lineales de las respectivas masas. Si el estudio fuese orientado a poner de manifiesto el papel jugado por la cubierta de la cuerda, en relación con el fenómeno de vibración de la misma conocido como *onda mucosa*, los resortes de acoplamiento entre masas de la cubierta presentarían valores acotados de rigidez, admitiendo que ambas masas pudiesen vibrar en forma relativamente libre. Pero como el presente estudio desea poner de relieve el papel relativo jugado entre las estructuras del cuerpo y de la cubierta de la cuerda, se supondrá que los resortes que acoplan las masas de cubierta deben verse como ligaduras fuertes (barras de rigidez ilimitada), de modo que la estructura completa puede reducirse al modelo más simple de *2 masas* que se muestra en la Figura 1.b, donde

$$M_{cl,r} = M_{il,r} + M_{jl,r}$$
$$K_{cl,r} = K_{il,r} + K_{jl,r} \qquad (1)$$

De este modo el estudio de la dinámica de la cuerda vocal puede dividirse en dos, uno dedicado al cuerpo de la cuerda, y otro a la cubierta de la misma. La metodología de procesado de la señal de voz utilizada en el presente estudio se basa en la obtención de la fuente glótica, una vez eliminado el tracto vocal de la traza de voz por filtrado inverso, según [1], y en la separación de la onda glótica en dos componentes: una señal de duración de un ciclo de fonación completo (intervalo entre dos cierres glóticos consecutivos, incluyendo la fase de cierre y la de apertura) denominada por Titze [16] como la *onda acústica promedio*, y una componente que conserva los contenidos de alta frecuencia, que se denomina *componente dinámica de cubierta* o también *correlato de onda mucosa*.

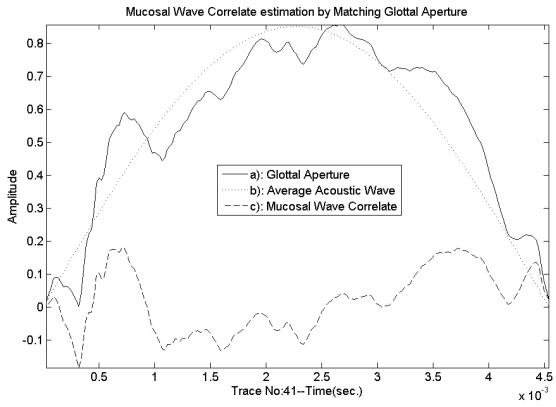


Figura 2. *Separación de la fuente glótica (línea continua) en la onda acústica promedio (línea de puntos) y la **componente dinámica de cubierta** (línea de trazos).*

La *fuente glótica* puede relacionarse con la *apertura glótica*, considerada como la distancia promedio entre las masas de cubierta (consideradas puntuales) durante un ciclo de fonación, y la *componente dinámica de cubierta* asociada con el movimiento relativo de las masas de la cubierta $M_{cl,r}$ con respecto a las masas del cuerpo $M_{bl,r}$ (Figura 1.b). Para llevar a cabo dicha separación se han ensayado diferentes algoritmos, entre ellos el filtrado paso bajo [5]. En la Figura 2 se muestran ambas componentes de la fuente glótica después de su separación para un ciclo de fonación completo.

3. ESTIMACIÓN DE LOS PARÁMETROS DE LA CUERDA VOCAL

En el proceso de estimación de los parámetros del cuerpo y la cubierta de la cuerda vocal se utiliza el equivalente electrodinámico representado por el sistema dado en la Figura 3

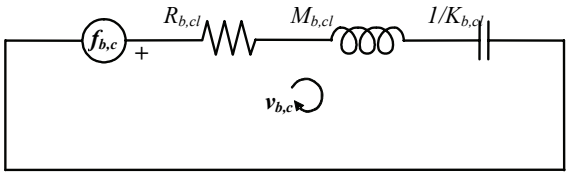


Figura 3. *Equivalentes electrodinámicos del cuerpo (b) y la cubierta (c).*

Se supondrá que la densidad espectral de potencia de los componentes de la *fuente glótica* se hallan relacionados (salvo por un factor de escala) con el módulo al cuadrado de la admitancia de entrada de cada equivalente electromecánico.

3.1. Estimación de los parámetros del cuerpo de la cuerda

La estimación de los parámetros del cuerpo de la cuerda a partir de la *onda acústica promedio* no resulta complicada, toda vez que su densidad espectral de potencia es suave y predecible, ya que la mayoría de la variabilidad presente en la *fuente glótica* reside en la *componente dinámica de cubierta*. El proceso de estimación se resume en los siguientes pasos

$$M_{bl,r} = \frac{\omega_2}{\omega_2^2 - \omega_r^2}\left[\frac{T_r - T_2}{T_r T_2}\right]^{1/2} \qquad (2)$$

siendo ω_r la frecuencia de resonancia característica del sistema electromecánico equivalente, dada por

$$\omega_r^2 = \frac{K_{bl,r}}{M_{bl,r}} \qquad (3)$$

donde el módulo al cuadrado de la admitancia de entrada viene dado por

$$T(\omega) = \frac{1}{\left[R_{bl,r}^2 + \varpi^2 M_{bl,r}^2\right]^2} \qquad (4)$$

siendo ϖ la desviación de frecuencia relativa al punto de resonancia, dada por

$$\varpi = \frac{\omega^2 - \omega_r^2}{\omega} \qquad (5)$$

reflejando T_r y T_2 los valores de la admitancia dada en (4) para el punto de resonancia y para su frecuencia doble

$$T_r = T(\omega = \omega_r) = \frac{1}{R_{bl,r}^2}$$
$$T_2 = T(\omega = 2\omega_r) \qquad (6)$$

El procedimiento de estimación debe detectar el valor del *pitch*, que se utiliza para estimar ω_r. La determinación de T_r y T_2 se lleva a cabo sobre la densidad espectral de potencia de la *onda acústica promedio*. Ello conduce a la determinación de las pérdidas a partir de (6), y de éstas a las masas a partir de (2) y de las tensiones por medio de (3). Como avance a los resultados que se presentan en la sección 4, en la Figura 5 (plantillas *35-37*) se pueden ver los resultados de la estimación de los parámetros biomecánicos para una traza de voz de *0,2 sg.* de duración, correspondiente a una fonación de la vocal */a/* por el sujeto *198* (ver la Tabla 1) que se ofrecen junto con su desviación estadística para dar fe de la fiabilidad de las estimaciones.

3.2. Estimación iterativa de los parámetros de la cubierta de la cuerda

La estimación de los parámetros de la *componente dinámica de cubierta* se lleva a cabo en dos partes. Primero se realiza una estimación de tipo grueso siguiendo los pasos (2)-(6) como en el caso de la *onda acústica promedio* anteriormente referido. Como la densidad espectral de potencia de la *componente dinámica de cubierta* es más irregular, y ofrece mayor variabilidad, las estimaciones de los pares [ω_r, T_r] y [ω_n, T_n] resultan menos robustas, por lo que se lleva a cabo un refinamiento iterativo a posteriori. Éste se basa en la minimización de la siguiente función de coste

$$L = \int_0^{\omega_s} \left[T_u(\omega) - T_i(\omega)\right]^2 d\omega \qquad (7)$$

donde ω_s es la *frecuencia de Nyquist*, T_u es la densidad espectral de potencia de la *componente dinámica de cubierta*, y T_i es la estimación iterativa de (4). Esta función muestra un mínimo único, que puede obtenerse, bien por medio del establecimiento de una expresión funcional (lo que no es sencillo dado el carácter no lineal de la misma) o por medio de refinamientos paso a paso de la estimación utilizando la técnica de descenso por gradiente, dada por

$$M_{bl,r}^i = M_{bl,r}^{i-1}\left[1 - 4\int_0^{\omega_s}(T_u - T_i)T_i^2\varpi^2 d\omega\right] \qquad (8)$$

donde *i* es el índice de iteración, repitiéndose ésta hasta que (7) se reduzca a un valor tan pequeño como se desee. Los resultados de dicha iteración para un ciclo de fonación correspondiente a la misma muestra de voz estudiada anteriormente se muestran en la Figura 4.

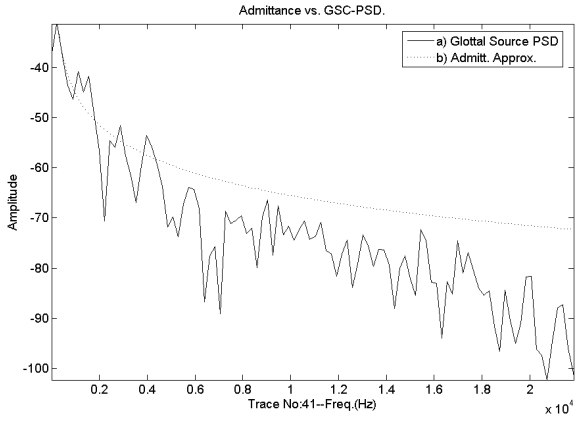


Figura 4. *Ajuste de la admitancia del modelo equivalente de* 1 masa *(línea de puntos) sobre la densidad espectral de potencia (en dB) de la* componente dinámica de cubierta *(línea continua) tras la última iteración para la estimación de la masa.*

Los resultados de estimar los parámetros biomecánicos de la cubierta para cada ciclo de fonación contenido en la traza de voz referida se muestran en la Figura 5 (plantillas 41-43).

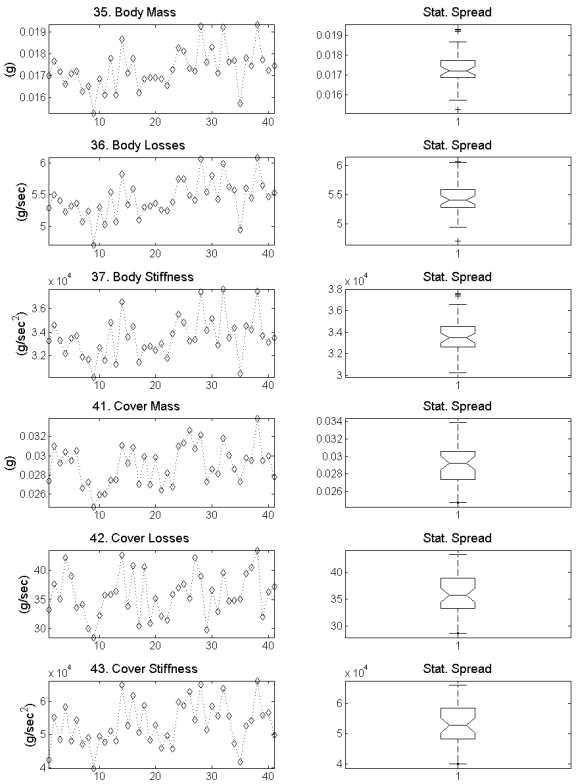


Figura 5. *Columna izquierda: Estimaciones de los parámetros de masa, pérdidas y tensión para el cuerpo (35-37) y la cubierta (41-43) de la cuerda del sujeto 198 en cada ciclo de fonación (eje x). Columna derecha: dispersión estadística de cada parámetro.*

El refinamiento de la estimación por iteraciones sucesivas puede suponer variaciones de entre un 20% y un 50% del valor medio de la estimación total.

4. RESULTADOS Y DISCUSIÓN

Para estudiar la dispersión de los parámetros biomecánicos típicos en una población con voz sana, se seleccionaron *100 sujetos normofónicos*, de los que se obtuvieron diferentes grabaciones. La mitad de los sujetos grabados eran adultos, con edades comprendidas entre 18 y 50 años, y la otra mitad, adolescentes de entre 12 y 18 años. Los locutores se hallaban igualmente distribuidos por sexos. Se realizó la determinación de su condición de normofonía a partir de muestras electroglotográficas y videoendoscópicas, así como por una valoración subjetiva de su índice GRABS [10]. Los resultados que se presentan corresponden a un subconjunto elegido al azar correspondiente a *24 sujetos*, de los cuales *16* son adultos, y *8* son adolescentes (la distribución por sexos es paritaria en ambos casos). Las características de cada sujeto se muestran en la Tabla 1.

T\P	Char	M_b	R_b	K_b	M_c	R_c	K_c
191	YF	0.00209	0.626	$3.78x10^3$	0.00220	2.24	$3.24x10^3$
192	AM	0.00803	1.160	$3.33x10^3$	0.00610	4.26	$2.27x10^3$
193	YF	0.00521	1.370	$7.57x10^3$	0.00477	6.50	$7.80x10^3$
194	YM	0.01040	2.230	$9.53x10^3$	0.00797	8.36	$8.54x10^3$
195	YM	0.00563	1.380	$6.73x10^3$	0.00367	4.42	$4.69x10^3$
196	AM	0.00463	0.807	$2.80x10^3$	0.00290	2.49	$2.10x10^3$
197	YF	0.00396	1.270	$8.01x10^3$	0.00560	7.00	$1.50x10^4$
198	YM	0.01730	5.440	$3.36x10^4$	0.02900	35.9	$5.33x10^4$
199	AM	0.00584	0.855	$2.52x10^3$	0.00445	3.38	$3.92x10^3$
19A	YF	0.00451	1.180	$6.74x10^3$	0.00406	6.28	$1.16x10^4$
19B	AM	0.01180	2.100	$7.40x10^3$	0.01060	8.61	$9.59x10^3$
19C	AM	0.00950	1.410	$4.10x10^3$	0.00695	4.65	$2.52x10^3$
19D	YM	0.00910	2.990	$1.91x10^4$	0.00811	10.6	$1.49x10^4$
19E	AM	0.00542	0.852	$2.80x10^3$	0.00381	2.77	$1.70x10^3$
19F	AM	0.00674	1.180	$4.11x10^3$	0.00526	4.81	$4.52x10^3$
1A0	AM	0.00543	0.839	$2.49x10^3$	0.00362	2.39	$1.25x10^3$
1A5	AF	0.00729	1.950	$1.06x10^4$	0.00793	7.68	$8.73x10^3$
1AA	AF	0.00499	1.380	$8.01x10^3$	0.00400	6.11	$1.11x10^4$
1AB	AF	0.00523	1.440	$7.88x10^3$	0.00465	5.61	$7.29x10^3$
1AE	AF	0.00513	1.580	$9.82x10^3$	0.00766	9.84	$1.46x10^4$
1B1	AF	0.00406	1.180	$6.73x10^3$	0.00591	7.19	$8.05x10^3$
1B9	AF	0.00420	1.230	$7.18x10^3$	0.00396	5.30	$7.21x10^3$
1BF	AF	0.00436	1.190	$6.40x10^3$	0.00390	3.96	$4.58x10^3$
1C6	AF	0.00419	1.140	$6.19x10^3$	0.00227	2.94	$2.73x10^3$

Tabla 1. *Parámeros biomecánicos para el cuerpo y la cubierta obtenidos a partir de un conjunto de 24 sujetos normofónicos. Características del locutor: AM – adulto masculino, AF – adulto femenino, YM – adolescente masculino, YF – adolescente femenino.*

Las grabaciones incluyeron tres realizaciones de la vocal */a/* de unos *3 sg.* de duración cada una, de las cuales se seleccionaron fragmentos de *0,2 seg.* de duración para el análisis. Estos fragmentos se procesaron para obtener la *onda acústica promedio* y la *componente dinámica de cubierta*, a partir de las cuales se estimaron los parámetros biomecánicos de *cuerpo* y *cubierta* para cada ciclo de fonación. Las estimaciones para cada parámetro biomecánico correspondientes al sujeto *198* se muestran en secuencia en la columna izquierda de la Figura 5, evaluándose sus promedios y desviaciones típicas, que se adjuntan en las plantillas de la columna derecha de la misma figura. Los promedios resumidos para los *24* sujetos de la base de datos se dan en unidades *cgs* en las columnas 3-5 (cuerpo) y 6-8 (cubierta) de la Tabla 1. Puede verse a partir de las estimaciones de dichos parámetros biomecánicos que las diferencias fisiológicas en la masa dinámica del cuerpo no son relevantes en sí mismas, pues si bien las masas tienden a ser algo superiores en el caso masculino que en el femenino, y en adultos que en adolescentes, el máximo estimado no corresponde a un individuo adulto masculino (como cabría suponer) sino a un adolescente masculino (sujeto *198*). La distribución resulta ser mucho más relevante en el caso de las tensiones de cuerpo, donde en general, la voz adolescente y femenina tiende a ser más tensa que la adulta y masculina. Respecto a las masas de cubierta, se puede decir que se hallan casi siempre por debajo de las de cuerpo, salvo en el caso del sujeto *198*, donde las masas de cuerpo y cubierta son de *17,3 mg*. y *29 mg*. respectivamente. Esto mismo se observa para los sujetos *191*, *197, 1A5, 1AE,*

y *1B1*, aunque las diferencias no son tan acusadas. Por lo que se refiere a las tensiones de cubierta, se puede ver que en muchos casos pueden llegar a superar las de cuerpo (por ejemplo en los sujetos *193, 197, 198, 199, 19A, 19B, 19F, 1AA, 1AE, 1B1* y *1B9*). El parámetro de elasticidad es el responsable del ajuste de la frecuencia de resonancia, ya que es ajustable mediante el cambio de posición gestual del proceso tiro-aritenoide, y por tanto depende de la altura tonal de la fonación. En todos los casos estudiados el factor de pérdidas asociado a la cubierta es mayor que el observado para el cuerpo de la cuerda. Ello podría deberse a que la mayor parte de la turbulencia y el calentamiento por choque de cuerdas se generan sobre y alrededor de la cubierta.

4.1. Unicidad de las estimaciones

Uno de los problemas subyacentes en este tipo de metodología es el de la unicidad de las estimaciones de masa y elasticidad, ya que el conjunto $\{M,K\}$ acepta infinitos pares de valores mientras cumplan que la relación $K/M = \omega_r^2$ arroje un mismo valor. El factor de pérdidas, admite una referenciación más estricta, ya que viene dado por el ancho de banda relativo de las funciones de densidad espectral de potencia, medido como la altura relativa del pico de resonancia respecto a su anchura. Por otro lado, se pueden formular modelos de pérdidas basados en el proceso de aportación energética debido a la presión transglótica y a la disipación debida al choque y a la turbulencia generada en los períodos próximos al cierre glótico para cada ciclo de fonación, que predicen con bastante exactitud el comportamiento promedio de las pérdidas a lo largo de varios ciclos de fonación vecinos. En base a este hecho, se puede normalizar la densidad espectral de potencia en cada ciclo con referencia a la frecuencia de resonancia y la duración del intervalo de fonación, que permite obtener estimaciones bastante realistas del factor de pérdidas, siendo éste el punto de referencia para determinar el factor de escala de cada ciclo según (6).

4.2. Consistencia de las estimaciones

La afirmación de la validez de las estimaciones no es simple ya que obtener estimaciones directas de los valores de los parámetros asociados a las cuerdas vocales de un sujeto dado *in vivo* es poco menos que imposible. Estimaciones directas de la dinámica de las cuerdas en vibración, utilizando videoendoscopía [15] son posibles para cierto tipo de sonidos (vocales abiertas, sobre todo), a partir de las cuales se pueden establecer procedimientos de estimación similares al mostrado, aunque con los mismos problemas de autentificación, a los que habría que añadir el carácter semi-invasivo de la técnica endoscópica. Pueden diseñarse otros métodos basados en videoendoscopía para estimar la masa de las cuerdas a partir de imágenes de las mismas durante la fonación, aunque en este caso se tropieza con el problema de que la masa estática (cuerda en reposo) puede diferir sustancialmente de la masa dinámica de la misma (parámetro derivado de las estimaciones dinámicas entre fuerzas y aceleraciones), pues ésta última es solo una fracción de la masa estática, ya que no se puede afirmar que toda la estructura tisular de la cuerda se halle igualmente involucrada en la vibración (en otras palabras, las cuerdas vocales en vibración no son asimilables a las cuerdas libres pinzadas por sus extremos, como el caso de las de una guitarra, pues se hallan conectadas lateralmente en toda su longitud sobre paredes semirrígidas de tipo cartilaginoso por medio de tejido conjuntivo que las une al proceso tiro-aritenoide, lo cual limita grandemente la contribución de parte del cuerpo de la cuerda a la vibración). Esto se muestra con claridad para una cuerda idealizada de sección constante tipo columna, como la que se muestra en la Figura 6.

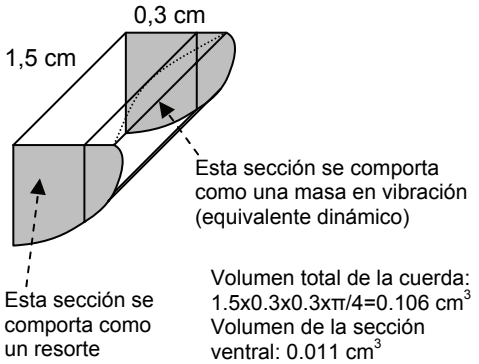


Figura 6. Diferencia entre masa estática y dinámica de la cuerda vocal.

En efecto, a partir de imágenes obtenidas por videoendoscopía estroboscópica se puede apreciar cómo una cuerda sana de las características de la expuesta vibra de forma desigual en toda su longitud, con máxima elongación hacia su zona ventral central, mientras que los extremos apenas vibran. La elongación en la zona central puede ser de alrededor de *1 mm*, mientras que la zona correspondiente al cuerpo que no participa en la vibración, sino que sufre compresiones y expansiones puede extenderse en aproximadamente *2 mm* de los *3 mm* típicos de dimensión transversal para la cuerda. Si se realizan estimaciones de la masa total de una cuerda de estas características se obtienen valores de alrededor de *100 mg*. (ver Figura 6), mientras que la masa efectiva de la sección vibrante sería de unos *10 mg*. Dependiendo del uso de la cuerda realizado por cada locutor en su gesto fonador propio, el porcentaje de cuerda implicado podría moverse entre un *10%* y un *30%* del total de la masa estática de la cuerda, lo que incluiría cuerpo y cubierta. Estas estimaciones pueden compararse contra los valores utilizados por los investigadores en modelado directo de la cuerda vocal. Por ejemplo, Story usa en su modelado masas de *0,058 g* (cuerpo), *0,082 g* (cubierta supraglótica) y *0,094 g* (cubierta infraglótica) y parámetros de rigidez de *112.250 dinas/cm*, *157.140 dinas/cm* y *203.660*

dinas/cm respectivamente para un modelo de *3 masas* [13], con cuerdas similares a las de la Figura 6 (*1,5 cm* de largo por *0,3 cm* de perfil). Teniendo en cuenta estas consideraciones puede verse que las estimaciones dadas en la Tabla 1 resultan ser más consistentes que los valores utilizados en simulación. Así, la masa total estática utilizada por Story en una sola cuerda sería de *0,234 g* (añadiendo la de cuerpo y las dos de cubierta), lo que sobrepasa en más de dos veces la masa que le correspondería de acuerdo con las dimensiones de la cuerda. Al sobredimensionar la masa en este tipo de modelado, habrá que sobredimensionar la tensión que sufre la cuerda, con lo que se utilizan valores que difícilmente serían soportables por el tejido fisiológico sin sufrir un deterioro notorio. Si se asume que los valores de las masas dinámicas son solo una parte reducida de la masa estática, las estimaciones de las tensiones adoptan valores más acordes con la realidad, compatibles con una mayor resistencia al deterioro. En cualquier caso, la tensión de la cuerda es una de las causas de la degeneración de la misma y de la aparición de patología, por lo que su estimación directa resulta de la mayor importancia. El seguimiento de dicho valor en diferentes situaciones gestuales (como en canto, por ejemplo) permitirá a foniatras y logopedas realizar una mejor labor diagnóstica, terapéutica y rehabilitadora.

5. CONCLUSIONES

El trabajo presentado describe una posible metodología para resulver un problema inverso de interés en el tratamiento de la voz: estimar los parámetros biomecánicos de la cuerda vocal a partir de la distribución espectral de potencia de la onda glótica obtenida por filtrado inverso de la voz. Los resultados para un conjunto de locutores de muestra con comportamiento normofónico indican que los rasgos fisiológicos del locutor juegan un papel menos relevante en la distribución de las estimaciones de las masas de las cuerdas que en las de sus tensiones. Por otro lado se han estimado valores para las masas un orden de magnitud por debajo de los utilizados clásicamente en el modelado directo de las cuerdas vocales. Una revisión cuidadosa de la estructura de una cuerda vocal típica y de la implicación de la misma en la vibración apoyaría resultados más en consonancia con los estimados por el procedimiento propuesto que con los utilizados en modelado clásico. Las desviaciones de los parámetros biomecánicos en modo intra-locutor pueden servir como indicadores de esfuerzo vocal abusivo, y en modo inter-locutor pueden utilizarse como detectores de patología y como elementos de caracterización biométrica.

6. BIBLIOGRAFÍA

[1] Akande, O. O. and Murphy, P. J., "Estimation of the vocal tract transfer function with application to glottal wave analysis", *Speech Communication*, Vol. 46, No. 1, May 2005, pp. 1-13.

[2] García, D., Aguilar, J., González, J. and Ortega, J., "Support Vector Machine Fusion of Idiolectal and Acoustic Speaker Information in Spanish Conversational Speech", *Proc. of the ICASSP'03*, Vol. II, pp. 229-232, 2003.

[3] Godino, J. I., Gomez, P., Automatic detection of voice impairments by means of short-term cepstral parameters and neural network based detectors. *IEEE Trans Biomed Eng.* Vol. 51, 2004, pp. 380-384.

[4] Gómez, P., Fernández, R., Nieto, A., Díaz, F., Fernández, F. J., Rodellar, V., Álvarez, A., Martínez, R., "Evaluation of Voice Pathology Based on the Estimation of Vocal Fold Biomechanical Parameters", *Journal of Voice*, March 17, 2006.

[5] Gómez, P., Godino, J. I., Álvarez, A., Martínez, R., Nieto, V., Rodellar, V., "Evidence of Glottal Source Spectral Features found in Vocal Fold Dynamics", *Proc. of the ICASSP'05*, 2005, pp. 441-444.

[6] Gómez, P., Godino, J. I., Díaz, F., Álvarez, A., Martínez, R., Rodellar, V., "Biomechanical Parameter Fingerprint in the Mucosal Wave Power Spectral Density", *Proc. of the ICSLP'04*, 2004, pp. 842-845.

[7] Gómez, P., Lázaro, C., Fernández, R., Nieto, A., Godino, J. I., Martínez, R., Díaz, F., Álvarez, A., Murphy, K., Nieto, V., Rodellar, V., Fernández, F. J., "Using biomechanical parameter estimates in voice pathology detection", *Proc. of the 4th Int. Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications MABEVA'05*, Florence, Italy, October 29-31, pp. 2005.

[8] Gómez, P., Martínez, R., Díaz, F., Lázaro, C., Álvarez, A., Rodellar, V., Nieto, V., "Estimation of vocal cord biomechanical parameters by non-linear inverse filtering of voice", *Proc. of the 3rd Int. Conf. on Non-Linear Speech Processing NOLISP'05*, Barcelona, Spain, April 19-22 2005, pp. 174-183.

[9] Hadjitodorov S, Mitev P. A computer system for acoustic analysis of pathological voices and laryngeal diseases screening. *Med Eng Phys.* 2002;24: 419-429.

[10] Hirano, M., Hibi, S., Yoshida, T., Hirade, Y., Kasuya, H., and Kikuchi, Y., "Acoustic analysis of pathological voice. Some results of clinical application," *Acta Otolaryngologica*, vol. 105, no. 5-6, pp. 432-438, 1988.

[11] McLaughlin, S., "Nonlinear Speech Synthesis", *Proc. of the EUSIPCO'02*, Vol III, 2002, pp. 211-218.

[12] Plumpe, M. D., Quatieri, T. F., Reynolds, D. A., "Modeling of the Glottal Flow Derivative Waveform with Application to Speaker Identification", *IEEE Trans. on Speech and Audio Proc.*, Vol. 7, No. 5, 1999, pp. 569-586.

[13] Story, B. H., "An overview of the physiology, physics and modeling of the sound source for vowels", *Acoust. Sci. & Tech.*, Vol. 23, No. 4, 2002, pp. 195-206.

[14] Story, B. H., and Titze, I. R., "Voice simulation with a bodycover model of the vocal folds", *J. Acoust. Soc. Am.*, Vol. 97 (1995), pp. 1249–1260.

[15] Švec, J. C., Horáček, J., Šram, F., Veselý, J., "Resonance properties of the vocal folds: *In vivo* laryngoscopic investigation of the externally excited laryngeal vibrations", *J. Acoust. Soc. Am.,* Vol. 108(4), pp. 1397-1407, 2000.

[16] Titze, IR. "Summary Statement", *Workshop on Acoustic Voice Analysis*, National Center for Voice and Speech, 1994.

ESTUDIO DEL USO DE INFORMACIÓN PROSÓDICA EN RECONOCIMIENTO DE LOCUTOR EN ÁMBITO FORENSE

Rubén Fernández Pozo[1], Carlos Fombella Mourelle[1], Doroteo Torre Toledano[2], Eduardo López Gonzalo[1] y Luis Hernández Gómez[1]

[1]Grupo de Aplicaciones de Procesado de Señales (Universidad Politécnica de Madrid),
[2]Área de Tratamiento de Voz y Señal (Universidad Autónoma de Madrid)
luis@gaps.ssr.upm.es

RESUMEN

En este trabajo se realiza un estudio de un conjunto de técnicas que hacen uso de la información prosódica aplicadas a la verificación de locutor dentro del campo de la Acústica Forense. Esta información ha sido aplicada siguiendo dos enfoques. Por un lado, se han implementado herramientas de modelado de lenguaje de la secuencia de segmentos obtenidos de la evolución de f0 y energía. Por otro lado, se han estudiado las posibilidades de mejorar un sistema básico de GMMs mediante una partición del espacio acústico fundamentada en la prominencia silábica. Estas técnicas se aplican sobre una base de datos de ámbito forense. Asimismo, se presentan los resultados obtenidos utilizando estas técnicas, y se muestra como se consigue una mejora de hasta un 10% en la tasa de error, al combinarlas con un sistema acústico de referencia.

1. INTRODUCCIÓN

En este articulo se presentan los resultados de una primera etapa de estudio de técnicas de utilización de la información de alto nivel que proporciona la evolución de las curvas de frecuencia fundamental y energía de la señal de voz, aplicadas al campo del reconocimiento de locutor en el ámbito forense. En este campo se tiene por objeto afrontar la problemática de la identificación del hablante presente en unas grabaciones ante un juez. Este escenario presenta una doble vertiente: Por un lado, disponer de una tecnología de identificación de locutor capaz de obtener tasas de error reducidas; y por otro, presentar de forma adecuada los resultados de la identificación ante el juez. Nuestro trabajo se enmarca dentro del primero de los enfoques presentados anteriormente, y tiene como objetivo principal el empleo de un conjunto de métodos recientes que hacen uso de la información de evolución de f0 y energía en una aplicación que se fundamenta en una base de datos especialmente desarrollada para acústica forense.

El estado del arte actual en los sistemas de identificación de locutor está representado por técnicas acústicas de bajo nivel, que hacen uso de información espectral en el dominio cepstral, representando los parámetros mediante GMMs, y adaptando los modelos mediante distintas técnicas a partir de un modelo universal [1]. Sin embargo, utilizando solamente este conjunto de técnicas, resulta difícil mejorar resultados actuales. En este sentido, los sistemas acústicos pueden verse favorecidos por la incorporación de técnicas de alto nivel, que aportan información incorrelada con la anterior, mejorando el rendimiento global del sistema. La información de alto nivel, además, presenta la ventaja de verse afectada en menor medida por el ruido y los efectos del canal. De este modo, al introducir elementos de nivel superior al acústico, se logra mayor robustez en el sistema, interesante también de cara a la aplicación planteada, pues las grabaciones en acústica forense no siempre se pueden realizar en condiciones idóneas [2].

En este trabajo se estudian las posibilidades para mejorar el rendimiento del sistema básico de GMMs a través de dos técnicas que hacen uso de la información prosódica. La primera de ellas se basa en los resultados presentados en [3], según los cuáles se hace uso de las trayectorias temporales de la f0 y la energía a través de una segmentación automática de la señal de voz en base a los máximos y mínimos de los parámetros citados, para caracterizar al locutor mediante un modelado de lenguaje de la secuencia de *tokens* obtenida. Con este enfoque se pretende utilizar esta información adicional de forma complementaria al sistema acústico de referencia a través de una fusión de ambos a nivel de resultados.

Por otra parte, proponemos también un nuevo esquema que utiliza la energía y frecuencia fundamental para identificar sílabas dentro de la señal de voz y asignar a cada una de ellas un valor de prominencia, es decir, un indicador de cómo se perciben estas unidades lingüísticas con respecto a las de su entorno. El objetivo de esta aproximación es analizar si es posible mejorar un sistema básico de GMMs modelando de forma controlada el espacio acústico en función de diferentes tipos de sílabas como se propone en [4], pero no clasificadas a través de su contenido fonético sino por su nivel de intensidad. Para ello se emplean técnicas de modelado acústico de forma similar a lo que se hacía en [4], pero utilizando en este caso información prosódica en vez de información fonética. Con esta metodología vamos a estudiar si la voz asociada a distintos niveles de prominencia contribuye también de manera diferente a la discriminación de voces en un sistema de verificación de locutor de la misma forma que lo hacían las distintas clases fonéticas [5].

Estas técnicas han sido aplicadas sobre la base de datos BDRA, muy adecuada para la tarea, dado que está constituida por grabaciones telefónicas de casos reales. Esta base de datos supone un escenario ideal para su uso en experimentos de identificación de locutor en Acústica Forense.

El artículo está estructurado de la siguiente forma: En la sección 2 se determinan las características de la base de datos empleada. En la sección 3 se describen las técnicas utilizadas y en la sección 4 se revisan los resultados obtenidos. Por último la sección 5 contiene las conclusiones y líneas futuras basadas en los resultados.

2. BDRA

Uno de los elementos fundamentales de este trabajo es la Base de Datos sobre la que se han realizado las pruebas. Los experimentos descritos en este artículo utilizan voz procedente de la base de datos B.D.R.A (Base de Datos de Registros Acústicos), integrada por voz extraída de casos forenses reales ya cerrados, y cuyo material grabado (intervenciones telefónicas, registros en contestadores, grabaciones ocultas, etc.) pasa a formar parte de la base de datos con fines de investigación mediante autorización judicial. Gracias al uso de esta base, aportada por la Guardia Civil dentro del Proyecto del Plan Nacional TIC2003-09068, las pruebas se han podido realizar en un ámbito muy similar al que podría darse en una situación real.

Esta base de datos contiene muchas horas de conversaciones que necesitan de un trabajo previo de selección y etiquetado para su uso en sistemas de identificación automática de locutor. Por ello, para las pruebas elaboradas en este trabajo se ha hecho una selección de 50 locutores diferentes, cada uno de ellos con al menos 13.5 minutos de conversación.

Las grabaciones utilizadas se han obtenido de conversaciones espontáneas desarrolladas tanto sobre líneas telefónicas fijas como sobre líneas GSM. Cada fichero contiene únicamente voz de un solo interlocutor (un único lado de conversación). Los datos de voz correspondientes al locutor en cuestión están concatenados directamente, de forma que en un solo fichero está toda la información del locutor. Por lo general, la voz está tomada de una única línea telefónica (no necesariamente de un único terminal), aunque hay casos en que se utilizaron dos líneas.

Para homogeneizar las pruebas se ha limitado la duración de todos los segmentos de voz de los distintos locutores a 13 minutos y medio (el mínimo disponible), desechando el resto. Asimismo, para dotar de mayor flexibilidad al estudio, cada segmento de voz se dividió en fragmentos de 30 segundos. De este modo, la base de pruebas final queda compuesta por 27 segmentos de 30 segundos para cada uno de los 50 locutores.

Para realizar el entrenamiento de los distintos sistemas se agruparon 10 de estos segmentos de 30 segundos, por un total de 5 minutos. Los ensayos se llevaron a cabo enfrentando segmentos de 30 segundos con los modelos, de modo que en total se dispuso de 16 *target tests* por locutor. Esto supone un total de 800 *target tests*. Para construir modelos universales se utilizó un fichero de 30 segundos de cada uno de los locutores (un segmento no utilizado ni como fichero de prueba ni como fichero de entrenamiento), para obtener un total de 25 minutos de entrenamiento. De este modo, se logra maximizar el número de ensayos y a la vez crear un UBM de cierta entidad, partiendo de los 50 locutores disponibles.

Por último, cabe señalar la dificultad que presenta esta base de datos para tareas de identificación de locutor, lo que va parejo precisamente a la buena caracterización que hace de una posible situación real de identificación. La voz ha sido capturada a través de distintos terminales, a lo que hay que añadir la variabilidad de estados de ánimo y el alto grado de espontaneidad que presentan los locutores en las grabaciones.

3. TÉCNICAS DESARROLLADAS

En este trabajo se han implementado dos conjuntos de técnicas diferentes, ambas basadas en la extracción de información derivada de la evolución de f0 y energía, pero con dos enfoques distintos.

En primer lugar se describe una técnica que utiliza modelos de lenguaje para caracterizar la variación temporal de los contornos de información prosódica bajo estudio, esto es, frecuencia fundamental y energía. El objetivo principal de este enfoque será utilizar esta información de alto nivel para mejorar el rendimiento del GMM básico.

Después, se detalla un sistema basado en los estudios realizados en [4], pero con una partición del espacio acústico fundamentada en la prominencia silábica. En este apartado se analizará también la relación de este tipo de modelado con el que utiliza información fonética. Por tanto, en este caso el objetivo es mejorar directamente el sistema de GMMs básico.

Finalmente, una vez determinado qué sistema acústico es el que presenta mejores resultados, se combinará con la información prosódica proveniente de los modelos de lenguaje con el objeto de mejorar el rendimiento global del sistema.

3.1. Sistema basado en modelado de lenguaje.

Este sistema utiliza modelos de lenguaje para representar las trayectorias cuantificadas de energía y f0, previamente segmentadas en base a los puntos de inflexión de ambas señales.

La segmentación se lleva a cabo sobre las señales de energía y f0, obtenidas utilizando el método *get_f0* incluido en la librería de tratamiento de voz *Snack* [6]. Las señales se filtran paso bajo y a continuación se extraen sus derivadas. Los puntos de inflexión quedan determinados por los cortes de estas derivadas con cero, siempre que en ese punto tengan una pendiente mayor que cierto umbral. Este umbral, de valor 0.1, fue obtenido de forma experimental conforme a un criterio que garantice segmentos no excesivamente cortos.

Una vez segmentadas las señales de energía y f0, se cuantifican ambas trayectorias, utilizando solamente dos niveles para cada una de ellas. De este modo se tienen segmentos crecientes o decrecientes tanto para energía como para f0. Esta información se combina en un solo flujo utilizando las fronteras de ambas señales de forma conjunta, del modo indicado en la figura 1.

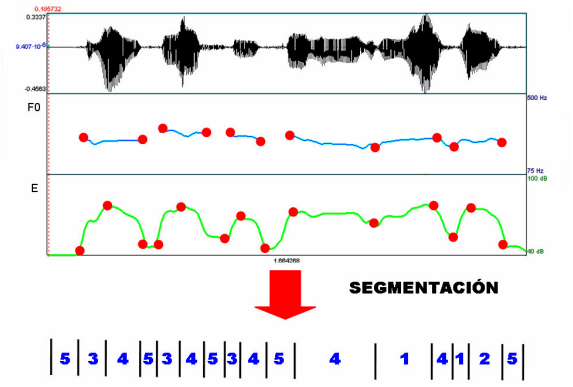


Figura 1. *Proceso de segmentación y cuantificación de las trayectorias de energía y f0.*

A cada segmento se le asigna un identificador de entre los 5 siguientes, en función de la combinación de pendientes de energía y f0.

Clase	Cuantificación de E y f0
1	F0 ascendente y Energía ascendente
2	F0 ascendente y Energía descendente
3	F0 descendente y Energía ascendente
4	F0 descendente y Energía descendente
5	Segmento no sonoro

Tabla 1. *Clasificación de segmentos*

Esta técnica se ha complementado ampliando las 5 clases anteriores con información sobre la duración del segmento (cuantificada en tres niveles: corto, medio o largo). De este modo se amplia el número de clases a 15.

Una vez cuantificados los segmentos, se procesa la secuencia resultante mediante modelos de lenguaje. Así, se crea un modelo de bigramas para cada locutor.

La puntuación utilizada para la evaluación del sistema se basa en la perplejidad de la secuencia de test frente al modelo de lenguaje. Esta puntuación se compara con la puntuación de la secuencia de test frente a un modelo de lenguaje universal, mediante una relación de verosimilitudes. Esta relación determina la puntuación final del segmento de test.

Esta técnica se ha fusionado a nivel de puntuaciones con un sistema fonético de reconocimiento similar al descrito en [7]. Éste emplea un transcriptor fonético que genera una secuencia de fonemas, modelados también a través de un sistema de modelos de lenguaje. En la sección 4 se presentan los resultados de la combinación de el sistema propuesto con el sistema acústico de referencia.

3.2. Sistema acústico por clases de sílabas según valor de prominencia

En este caso, intentaremos aprovechar la información de energía y frecuencia fundamental para segmentar el espacio acústico de características de un sistema de GMMs, de forma similar a la presentada en [4]. En este último trabajo, la señal de voz se divide en segmentos diferentes correspondientes a pseudo-sílabas con diferente estructura fonética. Posteriormente cada tipo de sílaba se modela utilizando un GMM específico de forma que el número total de gaussianas se distribuye uniformemente entre los GMMs de todas las clases fonéticas. En nuestro sistema la categorización se realizará a través de clases definidas por sílabas según el nivel de prominencia con el que se pronuncian. Con esto pretendemos estudiar si es posible conseguir un reparto más eficiente de las gaussianas que modelan toda la voz, controlando su distribución en funcion de los diferentes niveles de prominencia de las sílabas. En un sistema básico de GMMs no se puede asegurar de antemano un número determinado de gaussianas para cada clase, cosa que sí se puede garantizar con sistemas de GMMs categorizados por clases. Además, otro punto favorable para estos sistemas de clases, es que con este esquema de modelado se pueden eliminar clases que no contribuyen a diferenciar entre distintos locutores.

3.2.1 Segmentado en Sílabas

El primer paso para categorizar la voz según el nivel de prominencia consiste en el segmentado en sílabas de la señal de voz, para lo cual se han estudiado sistemas capaces de identificar automáticamente sílabas sin ayuda de ningún tipo de transcripción fonética previa. La implementación utilizada se basa en [8], sistema que trabaja con la envolvente de energía sobre la que buscan máximos y mínimos para localizar núcleos y límites de las sílabas. Este algoritmo es muy popular puesto que es particularmente sencillo, ya que trabaja directamente sobre la envolvente espectral sin la necesidad de utilizar ninguna herramienta estadística. El algoritmo implementado se puede resumir en los siguientes puntos [9].

1. Filtramos paso banda la señal de audio en el rango de 500 a 4000 Hz utilizando para ello un filtro de segundo orden de Butterworth.
2. Filtramos paso bajo el cuadrado de la señal resultante a 12 Hz obteniendo de esta forma la envolvente de energía. Para asegurar que no haya desfase se utilizará un filtro bidireccional.
3. Calculamos el *convex-hull* de la envolvente de energía.
4. Restamos la envolvente de energía al *convex-hull* de la envolvente. La señal resultante tiene picos que se corresponden a los valles de la envolvente.
5. Sujeto a ciertas condiciones resumidas abajo el máximo de la señal diferencia es seleccionado como límite de sílaba y el algoritmo (desde el paso 3) es ejecutado recursivamente sobre los subintervalos delimitados por el límite.
6. El algoritmo finaliza cuando no se encuentran límites que cumplan las condiciones dentro del subintervalo.

Las condiciones comentadas más arriba que hacen que el máximo de la señal diferencia sea considerado o no límite de sílaba se pueden ver a continuación:

- La máxima diferencia entre la envolvente de energía y el *convex-hull* debe ser mayor de 2 dB
- Los subintervalos a la izquierda y derecha del límite de sílaba deben ser mayores de 80 ms.
- La diferencia entre el pico de intensidad de cada subintervalo y el pico de intensidad de la señal entera no debe ser mayor de 25 dB

Todos estos parámetros han sido ajustados a las singulares condiciones de la BDRA. El algoritmo de Mermelstein generalmente funciona bien con sílabas claramente articuladas pero no detecta correctamente sílabas cortas no acentuadas. Las condiciones anteriores que impiden la inserción de nuevas sílabas son las responsables de este problema. Y es que la segmentación precisa de la señal de voz en sílabas es una tarea compleja y exigente, problema que se acentúa en mayor medida en bases de datos de ámbito forense como la BDRA.

3.2.2. Cuantificación de las sílabas por prominencia

Una vez dividida la señal de voz en sílabas, se procede a la cuantificación de los segmentos resultantes según su nivel de prominencia. Una sílaba se considera prominente si es percibida por encima de las de su entorno, lo que supone la aparición de al menos uno de los dos fenómenos prosódicos siguientes [10]:

- ***Pitch accent*** (acento de pitch): que está acústicamente relacionado con la evolución de la frecuencia fundamental y la energía total de la sílaba.

- ***Stress accent*** (acento tónico): Correlado con la duración del núcleo de la sílaba y la energía de la banda de 300-2200 Hz (*mid-to-high-frequency emphasis).*

A continuación examinaremos cada uno de los parámetros acústicos anteriormente citados y que nos permitirán cuantificar el nivel de prominencia de cada sílaba. Hay que resaltar aquí que todos ellos deben ser normalizados para evitar las variaciones naturales entre los diferentes locutores:

a. **Duración**: como comentamos anteriormente, la segmentación precisa de la señal de voz en sílabas es una tarea costosa que requiere de un esfuerzo minucioso al no poder automatizarse fácilmente. Es por ello, por lo que es necesario introducir una medida, que pueda ser automáticamente obtenida y que sea alternativa en la medición de la prominencia a la duración total de la sílaba. Este parámetro es la duración del núcleo de la sílaba que puede ser obtenida con facilidad mediante una variación del algoritmo de Mermelstein y que aporta la misma información que la duración de la sílaba en el cálculo de la prominencia.
b. **Energía de la banda de frecuencias media-alta**: como se puede ver en trabajos anteriores [11], la energía en la banda media del espectro (de 300 a 2200 Hz.) es un parámetro útil para determinar sílabas acentuadas.
c. **Energía**: en concreto el valor RMS de energía del núcleo de la sílaba.
d. **Contorno de f0**: siguiendo el modelo propuesto por Taylor, el contorno de f0 se convierte primero en un modelo RFC. La curva de f0 se divide en tramas de 0.02 seg. que son clasificadas, dependiendo de su gradiente y tras una interpolación lineal de los mismos, en subida (*rise*), bajada (*fall*) y contorno plano (*connection*). Contornos seguidos clasificados de la misma forma son fusionados en un solo intervalo y se mide la duración y amplitud de la sección de subida o bajada. Una vez obtenida esta representación RFC, es posible identificar en la curva eventos prosódicos tales como los acentos de *pitch* (una perfil de subida seguido de otro de bajada) que serán asignados al núcleo de la sílabas más cercana con un valor marcado por los parámetros del modelo TILT que están definidos a continuación:

$$A_{\text{event}} = |A_{\text{rise}}| + |A_{\text{fall}}|$$
$$D_{\text{event}} = |D_{\text{rise}}| + |D_{\text{fall}}|$$

donde A_{rise}, A_{fall}, D_{rise} y D_{fall} son respectivamente la amplitud y duración de los segmentos de subida y bajada del acento de *pitch*. En concreto se puede cuantificar la importancia del evento del acento del *pitch* multiplicando el evento amplitud (A_{event}) por su duración (D_{event}) y normalizar el producto por un factor de peso que exprese la relevancia del evento a lo largo de la locución (R_{event}). Este factor es calculado dividiendo el evento amplitud por la máxima variación de f0 y el máximo valor absoluto de f0 a lo largo de locución.

Como se muestra más arriba en esta misma sección, la prominencia puede ser definida desde un punto de vista teórico, como la combinación de dos eventos prosódicos (*pitch accent* y *stress accent*) que a su vez están relacionados cualitativamente con los parámetros acústicos vistos anteriormente. Es por esto, por lo que teniendo en cuenta estas relaciones, podemos construir una función continua de prominencia para cada una de la sílabas, directamente obtenida de los valores de los parámetros acústicos [10]:

$$Prom^i = \max \{ en^i_{300\text{-}2200} \cdot dur^i , en^i_{ov} \cdot (A^i_{\text{event}} \cdot D^i_{\text{event}} \cdot R^i_{\text{event}})\}$$

Este valor continuo se cuantificará en 4 niveles con una cuantificación de perfil exponencial con el fin de garantizar un reparto lo más uniforme posible de la señal de voz entre las clases de prominencia.

4. RESULTADOS

En esta sección se presentan los resultados obtenidos con la aplicación de las distintas técnicas desarrolladas en los puntos anteriores sobre la B.D.R.A.

Las pruebas llevadas a cabo se centran en la verificación de locutor, es decir, dada una frase dubitada, el sistema debe decidir si se corresponde con un modelo de locutor determinado o no.

En sistemas de ámbito forense se suelen utilizar como herramientas de evaluación las curvas Tippet [12]. Sin embargo, en nuestro estudio preliminar los resultados se compararán a través del *Equal Error Rate* (EER), o Tasa de Error Equivalente. Este indicador representa el error correspondiente al umbral que hace que los dos tipos de errores posibles en un sistema de verificación de locutor, Falsa Aceptación y Falso Rechazo, sean iguales. A menor EER, mejor es el rendimiento del sistema. Asimismo se proporcionan las curvas DET (*detection error tradeoff*), en cuyo punto central se sitúa el citado EER.

4.1. Sistema acústico básico de referencia.

El sistema básico de referencia que se ha empleado es un GMM de 256 gaussianas. Los modelos de locutor se han adaptado a partir de un modelo universal. Este sistema produce un EER de 4.5 %.

4.2. Sistema basado en modelado de lenguaje.

En la figura 2 se presentan los resultados correspondientes al sistema prosódico basado en trayectorias de f0 y energía. Las cuatro curvas representan el sistema prosódico básico, el sistema prosódico con adición de duraciones, el sistema fonético de referencia y la fusión de los dos últimos. La fusión se ha realizado a nivel de puntuaciones.

Los valores de EER figuran a continuación:

Sistema	EER
f0_e0	18.38 %
F0_e0_duración	15.75 %
Fonético	12.38 %
Fusión fonético + f0_e0_dur	8,37%

Tabla 2. *EER para el sistema basado en el modelado de lenguaje de segmentos prosódicos*

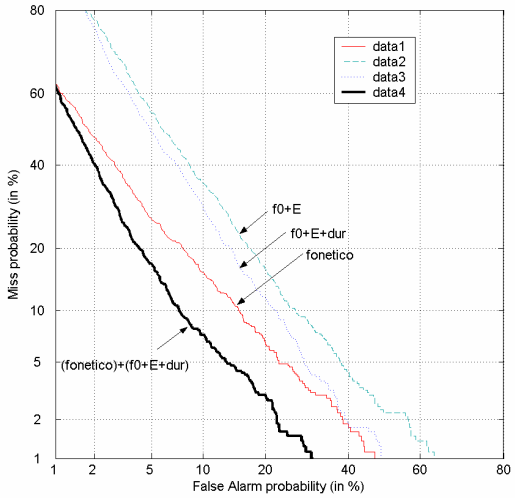


Figura 2. *Resultados para el sistema basado en el modelado de lenguaje de segmentos prosódicos*

A la vista de los resultados se puede concluir que la adición de nueva información prosódica a un sistema fonético de modelos de lenguaje supone una mejora en el rendimiento del sistema.

4.3. Sistema acústico por clases de sílabas según el valor de prominencia.

A continuación se detallan los resultados derivados del sistema acústico-lingüístico descrito en el apartado 3.2. La clasificación de GMMs por prominencias produce los siguientes resultados:

Clase	EER
Prominencia 1	7.25 %
Prominencia 2	7 %
Prominencia 3	7 %
Prominencia 4	6.125 %
Fusión de las 4 Clases	**5.875 %**

Tabla 3. *EER para clases de prominencias.*

Estos resultados muestran que las sílabas prominentes son las que poseen más capacidad discriminativa. No obstante, se puede observar que ninguna de las clases aisladas ni la fusión de ellas mejora el sistema acústico de referencia. La posible causa de este comportamiento puede ser que cada una de las clases resultantes tiene un contenido fonético heterogéneo. Esto hace que el espacio acústico de cada clase sea similar al de un GMM básico pero modelado con un menor número de gaussianas.

Para comprobar que una distribución adecuada del espacio acústico mejora el rendimiento del sistema básico de GMMs, se proporcionan además resultados (Tabla 4) obtenidos de un sistema en el que la partición del espacio acústico se obtiene a partir de una segmentación de la señal de voz en clases fonéticas, de modo similar al descrito en [5]:

Clase	EER
Clase 1 (vocales)	5.75 %
Clase 2 (nasales + vibrantes)	7.375 %
Clase 3 (fricativas + líquidas)	6.25 %
Clase 4 (oclusivas)	5.625 %
Fusión de las 4 Clases	**5.75 %**

Tabla 4. *EER para clases fonéticas*

Estos resultados nos muestran que este sistema tampoco mejora el GMM básico. La posible causa de estos resultados puede ser que la transcripción fonética no es precisa con las deficientes condiciones de articulación del habla de los locutores que se dan en la BDRA. Esto no suponía un problema con el sistema fonético descrito en la sección 3.1 puesto que son precisamente los errores en la transcripción los que contribuyen en mayor medida al proceso de reconocimiento de locutor [13]. Sin embargo, en nuestro sistema, para asegurar una adecuada distribución del espacio acústico sí es necesaria una segmentación precisa.

Por otro lado, puede resultar llamativo que las nasales aporten el peor de los resultados de entre todas las clases, cuando son junto a las vocales las que tienen mayor poder discriminativo [14]. Analizando el resultado de la transcripción fonética observamos que hay una carencia de datos correspondientes a la clase 2, lo que justifica su peor funcionamiento.

Se presentan además resultados de nuevas propuestas de fusión, mezclando las clases fonéticas y clases de prominencia más discriminativas, de modo que se construya un GMM que aproveche de forma más eficiente las gaussianas disponibles

Sistema	EER
Fus. C.Foneticas + C.Prominencia	5.375 %
Fus. CF1 + CF4 + CP3 + CP4	5.125 %

Tabla 5. *EER para la fusión de sistemas.*

En primer lugar, figura la fusión de las cuatro clases fonéticas y las cuatro clases de prominencia. En segundo lugar se presenta el EER de la fusión de las dos mejores clases fonéticas y las dos mejores clases de prominencia. Se puede ver como se mejora la tasa de error equivalente respecto a los sistemas aislados, aunque no mejora en ningún caso el rendimiento del GMM básico.

4.4. Fusión del sistema acústico básico con el sistema prosódico de modelos de lenguaje.

A partir de los resultados de los apartados anteriores, el modelado acústico que ha proporcionado mejores resultados es el uso de un GMM único. Por lo tanto se ha estudiado la fusión de este reconocedor con el de información prosódica y fonética (Sección 3.1). Así, fusionando a nivel de puntuaciones el sistema acústico con el sistema de alto nivel de modelos de lenguaje para secuencias de fonemas y *tokens* prosódicos, se obtiene un ERR del 4.05 % que supone una mejora del 10%. Esta mejora, demuestra que la inclusión de datos de alto nivel aporta información incorrelada al sistema acústico básico.

5. CONCLUSIONES Y LINEAS FUTURAS

A lo largo del artículo se ha podido ver como una misma información, en este caso, la energía y la frecuencia fundamental, se puede estudiar y emplear de dos formas muy distintas, con el objetivo de mejorar el rendimiento del sistema acústico básico. Por una parte, se ha implementado un sistema de modelos de lenguaje sobre los contornos de f0 y energía que, junto con un sistema fonético, produce una mejora de un 10% respecto al GMM básico.

Por otro lado, se ha propuesto un nuevo esquema en el uso de la información de f0 y energía, basado en [4], que permite segmentar el espacio acústico mediante la prominencia de las sílabas presentes en la locución. Sin embargo, se ha comprobado que los resultados de este sistema no superan al sistema básico de referencia. Por otro lado, el sistema basado en la segmentación fonética del espacio acústico, a pesar de resultar efectivo en [4], tampoco supone una mejora respecto al sistema de referencia sobre la base de datos utilizada. Una explicación para este comportamiento podría estar en la dificultad de realizar de forma precisa una segmentación fonética de la base de datos. Debido a las condiciones que presenta la BDRA, se pueden producir clases poco uniformes que incluyan segmentos pertenecientes a otras clases. No obstante, el marco presentado permite desarrollar líneas futuras de trabajo centradas en una mejor distribución del espacio acústico.

6. AGRADECIMIENTOS

Este trabajo ha sido posible gracias a la subvención aportada por el Proyecto TIC2003-09068-C02-02 del Plan Nacional de I+D.

7. REFERENCIAS

[1] D.A. Reynolds, T.F.Quatieri, y R.B. Dunn "Speaker verification using adapted gaussian mixture models", *Digital Signal Processing*, 10 (1-3), pp 19-41, 2000.

[2] SuperSID. Supersid: "Exploiting high-level information for high-performance speaker recognition", http://www.clsp.jhu.edu/ws2002/groups/supersid/. 2002.

[3] A. Adami, R. Mihaescu, D.A. Reynolds, y J. Godfrey, "Modelling prosodic dynamics for speaker Recognition", en *International Conference on Acoustics, Speech and Signal Processing, ICASSP* 2003.

[4] B. Baker, R. Vogt, y S. Sridharan, "Gaussian Mixture-Modelling of Broad Phonetic and Syllabic Events for Text-Independent Speaker Verification", en *International Conference on Speech and Language Processing* 2005

[5] M. Hébert, y L.P. Heck, "Phonetic class-based speaker verification", en *INTERSPEECH* 2003.

[6] K. Sjölander, "The Snack Sound Toolkit v.2.2.8", http://www.speech.kth.se/snack/ 2004

[7] W.D. Andrews, M.A. Kholer, y J.P. Campbell, "Phonetic speaker recognition" en *INTERSPEECH* 2001

[8] P. Mermelstein, "Automatic segmentation of speech into syllabic units", *Journal Acoustical Society of America*, 58 (4), pp 880-3, Octubre 1975

[9] R. Villing, J. Timoney, T. Ward, y J. Costello, "Automatic Blind Syllable Segmentation for Continuous Speech", en *Irish Signal and System Conference* Julio 2004

[10] F. Tamburini, y C. Caini, "An automatic system for detecting prosodic prominence in American English continuous speech", *International Journal of Speech Technology*, 8, pp 33-44, 2005

[11] A. Sluijter, y V. van Heuven, "Acoustic correlates of linguistic stress and accent in Dutch and American English", en *International Conference on Speech and Language Processing* 1996.

[12] J. Gonzalez, J. Fierrez, y J. Ortega, "Forensic identification reporting using automatic speaker recognition systems", en *International Conference on Acoustics, Speech and Signal Processing, ICASSP,* Abril 2003.

[13] D. Torre, C. Fombella, J. González, y L. Hernández, "On the Relationship between Phonetic Modeling Precision and Phonetic Speaker Recognition Accuracy", en *INTERSPEECH* 2005

[14] M. Antal, y G. Toderean, "Broad Phonetic Classes Expressing Speaker Individuality", *Studia Informatica*, Vol. LI, No. 1, pp. 49-58, 2006.

SELECCIÓN Y PESADO DE PARÁMETROS ACÚSTICOS MEDIANTE ALGORITMOS GENÉTICOS PARA EL RECONOCIMIENTO DEL LOCUTOR

Maider Zamalloa[1,2], Germán Bordel[1], Luis Javier Rodriguez[1], Mikel Peñagarikano[1], Juan Pedro Uribe[2]

(1) GTTS, Departamento de Electricidad y Electrónica, Universidad del País Vasco
(2) Ikerlan – Centro de Investigaciones Tecnológicas
mzamalloa001@ikasle.ehu.es

RESUMEN

Los coeficientes cepstrales constituyen la representación acústica más empleada en tareas de reconocimiento del locutor. Generalmente, la dimensión de esta representación se amplía introduciendo características dinámicas. Algunas de estas características pueden depender de otras o ser redundantes. Este artículo estudia la selección del conjunto óptimo de parámetros acústicos para reconocimiento del locutor, mediante una búsqueda heurística basada en algoritmos genéticos. Para mejorar el rendimiento del sistema, tras la selección se ha añadido un pesado. La búsqueda del conjunto óptimo de pesos se ha efectuado también mediante algoritmos genéticos. De este estudio se desprende que el conjunto óptimo de parámetros es el formado por los 12 cepstrales y la energía. Así pues, la información dinámica no parece tan relevante como en reconocimiento del habla. Además, se ha obtenido una reducción relativa del error de reconocimiento del 24.31%. Además de validar la metodología propuesta, estos resultados refuerzan la opción de utilizar representaciones acústicas reducidas para aligerar el coste computacional y aumentar la robustez de los sistemas de reconocimiento.[1]

1. INTRODUCCIÓN

El empleo de sistemas biométricos que analizan la voz para verificar o determinar identidades es actualmente la forma más natural y menos intrusiva de solucionar los problemas relacionados con el control de acceso a recursos críticos o privados (edificios, cuentas de Internet, etc.). Las aplicaciones de alta seguridad requieren que la eficacia de estos sistemas sea alta, algo especialmente difícil cuando el sistema interactúa con cientos de usuarios legítimos y miles de potenciales impostores. Sin embargo, es algo que se puede abordar adecuadamente por medio de grandes cantidades de datos de habla y redes de servidores computacionalmente potentes, que permiten la estimación de modelos robustos.

No obstante, también pueden beneficiarse de esta tecnología ciertas aplicaciones cuyo principal requerimiento no es la seguridad, sino la adaptación y personalización automática de los servicios que ofrecen. En este tipo de aplicaciones, la transparencia y la naturalidad son críticas, siendo preciso no interferir con el funcionamiento del servicio prestado. En un escenario cada vez más previsible para este tipo de aplicaciones, el cliente puede haber accedido al servicio por medio de un dispositivo portátil o embebido, con limitaciones de almacenamiento y capacidad computacional, lo que plantea la necesidad de aligerar el coste computacional de los procesos de reconocimiento con el menor impacto posible en su eficiencia.

Una vía para abordar esta reducción de costes computacionales pasa por la elección de parámetros acústicos, que es clave para el reconocimiento automático de locutores. Sería deseable contar únicamente con parámetros que sean relevantes para la tarea de clasificación, ignorando todo dato redundante o que aporte información innecesaria (el vector de parámetros acústicos es portador de información diversa, como el idioma, el locutor, el mensaje emitido, la emoción etc.). Una reducción del conjunto de parámetros permitirá reducir el coste computacional y, en función de las características de los datos a utilizar para la estimación de modelos, incluso aumentar la robustez del sistema. Yendo un poco más allá, aún limitándonos estrictamente al uso de los parámetros significativos, cabría hablar de mejores y peores parámetros en función su aportación relativa a la distinción entre locutores, cuestión que puede tenerse en cuenta para mejorar el rendimiento del sistema dado un determinado conjunto de parámetros.

Un parámetro acústico óptimo debe contar con las siguientes características: (1) alta variación inter-locutor, (2) baja variación intra-locutor, (3) facilidad de medición, (4) robustez frente a ataques por simulación, (5) robustez frente al ruido y (6) independencia frente a otros parámetros. Desafortunadamente, no existe ningún parámetro que cumpla con todos estos requerimientos**.** Los parámetros de alto nivel que caracterizan a un locutor, tales como los patrones de pronunciación, el uso del lenguaje, etc., son robustos frente al ruido pero requieren el empleo de reconocimiento del habla para obtener la secuencia de palabras y una gran cantidad de datos para estimar los modelos acústicos y de lenguaje. Esta complejidad es hoy por hoy inadmisible en cualquier sistema de reconocimiento del locutor. Por lo tanto, son los parámetros acústicos de bajo nivel los más empleados, ya que su extracción es fácil, no requieren el reconocimiento del habla y basta con una relativamente pequeña cantidad de datos para estimar buenos modelos. Su principal inconveniente es que pueden ser fácilmente corrompidos por el ruido ambiental y otras fuentes de distorsión.

Los sistemas actuales emplean los mismos parámetros (*Mel-Frequency Cepstral Coefficients*, *MFCC*) tanto para reconocimiento del habla como para reconocimiento del locutor, porque además de recoger la distribución frecuencial para la identificación de sonidos, transmiten información sobre el pulso glotal y la forma y longitud del tracto vocal,

[1] Este trabajo ha sido parcialmente financiado por el Gobierno Vasco, dentro del programa SAIOTEK, a traves del proyecto S-PE04UN18.

que son características específicas de cada hablante. Asimismo, diversos trabajos han demostrado que la información dinámica mejora de forma significativa la eficiencia de los sistemas de reconocimiento del habla, por lo que además de los MFCC y la energía, generalmente se utilizan también sus primeras y segundas derivadas. En función del módulo de procesamiento acústico, el vector de parámetros puede estar compuesto por un número de componentes que típicamente varía entre 20 y 50.

La determinación de un subconjunto de parámetros óptimo partiendo de un vector acústico *D*-dimensional, sin poner límites a toda la combinatoria posible, es computacionalmente prohibitiva incluso para valores moderados de *D,* sobre todo si tenemos en cuenta que hemos de usar la tasa de clasificación como función de evaluación. Una metodología más simple consistiría en evaluar de forma individual los *D* parámetros y seleccionar los *K* más discriminantes, sin tener en cuenta las dependencias entre ellos. De hecho, se han propuesto varias técnicas subóptimas en la literatura que sacrifican la optimización del conjunto seleccionado por la eficiencia computacional [1].

Una estrategia bastante razonable para realizar esta búsqueda sin establecer límites a priori a toda la combinatoria posible es la utilización de *algoritmos genéticos*. Los algoritmos genéticos, introducidos por Holland en 1975 [2], son técnicas de búsqueda heurísticas y aleatorias basadas en las estrategias de la evolución biológica, con tres mecanismos básicos: selección del más fuerte, mezcla y mutación. Las soluciones candidatas se representan por medio de individuos o *cromosomas*. La población inicial de individuos suele generarse de forma aleatoria. Después, el algoritmo genético, de forma iterativa, conduce a la población hacia un punto óptimo, de acuerdo a una métrica compleja (función de evaluación o de adecuación) que mide la eficiencia de los individuos para llevar a cabo una tarea concreta. El algoritmo selecciona los individuos más fuertes (los que reciben mejor evaluación) y a continuación mezcla, muta o mantiene sus representaciones. Una de las principales ventajas de los algoritmos genéticos sobre otras técnicas de búsqueda heurísticas consiste en que no realizan ninguna hipótesis sobre las propiedades de la función de evaluación pudiendo incluso definir y emplear funciones de evaluación multi-objetivo de forma completamente natural [3][4].

Hemos podido localizar varios trabajos que aplican estas técnicas para la extracción y selección de parámetros en tareas de reconocimiento del locutor [5][6][7]. Además de seleccionar los parámetros más adecuados, las mismas técnicas se pueden aplicar al pesado de parámetros, sin más que codificar, en vez de una decisión booleana, el peso que se asigna a cada parámetro. El pesado establece una modulación de los parámetros acústicos, aumentando la influencia de los más relevantes. De esta forma es posible mejorar la eficacia del reconocedor en sistemas de verificación e identificación del locutor [5][8].

En este trabajo se plantea la búsqueda de los parámetros acústicos más relevantes para el reconocimiento del locutor. Se trata de obtener conjuntos reducidos de parámetros que permitan ahorrar esfuerzo computacional con una baja degradación o incluso con una mejora del rendimiento del sistema. Será necesario analizar un gran número de subconjuntos de parámetros de diferentes dimensiones. Se parte de un vector acústico de 38 dimensiones, compuesto por los 12 MFCC, sus primeras y segundas derivadas, la energía y su derivada. Tras una primera fase de selección, en la que se determina el conjunto óptimo de *K* parámetros, se aborda una segunda fase de pesado, en la que se establece el conjunto óptimo de pesos para un conjunto dado de *K* parámetros.

El artículo se estructura como sigue. En el apartado 2 se describe el sistema de reconocimiento del locutor y los métodos de selección y pesado mediante algoritmos genéticos. En el apartado 3 se describen la base de datos de habla utilizada en los experimentos y la configuración del algoritmo genético empleado en las búsquedas. En el apartado 4 se muestran y comentan los resultados obtenidos. Por último, el apartado 5 contiene las conclusiones del trabajo y una breve descripción de las tareas pendientes.

2. METODOLOGÍA

2.1. El sistema de reconocimiento del locutor

2.1.1. Procesamiento acústico

El habla, adquirida a 16 KHz, se analiza en tramos de 25 ms, a intervalos de 10 ms. A continuación, se aplica una ventana de análisis tipo Hamming y se calcula una FFT de 512 puntos y se calcula la media de las amplitudes de la FFT en 24 filtros triangulares solapados, definidos de acuerdo a la escala Mel de bandas críticas. Por último, se calcula el logaritmo de los coeficientes del banco de filtros y se aplica una transformada discreta de cosenos (DCT), que produce 12 coeficientes cepstrales. Para aumentar la robustez frente a la distorsión del canal se resta la media en el tiempo de los cepstrales (calculada independientemente para cada señal) [9]. También se calculan las primeras y segundas derivadas de los coeficientes cepstrales, la energía (E) y su derivada, conformando un vector acústico de 38 componentes.

2.1.2. Cuantificación vectorial

Se aplica el algoritmo de cuantificación vectorial LBG [10], que produce como resultado un conjunto de centroides (*codebook*), *C*, que minimiza la distorsión media (distancia euclídea) en la cuantificación de los vectores acústicos del corpus de entrenamiento (que incluye a todos los locutores). A continuación, cada vector acústico de la base de datos se reemplaza por el índice del centroide más cercano. Durante el reconocimiento se aplica el mismo procedimiento. Cada señal de entrada es procesada para obtener una secuencia de vectores acústicos $X=\{X(1),X(2),...,X(T)\}$ y a cada $X(i)$ se le asigna el índice del centroide más cercano del codebook, $Y(i)$, obteniendo una secuencia equivalente de etiquetas acústicas $Y=\{Y(1),Y(2),...,Y(T)\}$.

2.1.3. Modelos de locutor

La aproximación clásica al reconocimiento de locutor mediante cuantificación vectorial consiste en estimar un codebook específico para cada locutor, y después a cada señal de entrada asignarle aquel locutor cuyo codebook haya producido la mínima distorsión al cuantificar los vectores acústicos. En este trabajo los modelos de locutor no son codebooks, sino distribuciones de etiquetas. El método consiste en calcular un único codebook compartido por todos los locutores, y almacenar como parámetros específicos de cada locutor las frecuencias de las etiquetas. Estos modelos tan sencillos han dado buenos resultados en adaptación al locutor mediante agrupamiento de locutores en tareas de reconocimiento del habla [11].

Supongamos que $U(i)$ es el conjunto de entrenamiento correspondiente al locutor i, $c(i)$ el número de etiquetas que

aparecen en *U(i)*, y *c(k,i)* el número de veces que aparece la etiqueta *k* en *U(i)*. La probabilidad de la etiqueta *k* dado el locutor *i*, *P(k|i),* se puede calcular empíricamente de la siguiente manera:

$$P(k \mid i) = \frac{c(k,i)}{c(i)} \tag{1}$$

Finalmente, suponiendo independencia estadística entre etiquetas sucesivas, la probabilidad de la secuencia de etiquetas *Y={Y(1),Y(2),...,Y(T)}*, dado el locutor *i*, puede calcularse como sigue:

$$P(Y \mid i) = \prod_{t=1}^{T} P(Y(t) \mid i) \tag{2}$$

2.1.4. Reconocimiento del locutor

Suponiendo que cada señal de entrada ha sido pronunciada por alguno de los *S* locutores conocidos, dada la secuencia de etiquetas *Y={Y(1),Y(2),...,Y(T)}*, el locutor más probable viene dado por la siguiente expresión:

$$\hat{\imath}(Y) = \arg\max_{i=1,\ldots,S} \{P(i \mid Y)\} \tag{3}$$

Aplicando la regla de Bayes, queda:

$$\hat{\imath}(Y) = \arg\max_{i=1,\ldots,S} \left\{ \frac{P(Y \mid i)P(i)}{P(Y)} \right\} = \arg\max_{i=1,\ldots,S} \{P(Y \mid i)P(i)\} \tag{4}$$

ya que la maximización sobre el conjunto de locutores no depende de la secuencia acústica. Por tanto, dado que además suponemos que todos los locutores tienen la misma probabilidad a priori:

$$\hat{\imath}(Y) = \arg\max_{i=1,\ldots,S} \{P(Y \mid i)\}, \tag{5}$$

e introduciendo (2) en (5):

$$\begin{aligned} \hat{\imath}(Y) &= \arg\max_{i=1,\ldots,S} \left\{ \prod_{t=1}^{T} P(Y(t) \mid i) \right\} \\ &= \arg\max_{i=1,\ldots,S} \left\{ \log \prod_{t=1}^{T} P(Y(t) \mid i) \right\} \\ &= \arg\max_{i=1,\ldots,S} \left\{ \sum_{t=1}^{T} \log P(Y(t) \mid i) \right\} \end{aligned} \tag{6}$$

De la expresión (6) deducimos que el coste computacional del reconocimiento es lineal con el número de locutores *(S)* y con la longitud de la frase de entrada *(T)*. En total se requieren *SxT* accesos a memoria, *Sx(T-1)* sumas y *S-1* comparaciones. Por conveniencia, se supone que las probabilidades de las etiquetas se almacenan en forma logarítmica.

2.1.5. Selección y pesado de los parámetros acústicos

El proceso de selección consiste en establecer cuáles son las *K* componentes del vector acústico más relevantes desde el punto de vista de la clasificación. Entre las numerosas tecnicas de búsqueda y criterios de optimización que se han propuesto a lo largo de los años (vease [12]), por razones que han quedado expuestas anteriormente, se ha optado por una búsqueda heurística basada en algoritmos genéticos.

El proceso comienza normalizando la base de datos para que las *D* componentes del espacio vectorial tengan media 0 y varianza 1. Cada subconjunto de *K* parámetros propuesto como candidato, $\Gamma^{(K)}=\{j_1, j_2, \ldots, j_K\}$ ha de ser evaluado, para finalmente quedarse con aquel subconjunto que reciba una mejor evaluación. Los pasos a seguir para hacerlo son: (1) cada vector de la base de datos es sustituido por el vector reducido, formado por las componentes enumeradas en $\Gamma^{(K)}$; (2) se genera un codebook reducido *C'*, utilizando las componentes $\Gamma^{(K)}$ del codebook completo *C*; (3) se etiquetan los corpus de entrenamiento y validación reducidos mediante *C'*; (4) utilizando las etiquetas del corpus de entrenamiento se estiman los modelos de locutor; (5) se aplican los modelos de locutor para clasificar las señales del corpus de validación; y (6) se asigna al conjunto $\Gamma^{(K)}$ la tasa de reconocimiento obtenida.

Una vez se ha establecido el subconjunto óptimo de *K* componentes, $\hat{\Gamma}^{(K)} = \{\hat{j}_1, \hat{j}_2, \ldots, \hat{j}_K\}$, se calcula un nuevo codebook, $C(\hat{\Gamma}^{(K)})$, se etiquetan con él los vectores reducidos de los corpus de entrenamiento y test, y se calculan los modelos de locutor. Por último, el subconjunto óptimo $\hat{\Gamma}^{(K)}$ se evalúa clasificando las señales del corpus de test. Los tres conjuntos de señales utilizados en este trabajo, entrenamiento, validación y test, son independientes, ya que contienen señales distintas del mismo conjunto de locutores.

Eliminar parámetros acústicos poco relevantes reduce, sin duda, los requerimientos computacionales del reconocedor, pero tal como apuntan los resultados obtenidos en un trabajo previo [8], aún es posible mejorar el rendimiento del sistema buscando y aplicando un conjunto adecuado de pesos *W=(w1,w2,...,wK),* que transforma cada vector acústico *X=(x1,...,xK)* en un vector ponderado *X'=(w1x1,w2x2,...,wKxK)*.

Es necesario, por tanto, efectuar un proceso de búsqueda en el espacio *K*-dimensional de los pesos. Cada conjunto de pesos candidato *W* se aplica a los vectores reducidos de la base de datos para obtener los vectores ponderados. Después se calcula un codebook específico *C(W),* utilizando los vectores ponderados del corpus de entrenamiento. Cada vector ponderado es sustituido por el índice del centroide más cercano en *C(W)*. Por último, utilizando las etiquetas del corpus de entrenamiento se estiman los modelos de locutor, y se asigna a *W* la tasa de reconocimiento de locutor obtenida sobre el corpus de validación utilizando dichos modelos. Una vez obtenido el conjunto de pesos óptimo, $\hat{W}$, como medida de evaluación final se calcula la tasa de reconocimiento sobre el corpus de test reducido y ponderado.

2.2. Búsqueda basada en algoritmos genéticos

La selección y el pesado de los parámetros acústicos se realizan por medio de un conocido algoritmo genético (*Simple Genetic Algorithm*, *SGA*) [13]. Tanto el codebook de cuantificación vectorial como los modelos de locutor se estiman a partir de los datos del corpus de entrenamiento. Sin embargo, la evaluación que tiene lugar dentro del proceso de búsqueda se efectúa a partir de los datos del conjunto de validación, lo que asegura una mayor robustez e independencia de los resultados. En este trabajo se utiliza como función de evaluación la tasa de reconocimiento de locutor sobre el corpus de validación.

El algoritmo SGA comienza evaluando todos los individuos de una población de partida generada

aleatoriamente. Por individuo entendemos o bien un subconjunto de K componentes en el caso de la selección, o bien un conjunto de K pesos en el caso del pesado. Una vez evaluados todos los candidatos, se seleccionan algunos de ellos, normalmente los más fuertes o adecuados, es decir, aquellos que ocupan los primeros puestos de acuerdo a la función de evaluación, para mezclarlos y mutarlos con objeto de obtener la población para la siguiente generación. La convergencia del algoritmo depende de varios elementos: la forma de representar la información, el tamaño de la población, la forma de obtener la población inicial, el criterio de selección de individuos, el tipo de mezcla y la tasa de mutación. En concreto, hay tres métodos de selección muy comunes: proporcional, por rango y por torneo. Los dos primeros favorecen mucho a los individuos más fuertes, mientras que la selección por torneo da más probabilidad a los individuos menos adaptados. La mezcla siempre se realiza entre dos individuos previamente seleccionados, pero puede realizarse de muchas formas. El método más común consiste en definir n puntos de corte, que determinan $n+1$ segmentos en cada individuo, e intercambiar los segmentos pares para producir los dos individuos hijos. Una mutación consiste en invertir el valor de ciertos bits tomados al azar en la representación de un individuo. Se introduce para evitar en lo posible que la búsqueda quede atrapada en extremos locales. La llamada *tasa de mutación* establece la probabilidad de que tras una mezcla tengan lugar mutaciones.

A veces se permite que ciertos individuos (generalmente los más fuertes) se transmitan sin mezclar ni mutar a la siguiente generación, lo que se conoce como *elitismo*. En este trabajo se ha aplicado el caso más simple de elitismo, que consiste en transmitir únicamente al individuo más fuerte. Esto garantiza que la evaluación del mejor individuo mejore monótonamente en sucesivas generaciones. Si la mejora es inferior a un cierto umbral, o si se ha alcanzado un número máximo de generaciones, el algoritmo finaliza y devuelve como solución el mejor individuo.

2.2.1. Representación de la información en la selección de parámetros

Para representar la selección de K componentes se utiliza un vector de D enteros en el rango [0, 255], $R=\{r_1, r_2, ..., r_D\}$. Los K valores más altos indican implícitamente qué componentes se están eligiendo: $\Gamma^{(K)} = \{\{j_1, j_2, ..., j_K\} \mid r_{j1} \geq r_{j2} \geq ... \geq r_{jK} \geq r_j, \forall j \notin \{j_1, j_2, ..., j_K\}\}$. Obviamente, un cierto conjunto de componente $\Gamma^{(K)}$ puede ser representado por muchos vectores distintos, o lo que es igual, dado un cierto individuo R, pequeños cambios en su especificación no modifican la elección de componentes. Esta redundancia en la representación facilita que la evolución del algoritmo sea suave y converja al subconjunto óptimo $\hat{\Gamma}^{(K)}$ idenpendientemente de la población inicial.

3. CONFIGURACIÓN DE LOS EXPERIMENTOS

3.1. La base de datos

Para los experimentos se ha utilizado *Albayzin* [14], base de datos fonéticamente equilibrada en castellano. Grabada a 16 kHz en condiciones de laboratorio, *Albayzin* fue diseñada para el entrenamiento de modelos acústicos en el ámbito del reconocimiento del habla. La base de datos está compuesta por 204 locutores, cada uno de ellos con un mínimo de 25 señales. La longitud media de cada señal es de 3.55 segundos.

En los experimentos se utilizan casi todas las señales de la base de datos. De cada locutor se toman 25 señales, 10 de ellas para el corpus de entrenamiento, 7 para el de validación y 8 para el de test. Así pues, el corpus de entrenamiento contiene 2040 señales, el de validación 1428 y el de test 1632.

3.2. Configuración del algoritmo genético

El algoritmo genético se ha implementado por medio de ECJ *"A Java-based Evolucionary Computation and Genetic Programming Research System"*, una herramienta para el desarrollo de aplicaciones de computación evolutiva y programación genética, creada y mantenida en la universidad George Mason de Estados Unidos, que se ofrece bajo una licencia especial de código abierto [15]. ECJ tiene una arquitectura muy flexible, permite definir representaciones arbitrarias, genomas de longitud fija y variable, métodos de optimización multiobjetivo y diversos operadores de selección.

Con objeto de ajustar los parámetros que controlan el rendimiento y la convergencia del algoritmo genético, se han llevado a cabo experimentos preliminares. El tamaño de la población es uno de los principales parámetros. Poblaciones demasiados extensas retrasan excesivamente la convergencia del algoritmo, mientras que poblaciones demasiado pequeñas limitan el rendimiento de la búsqueda. Se ha comprobado que el tamaño de la población depende del espacio de búsqueda. En los experimentos de selección todos los individuos contienen 38 genes, uno por cada parámetro del vector acústico. Así pues, se ha llegado a un tamaño de población óptimo para todos ellos. Sin embargo, en los experimentos de pesado el número de genes es proporcional al número de parámetros a pesar. Por tanto, los experimentos de pesado requieren poblaciones cada vez más pequeñas a medida que se reduce el número de parámetros.

Se ha comprobado que 40 generaciones son suficientes para alcanzar la convergencia, por lo que no se ha aplicado ningún otro criterio de parada. Para reducir el tamaño de la representación y, por tanto, el coste computacional, se han utilizado tan sólo 8 bits por gen, es decir, enteros en el rango [0, 255]. A la hora de elegir qué individuos de la actual generación van a cruzarse para producir los individuos de la siguiente, se ha optado por la selección por torneo, con una pre-selección aleatoria de 7 o 2 individuos (de nuevo, este número depende del tamaño de la población), entre los que se escoge el más fuerte. Este es el método recomendado por los desarrolladores de la herramienta, ya que permite mantener un cierto grado de diversidad y, al mismo tiempo, dirigir el algoritmo hacia poblaciones cada vez mejor adaptadas. Se ha aplicado una mezcla con un punto de corte, es decir, cada uno de los dos individuos seleccionados se rompe en dos segmentos que se intercambian. Las tasas de mezcla y mutación se han fijado heurísticamente a aquellos valores que han producido mejores resultados (1.0 y 0.01, respectivamente). Por último, se ha aplicado el caso básico de elitismo, manteniendo el mejor individuo de cada generación.

4. RESULTADOS DE LOS EXPERIMENTOS

4.1. Selección de parámetros

Con objeto de identificar el subconjunto óptimo de parámetros tanto en cuanto a coste computacional como en cuanto a eficacia del sistema de reconocimiento, se ha lanzado

una amplia batería de experimentos con subconjuntos de diferentes dimensiones. Puesto que el coste computacional es lineal con el número de parámetros, los experimentos tratan de encontrar el mejor compromiso posible entre la tasa de reconocimiento y el número de parámetros. Se han llevado a cabo experimentos con *K = 13, 12, 10, 8, 6, 4* y *2*. En la Tabla 1 se muestran los conjuntos óptimos, así como las tasas de error correspondientes, obtenidas sobre el corpus de test.

Tabla 1. S*ubconjuntos óptimos de parámetros obtenidos tras la selección mediante algoritmos genéticos, y tasas de error en experimentos de reconocimiento sobre el corpus de test de Albayzin, para varios valores de K. El identificador cXX hace referencia al cepstral XX, y E a la energía.*

K	*%Error*	*Parámetros Seleccionados*
13	5.45	c01, c02, c03, c04, c05, c06, c07, c08,c09, c10, c11, c12, E
12	**5.39**	c01, c02, c03, c04, c06, c07, c08, c09, c10, c11, c12, E
10	6.25	c01, c02, c04, c05, c06, c08, c09, c10, c11, c12
8	9.13	c01, c02, c04, c07, c08, c09, c10, c12
6	16.23	c01, c02, c03, c06, c08, c12
4	30.94	c01, c02, c03, c10
2	73.16	c01, c04

En todos los experimentos de selección desde *K=12* hasta *K=2,* la búsqueda, que se efectúa sobre un espacio de 38 dimensiones, ha dado como resultado un subconjunto óptimo formado por algunos cepstrales y la energía. El hecho de que no haya resultado seleccionada ninguna de las características dinámicas parece indicar que éstas no tienen la misma relevancia en reconocimiento del locutor que en reconocimiento del habla. El experimento de selección con *K=13* se ha llevado a cabo precisamente para verificar esta intuición. Atendiendo a las tasas de error, se concluye que el mejor subconjunto es el compuesto por 12 parámetros (11 cepstrales más la energía). No obstante, los subconjuntos de 10 y 13 parámetros ofrecen un rendimiento muy similar.

Puesto que la energía no aparece en los subconjuntos óptimos de tamaño inferior a 12, se ha considerado oportuno evaluar el rendimiento potencial del conjunto formado por los 12 cepstrales. La tasa de error obtenida sobre el conjunto de test utilizando 12 cepstrales es del *5.94%*, sólo ligeramente peor que la de los subconjuntos óptimos de tamaño 12 y 13 obtenidos mediante algoritmos genéticos.

4.2. Pesado de parámetros

En un trabajo previo se presentaron resultados interesantes de pesado con el conjunto completo de 38 componentes [8]. Sin embargo, el objetivo principal de este trabajo es identificar subconjuntos de parámetros que permitan reducir los costes computacionales de los sistemas de reconocimiento de locutor. Pero reducir el número de componentes puede repercutir negativamente en la tasa de reconocimiento, tal como se observa en la Tabla 1 al ir reduciendo el número de componentes de 10 a 2. El pesado se introduce, precisamente, con objeto de mejorar el rendimiento del sistema cuando se utilizan conjuntos reducidos de parámetros acústicos.

Así pues, se ha realizado una búsqueda mediante algoritmos genéticos de los pesos óptimos para los vectores reducidos obtenidos en la fase de selección, para *K=13, 10, 8, 6, 4* y *2*. Además, como referencia, se han obtenido también los pesos óptimos para el vector original de 38 componentes y para el vector formado por los 12 cepstrales. En la Tabla 2 se muestra el error de reconocimiento obtenido sobre el corpus de test utilizando parámetros sin pesar y parámetros pesados con los pesos óptimos.

Tabla 2. *Resultados de reconocimiento de locutor sobre el corpus de test de Albayzin utilizando parámetros sin pesar y parámetros pesados con los pesos óptimos obtenidos mediante algoritmos genéticos, para el conjunto original de 38 componentes, el conjunto de los 12 cepstrales y los subconjuntos óptimos obtenidos en la fase de selección.*

		% ERROR	
K		*Sin pesos*	*Pesos óptimos*
38 (vector completo)		19.42	5.51
12 cepstrales		5.94	4.53
Subconjuntos óptimos obtenidos mediante algoritmos genéticos	13	5.45	**4.17**
	10	6.25	5.15
	8	9.13	7.23
	6	16.23	14.83
	4	30.94	28.86
	2	73.16	74.08

Los parámetros pesados proporcionan casi en todos los casos tasas de error más pequeñas. La única excepción se produce para *K=2*, debido a la no convergencia del algoritmo genético. El uso de los 38 parámetros originales pesados conlleva una reducción relativa del error del 71.62%. Por otra parte, la búsqueda de los pesos óptimos para *K=38* presenta una convergencia extraordinariamente rápida. Esto es debido a la normalización de la base de datos, que unifica las varianzas de todos los parámetros. En particular, hace que las derivadas adquieran más relevancia en el proceso de clasificación e inserten ruido. Por tanto, resulta muy beneficioso asignarles un peso pequeño, es decir, devolverles la pequeña varianza que tenían. Este hecho se ve reflejado en los pesos óptimos obtenidos para los 38 parámetros originales, entre los que claramente prevalecen los de los cepstrales y la energía (véase la Tabla 3). Hay que tener en cuenta que los pesos determinan la relevancia de cada parámetro en el proceso de reconocimiento.

Tabla 3. *Pesos óptimos obtenidos para el conjunto original de 38 parámetros acústicos. El identificador cXX se refiere al cepstral XX, dXX y ddXX a su primera y segunda derivada, E a la energía y dE a su derivada.*

Parámetro	Peso	Parámetro	Peso	Parámetro	Peso	Parámetro	Peso
c01	225	d01	58	dd01	14	E	220
c02	152	d02	93	dd02	61	dE	25
c03	228	d03	62	dd03	147		
c04	223	d04	118	dd04	161		
c05	238	d05	33	dd05	104		
c06	218	d06	11	dd06	114		
c07	189	d07	65	dd07	61		
c08	209	d08	39	dd08	14		
c09	207	d09	45	dd09	43		
c10	204	d10	36	dd10	60		
c11	184	d11	124	dd11	112		
c12	237	d12	73	dd12	84		

Atendiendo a los resultados mostrados en la Tabla 2, se puede concluir que el subconjunto óptimo de parámetros acústicos pesados es el seleccionado para *K=13*, compuesto por los 12 cepstrales y la energía. Este conjunto proporciona un error del 4.17%, 1.34 puntos inferior al obtenido con los 38 parámetros originales, lo que supone una reducción relativa del error del 24.31%. Los pesos óptimos para el conjunto formado por los 12 cepstrales y la energía se muestran en la Tabla 4. Finalmente, entre los conjuntos de parámetros pesados, sólo tres superan al formado por los 38 parámetros originales: los subconjuntos seleccionados para *K=10* y *K=13*, y el conjunto formado por los 12 cepstrales.

Tabla 4. *Pesos óptimos obtenidos para el conjunto de los 12 cepstrales y la energía.*

c01	c02	c03	c04	c05	c06	c07	c08	c09	c10	c11	c12	E
168	174	136	197	189	149	148	149	160	137	166	138	240

5. CONCLUSIONES

Los coeficientes cepstrales y sus derivadas son los parámetros más empleados en los sistemas actuales de reconocimiento de locutor. El mismo conjunto de parámetros se emplea también en otras tareas que requieren el procesamiento de señales de voz, como el reconocimiento del habla o la identificación de la lengua. Sin embargo, algunos de estos parámetros pueden ser redundantes, dependientes entre sí o incluso perjudiciales para la clasificación. Desde ese presupuesto, este trabajo trata de determinar el subconjunto óptimo de parámetros acústicos para el reconocimiento de locutor, tanto en cuanto al rendimiento del sistema como en cuanto al coste computacional, que en este caso es lineal con el número de parámetros.

El estudio se basa en la selección y posterior pesado de diferentes conjuntos de parámetros, mediante técnicas heurísticas de búsqueda, en concreto, mediante algoritmos genéticos. El algoritmo de selección trata de identificar los parámetros más relevantes, mientras que el pesado escala los parámetros seleccionados y mejora la robustez del sistema de reconocimiento.

Los resultados obtenidos en los experimentos realizados sobre la base de datos de habla leída *Albayzin*, con 204 locutores, determinan que el subconjunto óptimo pesado para el reconocimiento del locutor es el compuesto por los 12 cepstrales y la energía. Cabe destacar que este subconjunto, que excluye la información dinámica, es más eficaz que el conjunto pesado compuesto por los 38 parámetros originales, con una reducción relativa del error del 24.31%. Este resultado pone en evidencia la importancia de eliminar la información no relevante del vector acústico, puesto que al excluir los parámetros dinámicos no sólo disminuye el coste computacional sino que aumenta el rendimiento del sistema de reconocimiento. En definitiva, se puede concluir que los parámetros dinámicos no aportan información relevante para la clasificación de locutores, sino que más bien entorpecen dicha tarea.

Puesto que el pesado no es más que una transformación lineal con matriz diagonal, una vez comprobado el efecto positivo que tiene esta medida, se plantea como trabajo futuro la búsqueda de matrices de transformación que optimicen el rendimiento del sistema de reconocimiento. En un sentido más práctico, actualmente se está implementando una Aplicación Orientada a Servicio de reconocimiento de locutor para dispositivos con recursos computacionales limitados (teléfonos móviles y PDAs).

6. BIBLIOGRAFÍA

[1] A.K. Jain, R.P.W. Duin, J. Mao. *"Statistical Pattern Recognition: A Review"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1, pp. 4—37, Enero 2000.

[2] J.H. Holland. *"Adaptation in natural and artificial systems"*. University of Michigan Press, 1975 (reeditado en 1992 por MIT Press, Cambridge, MA).

[3] J. Yang, V. Honavar. *"Feature subset selection using a genetic algorithm"*, IEEE Intelligent Systems, Vol. 13, No. 2, pp. 44—49, Marzo 1998.

[4] L.S. Oliveira, R. Sabourin, F. Bortolozzi, C.Y. Suen. *"A Methodology for Feature Selection Using Multiobjective Genetic Algorithms for Handwritten Digit String Recognition"*, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 17, No. 6, pp. 903—929, 2003.

[5] D. Charlet, D. Jouvet. *"Optimizing feature set for speaker verification"*, Pattern Recognition Letters, Vol. 18, No. 9, pp. 873—879, Septiembre 1997.

[6] M. Demirekler, A. Haydar. *"Feature Selection Using a Genetics-Based Algorithm and its Application to Speaker Identification"*, Proceedings of the IEEE ICASSP'99, pp. 329—332, Phoenix, Arizona, 1999.

[7] C. Charbuillet, B. Gas, M. Chetouani, J.L. Zarader. *"Filter Bank Design for Speaker Diarization Based on Genetic Algorithms"* Proceedings of the IEEE ICASSP'06, Toulouse, Francia, Mayo 2006.

[8] M. Zamalloa, G. Bordel, L.J. Rodriguez, M. Peñagarikano, J.P. Uribe. *"Using Genetic Algorithms to Weight Acoustic Features for Speaker Recognition"*, Proceedings of the ISCLP´06, Pittsburgh, USA, September 2006.

[9] A.E. Rosenberg, C.H. Lee, F.K. Soong. *"Cepstral Channel Normalization Techniques for HMM-Based Speaker Verification"*, Proceedings of the ICSLP'94, pp. 1835—1838, Yokohama, Japón, 1994.

[10] Y. Linde, A. Buzo, R.M. Gray. *"An Algorithm for Vector Quantizer Design"*, IEEE Transactions on Communications, Vol. 28, No. 1, pp. 84—95, Enero 1980.

[11] L.J. Rodríguez, M.I. Torres. *"A Speaker Clustering Algorithm for Fast Speaker Adaptation in Continuous Speech Recognition"*, in P. Sojka, I. Kopecek and K. Pala Eds., Proceedings of the 7th International Conference on Text, Speech and Dialogue (Brno, República Checa, Septiembre 2004), pp. 433—440, LNCS/LNAI 3206, Springer-Verlag, 2004.

[12] M. Zamalloa, G. Bordel, L.J. Rodriguez, M. Peñagarikano. *"Features Selection Based on Genetic Algorithms for Speaker Recognition"*, IEEE Speaker Odyssey: The Speaker and Language Recognition Workshop, Puerto Rico, Junio 2006.

[13] D.E. Goldberg. *"Genetic Algorithms in Search, Optimization and Machine Learning"*, Addison-Wesley, 1989.

[14] F. Casacuberta, R. García, J. Llisterri, C. Nadeu, J.M. Pardo, A. Rubio. *"Development of Spanish Corpora for Speech Research (Albayzín)"*, in G. Castagneri Ed., Proceedings of the Workshop on International Cooperation and Standardization of Speech Databases and Speech I/O Assessment Methods, pp. 26-28, , Italia, Septiembre 1991.

[15] ECJ 15, http://cs.gmu.edu/~eclab/projects/ecj/.

On the use of high-level information in speaker and language recognition

Alberto Montero-Asenjo, Javier Gonzalez-Dominguez, Daniel Ramos-Castro, Ignacio Lopez-Moreno, Doroteo Torre Toledano y Joaquin Gonzalez-Rodriguez

ATVS (Speech and Signal Processing Group)
Escuela Politecnica Superior
Universidad Autonoma de Madrid

Abstract

Automatic Speaker Recognition systems have been largely dominated by acoustic-spectral based systems, relying in proper modelling of the short-term vocal tract of speakers. However, there is scientific and intuitive evidence that speaker specific information is embedded in the speech signal in multiple short- and long-term characteristics. In this work, a multilevel speaker recognition system combining acoustic, phonotactic and prosodic subsystems is presented and assessed using NIST 2005 Speaker Recognition Evaluation data.

For language recognition systems, the NIST 2005 Language Recognition Evaluation was selected to measure performance of a high-level language recognition systems.

1. Introduction

Speaker recognition systems are automatic systems that provide information about the identity of the speaker of a given speech segment. As a verification system it must decide whether or not the identity of the speaker is a claimed one. As an identification system the goal is to determine the identity of the speaker among a set of predefined speakers.

Regardless the operation mode, the system works by computing similarity measures (scores) between the speech segment and a previously stored model (extracted from other speech segment). Based on the computed measures, the system may perform a hard decision (yes/no, accepted/rejected) or a soft one, providing a score for a subsequent module. For the hard decisions case, a threshold must be determined, and the decision emerge from the comparison between the score and the threshold.

Despite the fact that text-independent identification of speakers by their voices has been a subject of interest for decades, the first really successful results in actual telephone conversational speech came in the 90s, where acoustic-spectral based systems [1] were able to obtain remarkable performance in really challenging out-of-laboratory tasks. The series of NIST Speaker Recognition Evaluations (SRE) has fostered research and development in this area since the mid-90s [2]. This important forum has led to yearly significant improvements in the speaker recognition technology, which has been shared among participants to these evaluations. However, there was by that time significant room for improvement which was not taken into account in the use of higher non-acoustic levels of information. This information has demonstrated to be extremely characteristic in the inter-speaker communication process and well-known in linguistics, but it was not exploited at that time by automatic speaker recognition technology. It was in early 00s when the pioneering work on idiolectal differences between speakers [3] and specially the confluence of different sources of knowledge that were presented in the SuperSID project [4] gave a major impulse to multilevel and fusion approaches to automatic speaker recognition. Presently, multilevel speaker recognition systems may include generative [1] or discriminative [5] acoustic-spectral sub-systems, prosodic [6], and phonotactic [4] sub-systems among others [7].

2. Speaker recognition techniques overview

Speaker recognition techniques can be broadly classified as those relaying in short-time spectral and acoustic information and those using high-level features (phonetic, prosodic, lexical information, etc.). A third group may be considered, in which would be included those systems made by combination (fusion) of several other systems.

Short-time spectral information systems include Gaussian Mixture Models (GMM)[1] and Support Vector Machines (SVM) [5]. GMM has been the reference in the past decade, but SVM systems are now competitive.

High-level features based systems include prosodic and phonotactic based systems. In both situations, the speech is converted to streams of tokens used to compute the score. Prosodic systems use pitch and energy as tokens while phonotactic systems use phones in some language.

GMM and SVM systems perform much better than prosodic or phonotactic systems, but the combination of them may improve the final systems [8][7].

3. High-level speaker recognition techniques

The interest in the use of these higher level features was motivated by the work of Doddington [3], who used the lexical content of the speech, modeled through statistical language models (word n-grams), for speaker recognition using the Switchboard-II corpus. This relatively simple technique improved the results obtained by an acoustic-only speaker recognition system. After the work of Doddington a number of research works have continued exploring the use of higher level features in the field of speaker recognition. Some of these works [4] made use of similar techniques (n-gram statistical language models) applied to the output of phonetic decoders (i.e. speech recognition engines configured to recognize any phonetic sequence), leading to the techniques known as phonotactic speaker recognition. Instead of modeling the lexical content, these techniques aim to model speaker pronunciation idiosyncrasies. This technique also yielded promising results, particularly when several phonetic decoders for different languages were used and combined. More recently, similar modeling techniques (n-gram

statistical language models) have been applied to model the prosody (mainly fundamental frequency and energy) of the different speakers [4][6], giving rise to the field known as prosodic speaker recognition. As in the initial work of Doddington [3], all of these higher-level techniques were particularly useful in combination with traditional acoustic-only speaker recognition systems.

3.1. Statistical grammar modelling

The most common modelling technique for tokens sequences is statistical modelling, where the probability of a sequence given a language model is used as the basis for scoring.

Given a sequence of tokens (words, phones, prosodic tokens, data driven units, etc.)

$$S = (w_1, w_2, \ldots, w_m)$$

the probability of occurrence can be decomposed as a product of conditional probabilities

$$P(w_1, w_2, \ldots, w_m) = \prod_{i=1}^{m} P(w_i \mid w_1, \ldots, w_{i-1}) \quad (1)$$

Usually eq.1 is approximated by limiting the context:

$$P(w_1, w_2, \ldots, w_m) \simeq \prod_{i=1}^{m} P(w_i \mid w_{i-n+1}, \ldots, w_{i-1}) \quad (2)$$

for some $n \geq 1$. Due to reasons of data sparsity n is usually selected in the range of 1 to 4.

Estimates of probabilities in n-gram models are commonly based on maximum likelihood estimates – that is, by counting events in context on some given training text:

$$P(w_i|w_{i-n+1}, \ldots, w_{i-1}) = \frac{C(w_{i-n+1}, \ldots, w_i)}{C(w_{i-n+1}, \ldots, w_{i-1})} \quad (3)$$

where $C(.)$ is the count of a given word sequence in the training text. For robust estimation, probability smoothing techniques can be applied.

3.2. Process overview

The process starts with speaker models training, from speech transcriptions or tokenization. Direct robust estimation of probabilities requires a big amount of data, usually not available for a single speaker. The procedure to train more robust models is to train the UBM model using a lot of data and then adapt that model to the data available for a particular speaker. This adaptation is made by linearly interpolating the n-grams models of the UBM and the one trained only with speaker data. That interpolation is governed by an weighting factor that has to be empirically determined.

To obtain a score related to a given sequence, usually a log-likelihood detector is used. The target model SPM_i will be the n-gram model adapted to transcriptions from speech from the speaker. The alternate model will be a n-gram model trained with transcriptions from speech from many speakers (UBM model). The final score for sequence S of m tokens is obtained as

$$s_i = \frac{1}{m} \log \frac{P(S/SPM_i)}{P(S/UBM)} \quad (4)$$

where $P(S/SPM_i)$ and $P(S/UBM)$ are calculated as shown in eq. 2.

3.3. Phonotactic systems

Phonotactic systems use phonetic transcribers to convert speech into a sequence of tokens where each token is a phone.

A typical phonotactic speaker recognition system consists of two main building blocks: the phonetic decoders, which transform speech into a sequence of phonetic labels and the n-gram statistical language modeling stage, which models the frequencies of phones and phone sequences for each particular speaker. The phonetic decoders can either be taken from a pre-existing speech recognizer or trained ad hoc. Any speech recognition technology can be used, but usually phonetic decoders are based on Hidden Markov Models and null grammars.

Reported experiments have been performed using phonetic decoders for Castillian Spanish (using the Albayzin Corpus [9]), American English (using the TIMIT corpus) and Basque (using Basque-SpeechDAT corpus).

Once the phonetic sequence has been obtained the scoring process is performed as explained above.

3.4. Prosodic systems

A prosodic speaker recognition system consists of two main building blocks: the prosodic tokenizer, which analyses the prosody, and represents it as a sequence of prosodic labels or tokens, and the n-gram statistical language modeling stage, which models the frequencies of prosodic tokens and their sequences for each particular speaker, as explained above. The tokenization process carried out consists of two stages. Firstly, for each speech utterance, both temporal trajectories of the prosodic features, (fundamental frequency or pitch- and energy) are extracted. Secondly, both contours are segmented and labelled by means of a slope quantification process. The slope quantification process was performed as follows: first, a finite set of tokens were defined using a four level quantization of the slopes (fast-rising, slow-rising, fast-falling, slow-falling) for both energy and pitch contours [6] .Thus, the combination of levels generate sixteen different tokens when combined pitch and energy contours are considered. Second, both contours were segmented using the start and end of voicing and the maximums and minimums of the contours. These points were detected as the zero-crossings of the contours derivatives using a ± 2 frame span. On the other hand, silence intervals were detected with an energy-based voice activity detector. Finally, each segment was converted into a set of tokens which describe the joint-dynamic variations of slopes. Therefore, utterances with different sequences of tokens contain different prosodic information. Since errors in the pitch and energy estimation are likely to generate small segments, all segments smaller than a certain amount (tipically 30 ms) are removed from the sequence of joint-state classes.

Additionally to the sixteen tokens defined for the joint-dynamic of the prosodic features, a special token must be added to represent unvoiced segments.

3.5. Data-driven phonotactic systems

Both phonetic and phonotactic systems use as tokens items with linguistic meaning: phones, energy trajectories, pitch. But from an engineering point of view the tokens do not need to be limited to these. Some researchers have proposed the use of units extracted automatically from speech samples.

ALISP systems are based on this idea. Units are similar to phones but the information modelled by each unit is determined by means of an automatic clustering process. Results reported

in [10] show that this approach can outperform classical phonetic systems.

3.6. PhoneSVM systems

PhoneSVM systems share many elements with a phonotactic system. Both rely on phonetic transcriptions and n-gram model estimation, but differ in the way the score is computed. While in classical phonotactic systems the scores are calculated as log-likelihood ratios, in phoneSVM systems a SVM is used to determine the score.

The training process starts by estimating n-gram models. One for the speaker under consideration and many more from other different speakers. The probabilities of the different n-grams are considered as the coordinates of a vector, thus transforming the problem into a one suitable for training a SVM. Results presented in [11] show major improvements.

3.7. Phonetic decoding using phone lattices

Phone lattices is a technique initially proposed for language recognition [12] that was later successfully applied to speaker recognition [13].

The underlying idea is how to increase the amount of data available for n-gram estimation. Extracting a single phone sequence from a speech segment to estimate n-grams is an idea borrowed from the speech recognition field, where only one sequence is useful as the transcription of a speech segment (unless further processing is performed, but at last, only one sequence will be output). But in speaker and language recognition fields, the sequence itself is not important, but for estimation purposes, so if instead of considering the best sequence, the N best sequences are considered, the amount of data for parameter estimation largely increases, thus relying to better estimates. The decrease in error rates (as reported in [12] and [13]) largely compensates the increase in execution time.

4. Multilevel speaker recognition techniques fusion

There are many works related to the combination of different speaker characteristics and modelling methods for a speaker recognition system, such as [14][3][8]. State of the art systems as [15] are commonly not a single system but the fusion of several of them. The performance improvement of a fused system is based on the fact that different systems provide different information about the speaker, and therefore errors committed by a certain system may be cancelled out by other systems. In fact, the potential benefits from fusion increase with the uncorrelation between the involved systems.

Some research has been done in the adaptation of fusion schemes to each user [16]. While there is some research effort in fusion of multiple biometrics systems [17][18], it is not very common within different speaker recognition systems, but improvements can be obtained from that adaptation.

The effect of fusion on error rates will be shown in sections 5 where specific systems are described and tested on reference data.

5. The ATVS speaker recognition system at NIST 2006 Speaker Recognition Evaluation

The evaluation is a speaker detection task, where the goal is to determine if a specified speaker is present in a given segment of conversational speech. Systems must provide a decision about the segment (T/F) and a confidence score.

The different task conditions are defined by a training and a test condition. For the core task, also called 1conv4w-1conv4w, 5 minutes of speech are provided (before silence removal) and other five minutes for testing.

ATVS systems have been tested in the core task, 8conv4w-1conv4w task and 1conv4w-10sec4w. For the 8conv4w training condition 8 speech segments of about 5 minutes are provied, and in the 10sec4w testing condition only 10 seconds of speech are available. Four different systems were developed and fused in different ways for the different tasks.

5.1. Development process

For the development process several partions of data were defined. For the training of the UBM model for all systems, data from NIST 2005, NIST 2004, SWITCHBOARD I and SWITCHBOARD II Extended-data task was used.

5.2. GMM system

A root UBM (needed for feature mapping[19]) was trained using 5 hours of channel- and gender-balanced speech after silence removal. Data from MIXER (NIST SRE 2004 and 2005), Switchboard I and Switchboard II was used. The UBM was trained using 1024 Gaussian mixtures and ML estimation via EM algorithm. Fourteen channel models (7 per gender) were adapted from the UBM in order to perform Feature Mapping. An average value of 2 hours of speech was used for each channel model training.

Target models were 1024 mixtures GMM, MAP adapted with one iteration (only means) from the 1024 root UBM. Only 5 Gaussian per frame were used in likelihood computations.

Score normalization was performed using Tnorm [20] and KL-Tnorm [21].

5.3. Acoustic SVM system

The acoustic SVM system uses a explicit normalized three degree polynomial expansion [22] followed by a decomposed Generalized Linear Discriminant Sequence Kernel (GLDS) as described in [5]. SVMTorch [5] was used to train the target models.

5.4. Prosodic system

The submitted prosodic systems was similar to the one described in section 3.4.

HTK 3.2.1 n-gram modeling tools [23] are used to train gender-dependent UBMs and the target-speaker models. Male and female training data from NIST 2005, NIST 2004, SWITCHBOARD I and SWITCHBOARD II Extended-data task has been used to train the male and female UBMs, respectively. N-gram modelling used trigrams. The target-speaker models are created by linear interpolation of the corresponding UBM (gender-dependent) and the speaker training data. The interpolation coefficients are set to 0.8 for the speaker data and 0.2 for the UBM. By including the general knowledge provided by the UBM into the target-speaker models, the amount of data needed for a good estimation of the trigram models is reduced. TNorm technique was applied for score normalization. Cohorts consist of 60 models from NIST SRE 2004 database.

5.5. Phonotactic system

The phonotactic system was built over three independent speech recognisers (English, Spanish and Basque) based on Hidden Markov Models (HMMs). The HMM topology is three- state left-to-right with no skips. The output pdfs of each state are modelled as GMMs. The number of Gaussians per state were adjusted on the NIST SRE05 data task corpus to minimize speaker recognition EER.

Two schemes have been used for feature extraction:

- The Advanced Distributed Speech Recognition Standard Front-End defined in the standard ETSI ES 202 050 [24].
- Sphinx [25] feature extraction system. This system is based on 13 MFCC coefficients along with delta and double delta coefficients and C0.

The set of English phone HMMs was trained on the TIMIT corpus. Since this corpus is microphone speech sampled at 16 kHz, audio was filtered to simulate the telephone channel and then downsampled it to 8 kHz. One Gaussian/state was used to model output pdfs. The set of Spanish phone HMM was trained on the Albayzin corpus. The same subsampling process as described above was applied for this case. Five gaussians/state were used to model output pdfs. Both systems use the ETSI ES 202 050 parameterizer. Basque SpeechDAT was used in order to train the Basque phone HMM set, modelling output pdfs with 20 gaussians per state. The parameterisation was performed using the Sphinx parameteriser. All sets were trained using HTK v3.2.1.

Acoustic-phonetic decoding (phone recognition) was performed with every recogniser on Switchboard I, Switchboard II, NIST SRE04, NIST SRE05, NIST SRE06 train and test files using HTK v3.2.1, the trained models and a null grammar. The only information used from the acoustic-phonetic decoding was the phone streams. The output phone streams were filtered to avoid repetitions of inter-word silences.

The Universal Background Phone Model (UBM) is a trigram language model trained with data from Switchboard I, Switchboard II, NIST SRE04 and NIST SRE05. Smoothing of unlikely trigrams was performed with absolute discounting. No cut-off factor was applied. A different UBM was used for each phonetic decoder. Speaker Phone Models (SPM_i) are created by linear interpolation of the 8 sides training material for each target speaker from NIST SRE06 training data. The interpolation factor (weight of the UBM) for this adaptation was adjusted on NIST SRE05 extended data task and was found to be optimal for an UBM weight of 0.7.

For score normalization Tnorm was applied using as cohort a gender dependent set of 60 models extracted from NIST SRE04.

5.6. The system for the 8conv4w-1conv4w task

High-level systems require lots of data for reliable parameter estimation, so they can only be applied to certain tasks. The 8conv4w-1conv4w task provides about 40 minutes of untranscribed speech for model training (prior to silence removal) and 5 minutes for testing.

The submitted system for this task was a combination of all available systems: GMM, SVM, prosodic system, phonotactic systems. All systems were fused using a linear SVM.

For the development stage, NIST05 data was used to train and test the fusion rule. This makes shown results a little bit optimistic, but as the used classifier is a very simple one (a linear SVM) the overfitting is not expected to be strong. With this setup, the obtained results are shown in figure 1. This figure shows that acoustic systems (GMM's and SVM's) perform much better than high-level systems, but the fusion of all improves results significantly.

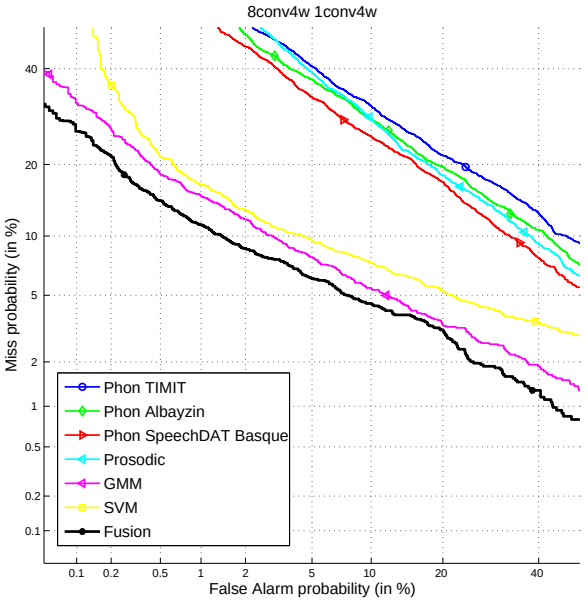


Figure 1: 8conv4w-1conv4w task involved systems results

6. Application of high-level speaker recognition techniques to language recognition

The language recognition field shares many techniques with the speaker recognition field. In [26] there is a detailed explanation of several techniques for language recognition.

First research efforts showed that high-level systems performed better than acoustic systems [26], but there have been important improvements in acoustic systems, as can be shown in [27] and [28]. Both techniques are applied in similar ways in both fields.

One of the most common technique for language recognition is an extension of phonotactic systems called Parallel Phone Recognition and Language Modelling (PPRLM). Basically it consists on the fusion of several phonotactic systems as described in section 3.3 related to phonetic decoders in several languages, not necessarily related to the target ones. Using the transcriptions, statistical grammars are applied and the scoring process is performed in the same way as for speaker recognition. Sum fusion is the most common applied fusion technique. In order to train each of the underlying phonetic recognisers, multilingual speech corpus are required, but they do not need to contain labelled speech in the target language. The only requirement is to have labelled in a certain number of language (and in the appropriate amount to train a phonetic recogniser). The most common corpus for this purpose is OGI Multilingual Telephone Speech [29]. This corpus has speech in 11 languages, and for 6 of them it contains labelled speech. Other corpora used to train language recognition systems is SpeechDAT. The main advantage of SpeechDAT over OGI Mul-

tilingual Telephone Speech is that the former is a much larger database, which seems to be important, as shown in [30].

Language recognition can also be performed using speech recognisers in the target languages. This technique is called Parallel Phone Recognition (PPR) and it basically consists of N parallel speech recognisers (one for each target language), each one providing a speech recognition score, used for decisions. To perform language recognition using PPR labelled speech in all target languages is required, in order to train the speech recognisers. As this may be difficult for certain languages, PPR is not very common.

6.1. The ATVS PPRLM speaker recognition system at NIST 2005 Language Recognition Evaluation

For the 2003 evaluation, twelve target languages were defined (Arabic, English, Farsi, French, German, Hindi, Japanese, Korean, Mandarin, Spanish, Tamil and Vietnamese). Test material was extracted mainly from the CallFriend corpus from LDC, with some test segment from other corpora.

In the 2005 evaluation, the target languages set was reduced to seven (English, Hindi, Japanese, Korean, Mandarin, Spanish and Tamil) along with some dialectal variations (American/Indian English, Mainland Mandarin and Taiwanese Mandarin). A new database was collected, including a big amount of cellular data, that was not present in previous evaluations data.

The testing segments have nominal durations of 30, 10 and 3 seconds.

The full description of submitted system along with the details of the development process can be found in [31].

The best submitted system was a fusion of two PPRLM systems (as described above) based on phonetic recognisers trained with the OGI Multilingual Telephone Speech corpus. Both PPRLM systems were the fusion of 6 language recognisers based on phonetic recognisers of Mandarin, German, Japanese, Spanish, Hindi and English, using in all cases trigrams. The two PPRLM systems were similar but differ in the the number of Gaussian used to model HMM state output pdf's in the phonetic recognisers. To normalise scores, Tnorm was used. (This system appears in plots as ATVS1).

Other system was submitted, but consists only of a single PPRLM system, built using 10 Gaussian per state. (This system appears in plots as ATVS2).

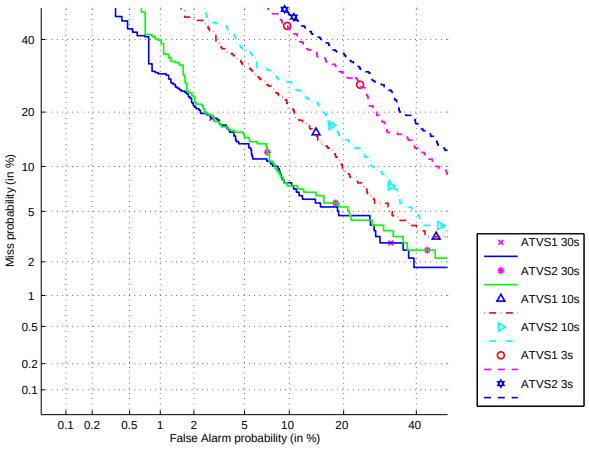


Figure 2: ATVS PPRLM results tested on a NIST 2003 LRE subset involving only NIST 2005 LRE target languages.

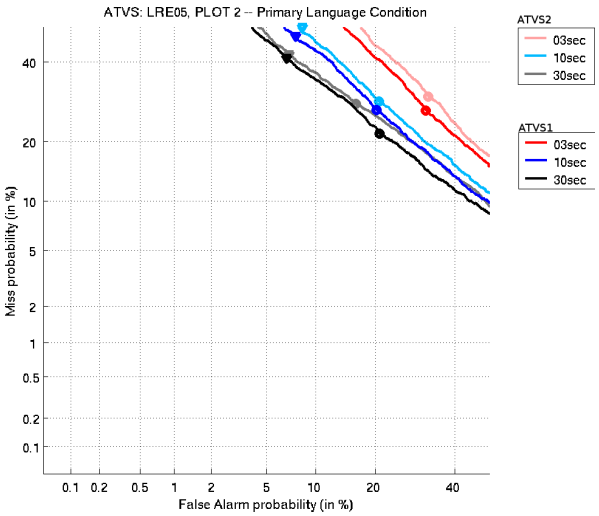


Figure 3: ATVS PPRLM submitted systems results at NIST 2005 LRE.

Figure 2 shows the results of the development process, with testing data coming from 2003 evaluation data, but restricted to the 2005 evaluation target languages. Figure 3 shows the results obtained in the 2005 evaluation. As aforementioned, in 1996 and 2003 editions, evaluation data was mainly extracted from the CallFriend database. But for the 2005 evaluation a new corpus was collected by OHSU. This corpus includes a large proportion of cellular data, which is not present neither in the CallFriend database nor in the NIST 2003 Evalu- ation data. The aforementioned systems were submitted to the 2005 evaluation, performing as shown 3. Those new channel conditions could explain the similar degradation in re- sults, by a factor of two or even more, that affected all LRE05 participants, from dev to eval data.

7. Conclusions

Several techniques are available in the field of speaker recognition. They are usually divided into acoustics (GMM, SVM) and high-level (phonotactic and prosodic) systems. Results obtained by each type of systems are quite different, and acoustics systems clearly outperform high-level systems. The main purpose of high-level systems is to be combined, in order to improve results. The kind of knowledge, parameters and modelling techniques these systems use are quite different from one to another (discriminative vs. generative approaches, phones, energy), thus providing a chance for fusions, which is performed at score level, providing important improvements in recognition rates.

8. References

[1] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[2] "Speaker recognition evaluations," http://www.nist.gov/speech/tests/spk/.

[3] G. Doddington, "Speaker recognition based on idiolectal differences between speakers," in *Proceedings of EUROSPEECH*, 2001, vol. 4, pp. 2517–2520.

[4] D. Reynolds et al., "Supersid project: Exploiting high-level information for high-accuracy speaker recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2003.

[5] W.M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proceedings of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2002, pp. 161–164.

[6] A. G. Adami et al, "Modeling prosodic dynamics for speaker recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2003, vol. IV, pp. 788–791.

[7] J. Gonzalez-Rodriguez et al, "On the use of high-level information for speaker recognition: the atvs-uam system at nist sre 05," *IEEE Aerospace and Electronic Systems Magazine*, (in press).

[8] D. Garcia-Romero, J. Fierrez-Aguilar, J. Ortega-Garcia, and J. Gonzalez-Rodriguez, "Support vector machine fusion of idiolectal and acoustic speaker information in spanish conversational speech," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2003, pp. 229–232.

[9] A. Moreno, D.Poch, A.Bonafonte, E.Lleida, J.Llisterri, and C.Nadeu J.Marino, "Albayzin speech database: Design of the phonetic corpus," *Proc. Eurospeech'93*, vol. vol. 1, pp. 175–178, 1993.

[10] Asmaa El Hannani, Doroteo T. Toledano, Dijana Petrovska-Delacrétaz, Alberto Montero-Asenjo, and Jean Hennebert, "Using data-driven and phonetic units for speaker verication," in *Proceedings of Odyssey06: The speaker and language recognition workshop*, 2006.

[11] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek, "High-level speaker verification with support vector machines," in *Proc. of ICASSP*, 2004, pp. 73–76.

[12] Jean-Luc Gauvain, Abdel Messaoudi, and Holger Schwenk, "Language Recognition Using Phone Lattices," Jeju Island, October 2004, pp. 1283–1286.

[13] Andrew O. Hatch, Barbara Peskin, and Andreas Stolcke, "Improved phonetic speaker recognition using lattice decoding," in *Proc. of ICASSP*, 2005, pp. 165–168.

[14] William M. Campbell, Douglas A. Reynolds, and Joseph P. Campbell, "Fusing discriminative and generative methods for speaker recognition: experiments on switchboard and NFI/TNO field data," in *Odyssey: The Speaker and Language Recognition Workshop*, 2004, pp. 41–44.

[15] D. A. Reynolds et al., "The 2004 MIT Lincoln Labs speaker recognition system," in *Proceedings of ICASSP*, 2005, pp. 177–180.

[16] Julian Fierrez-Aguilar et al., "Speaker verification using adapted user-dependent multilevel fusion," in *Proc. 6th IAPR Intl. Workshop on Multiple Classifier Systems, MCS, Springer LNCS-3541*, 2005, pp. 356–365.

[17] J. Kittler et al., "On combining classifiers," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226–239, 1998.

[18] J. Fierrez-Aguilar et al., "Bayesian adaptation for user-dependent multimodal biometric authentication," *Pattern Recognition*, vol. 38, no. 8, August 2005.

[19] D. A. Reynolds, "Channel robust speaker verification via feature mapping," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2003, vol. 2, pp. 53–56.

[20] Roland Auckenthaler, Michael Carey, and Harvey Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, pp. 42–54, 2000.

[21] D. Sturim and D. A. Reynolds, "Speaker adaptive cohort selection for tnorm in text-independent speaker verification," in *Proc. of ICASSP*, 2005.

[22] V. Wan and W. Campbell, "Support vector machines for speaker verification and identification," 2000.

[23] Steve Young et al., *The HTK Book*, Dec. 2002.

[24] ETSI ES 202 050 (v1.1.3), "Speech processing, transmission and quality aspects (STQ); Distributed speech recognition; Advanced front-end features extraction algorithm; Compression algorithms.," http://www.etsi.org/.

[25] "Sphinx systems homepage," http://cmusphinx.sourceforge.net/html/cmusphinx.php.

[26] Marc A. Zissman, "Comparison of Four Approaches to Automatic Language Identification of Telephone Speech," *IEEE Transactions on Speech and Audio Processing*, pp. 31–44, Jan. 1996.

[27] Pedro A. Torres-Carrasquillo et al., "Approaches to language identification using gaussian mixture models and shifted delta cepstral features," in *Proc. of ICSLP*, 2002.

[28] W. M. Campbell et al., "Language recognition with support vector machines," in *Proceedings of Odyssey04: The speaker and language recognition workshop*, 2004, pp. 285–288.

[29] OGI, "OGI Multilanguage Telephone Speech v1.2," http://cslu.cse.ogi.edu/corpora/mlts/.

[30] Matejka Pavel, Schwarz Petr, Cernocky Jan, and Chytil Pavel, "Phonotactic language identification using high quality phoneme recognition," in *Interspeech'2005 - Eurospeech - 9th European Conference on Speech Communication and Technology*, Lisbon, 2005, pp. 2237–2240.

[31] Alberto Montero-Asenjo, Doroteo T. Toledano, Javier Gonzalez-Dominguez, Joaquin Gonzalez-Rodriguez, and Javier Ortega-Garcia, "Exploring PPRLM performance for nist 2005 language recognition evaluation," in *Proceedings of Odyssey06: The speaker and language recognition workshop*, 2006.

WHERE ARE THE SPEECH PRODUCTION MODELS IN OUR SPEECH PROCESSING SYSTEMS?

Richard Rose

Department of Electrical and Computer Engineering
McGill University
Montreal, Quebec

RESUMEN

The conjecture that automatic speech recognition and speech synthesis systems should profit from speech production-oriented representations of speech has motivated research in a number of areas over several decades. The focus of this research has ranged from attempts to construct complete multidimensional articulatory models to the development of systems that detect and integrate acoustic-phonetic features in more traditional ASR systems. It is believed by many researchers that this class of approaches may provide the only solution to some of the fundamental problems in existing state of the art systems. These problems include the limitations imposed by the use of canonical dictionaries in ASR and by the inability to maintain robust performance as portions of the speech spectrum are masked by noise. The impact of these limitations has become ever more apparent as academic and industrial laboratories focus on task domains that evoke increasingly more spontaneous speech in increasingly difficult acoustic environments. It is generally agreed, however, that the major advances in ASR have come from the development of more powerful statistical modeling formalisms which incorporate only minimal speech production or perceptual knowledge. This talk will attempt to explain why speech production-oriented approaches are important and at the same time explain why the development of efficient formalisms that directly incorporate knowledge of actual articulatory dynamics for ASR is extremely difficult. In light of these difficulties, it will address the progress and potential success of efforts to incorporate speech production knowledge into existing statistical modeling formalisms.

Biografía: Richard Rose

Richard Rose received B.S. and M.S. degrees in Electrical Engineering from the University of Illinois, and Ph.D. E.E. degree from the Georgia Institute of Technology. From 1980 to 1984, he was with Bell Laboratories working on signal processing and digital switching systems. From 1988 to 1992, he was with MIT Lincoln Laboratory working on speech recognition and speaker recognition. He was with AT&T from 1992 to 2003, specifically in the Speech and Image Processing Services Laboratory at AT&T Labs - Research in Florham Park, NJ after 1996. Currently, he is an associate professor of Electrical and Computer Engineering at McGill University in Montreal, Quebec. Professor Rose served as a member of the IEEE Signal Processing Society Technical Committee on Digital Signal Processing from 1990 to 1995, and was on the organizing committee of the 1990 and 1992 DSP workshops. He was elected as an at large member of the Board of Governors for the Signal Processing Society during the period from 1995 to 1997. He served as an associate editor for the IEEE Transactions on Speech and Audio Processing from 1997 to 1999 and is currently a member of the editorial boards for the Speech Communication Journal and for the IEEE SPS Newsletter. He was a member of the IEEE SPS Speech Technical Committee (STC) from 2003 to 2005 and was founding editor of the STC Newsletter. Most recently, he was the Co-chair of the IEEE 2005 Workshop on Automatic Speech Recognition and Understanding.

JOINT TRAINING OF CODEBOOKS AND ACOUSTIC MODELS IN AUTOMATIC SPEECH RECOGNITION USING SEMI-CONTINUOUS HMMS

Albino Nogueiras, Mónica Caballero, José B. Mariño

TALP Research Center
Universitat Politècnica de Catalunya (UPC). Barcelona, Spain.
{albino,monica,canton}@gps.tsc.upc.edu

RESUMEN

In this paper, three different techniques for building semi-continuous HMM based speech recognisers are compared: the classical one, using Euclidean generated codebooks and independently trained acoustic models; jointly reestimating the codebooks and models obtained with the classical method; and jointly creating codebooks and models growing their size from one centroid to the desired number of them. The way this growth may be done is carefully addressed, focusing on the selection of the splitting direction and the way splitting is implemented. Results in a large vocabulary task show the efficiency of the approach, with noticeable improvements both in accuracy and CPU consumption. Moreover, this scheme enables the use of the concatenation of features, avoiding the independence assumption usually needed in semi-continuous HMM modelling, and leading to further improvements in accuracy and CPU.

1. INTRODUCTION

Along the rest of this Introduction, an overview of semi-continuous Hidden Markov Models (HMMs), and a description of the experimental environment can be found. Follows, on Section 2, a description and comparison of the three alternatives for training codebooks and acoustic models considered. Finally, on Section 3 some conclusions are sketched.

1.1. Semi-continuous HMMs

An HMM is a collection of *states*. Each frame of voice can be in one and just one of the states at any time. Each HMM is formed of two different parts: a *transition matrix*, along with the *initial distribution*, and a set of *emission probability functions*. The transition matrix of an S-state HMM is an $S \times S$ matrix. Each element of the matrix represents the probability of moving from one of the states to another. The initial distribution represents the probability of being at each of the states at the beginning of the utterance. The transition matrix and initial distribution are common to all kind of HMMs, and are fundamental in the time warping capabilities of the method. Nevertheless, their exact value has shown little influence on the final results.

Each state in the model also contains an emission probability function that provides the probability with which this state generates any frame. There are several ways of defining emission probability functions, but the most usual way is by means of mixtures of Gaussian densities, trained with the expectation maximisation algorithm. Yet two alternatives are possible: continuous HMMs, and semi-continuous HMMs.

In continuous HMMs each state is modelled with a mixture of private Gaussians with diagonal covariance. As the number of Gaussians rapidly grows when the number of units and/or states grows, it is usual to tie together groups of them. This means that several states of several units share some of the Gaussian distributions, but not the mixture weights, which are still private. In semi-continuous HMMs all the Gaussians are shared, and form a vector quantifier. For each frame, the probability density of all the Gaussians is calculated, and a score vector is formed out of the highest of them. The probability of this frame being in a given state is equal to the sum-product of the score vector that represents the frame, and the mixture vector that represents the state.

Being $o = o_1 o_2 ... o_T$ an utterance of length T, λ the acoustic model corresponding to this utterance, we define $\mathcal{G}^k(o_t)$ as the value of the k^{th} Gaussian of the codebook for frame o_t, and c_{sk} the mixture weight of this Gaussian at state s. The likelihood of o_t at state s is given, for semi-continuous HMMs, by:

$$P(o_t/q_t = s) = \sum_{k=1}^{K} c_{sk} \mathcal{G}^k(o_t) \qquad (1)$$

The mixture weights c_{sk} are usually trained using the expectation maximisation algorithm. It holds that the likelihood is maximised at each iteration if the weights are updated with the value of the ratio between the expected number of times at state s and observing symbol k, and the expected number of times at state s. Defining the contribution of frame o_t to the k^{th} symbol at state s, $C_{sk}(o_t)$, as:

$$C_{sk}(o_t) = \alpha_t(s)\frac{c_{sk}\mathcal{G}^k(o_t)}{\sum_{k'} c_{sk'}\mathcal{G}^k(o_t)}\beta_t(s) \qquad (2)$$

Where $\alpha_t(s)$ is the *forward probability*, equal to the probability of o_t at state s given the model and all the frames of o from the beginning of the utterance until the current one: $\alpha_t(s) = P(o_1 o_2 \cdots o_{t-1} o_t, q_t = s/\lambda)$; and $\beta_t(s)$ is the *backward probability*, equal to the probability of o_t at state s given the model and all the following frames until the end of the utterance: $\beta_t(s) = P(o_{t+1} o_{t+2} \cdots o_T, q_t = s/\lambda)$. We have:

$$\bar{c}_{sk} = \frac{\sum_{t=1}^{T} C_{sk}(o_t)}{\sum_{k=1}^{K}\sum_{t=1}^{T} C_{sk}(o_t)} \qquad (3)$$

1.2. System Overview

Along the rest of the paper some alternatives of codebook and acoustic models training will be tested. The experimental framework within which this will be done is the recognition of a large vocabulary continuous speech task using RAMSES [1], the speech recognition system developed at UPC. Its main features are:

- Speech is windowed every 15ms with 30ms frame length. Each frame is parameterised with 12 Mel-frequency cepstral coefficients (MFCC) and energy, plus their first and second derivatives. Mean subtraction is applied to static parameters. A liftering window of length 22 is applied to the cepstral coefficients.
- Spectral parameters are quantified to 512 diagonal Gaussian centroids; energy ones are quantified to 128. Only the 32 highest scores are considered for spectral parameters; 8 in the case of energy ones.
- Demiphones are used as acoustic unit [2]. Each of them represents half phoneme with its closest neighbour explicitly considered. Contexts across words are not. 1000 demiphones are selected using a minimum entropy decision tree.
- Each demiphone is modelled with three states. The first and second states are possible entries to the model, and the second and third states are possible exits from it.
- Several pruning configurations are tested at recognition time. On the final results, only those leading to the highest accuracy at each level of CPU consumption are kept.

1.3. The Recognition Task

The task used is the recognition of free speech telephonic dialogues in a tourism information retrieval semantic domain. Both the training and the test material used were recorded inside the scope of the European Commission funded project LC-STAR [3], which is devoted to collecting lexica and corpora for automatic speech-to-speech translation.

The corpus is composed of 211 dialogues (422 different speakers), of which 16 define the standard test material. The remaining 195 dialogues are used for training the acoustic models. In mean, each dialogue lasts 9 minutes and is composed of 45 utterances, each of about 30 words. In total, 8418 utterances (29h45m, 230,000 words) were used for training, and 1040 utterances (3h40m, 28,000 words) were used as test material. Neither segmentation nor silence detection were carried by hand.

The objective of the task is recognising the words pronounced during the tourism information retrieval dialogues. The vocabulary of the task is composed of 7466 words, and the grammar perplexity is around 70. Both vocabulary and grammar were obtained from the training material. The test material presents an out-of-vocabulary rate over 1 %. For the results presented in this paper, the margins for a 95 % confidence in the error rates is around half a point.

2. THE DIFFERENT ALTERNATIVES

2.1. The Classical Approach: Using Euclidean Codebooks

In their classical implementation, codebooks and HMMs are trained in two phases: first we generate the codebooks, and then the HMMs are trained using them. The generation of the codebooks is done using the Lloyd algorithm in order to minimise the mean Euclidean distortion. The Gaussians of the final codebook are modelled with the means and variances of the clusters obtained in this way. During all the process of construction of the codebook each frame is quantified discretely, i.e. only to the closest centroid. At training and recognition times each frame is quantified to all the centroids in the codebook although, as just the highest are usually relevant, the lowest are usually discarded in order to alleviate the CPU load.

This approach has several advantages. Mainly its robustness, simplicity and low CPU consumption. Its main drawback is that it relies heavily on the Euclidean distance defined on the parameter space. This means that this space must be designed in a way that this distance is meaningful in terms of phonetic similarity. For instance, Euclidean distance is sensitive to the individual variance of each component in the feature vector, but this variance does not necessarily account for phonetic information, it may just account for a greater magnitude of this component respect to the rest of the vector. For instance, liftering of the cepstral coefficients leads to improvements of more than half a point in the experiments herein presented, and it is just a scaling of the cepstral coefficients, that should be neutralised by the Gaussian modelling.

A way to improve the effectiveness of the Euclidean

codebook modelling is to linearly transform the feature space in such a way that differences between components really reflect phonetic differences. Several strategies have been proposed—PCA, LDA, etc.—but either their implementation is too difficult, or it is not clear the benefits we may get. Actually, any linear transformation of the space should be almost completely voided by the Gaussian mixture modelling. If we used full covariance Gaussians, such a transformation would lead to new means and variances, but the probabilities given to each point of the space would be the same. In the case of using diagonal covariance Gaussians, the linear transformation also changes the component independence assumption, but this change should not have a remarkable effect if convenient estimation of the Gaussians is done. It is not the case if Euclidean codebooks are used, because they assume covariance equal to the identity.

The error rate achieved with this strategy at different CPU consumption levels is plotted on top of Fig. 1 (**classic**). This kind of system was soon superseded because of the mentioned limitations. We include it in this paper because it presents some benefits that make it interesting for certain applications. For instance, this kind of training is useful when different tasks are added to the system, and we do not want to train again all the rest of tasks. The use of a common, blind codebook enables us to simply train the mixture weights of the task specific models, without affecting the rest of them. In the rest of systems studied in this paper, the addition of a new task represents training again the acoustic models of the rest of tasks. Besides, the CPU consumption of the training phase is much lower than for the rest of alternatives—using Euclidean codebooks, the whole system used in this paper is trained in less than one day; the best results presented in this paper needed almost three weeks—.

2.2. The Baseline: Joint Re-Estimation of Codebooks and HMMs.

A first approach to mitigate the *blind* nature of the Euclidean distance based quantisation used in the classical system is the joint re-estimation of codebooks and HMMs. In this case, an initial codebook is used to train initial acoustic models as in the classical system, but then another optimisation process is performed where both the acoustic models and the Gaussians are re-estimated using expectation maximisation. The mixture weights re-estimation is performed in the same way as before. The formulae for reestimating the means and variances of the Gaussians are, using the contributions defined in (2):

$$\bar{\mu}_k = \frac{\sum_{s=1}^{S}\sum_{t=1}^{T} C_{sk}(o_t)\cdot o_t}{\sum_{s=1}^{S}\sum_{t=1}^{T} C_{sk}(o_t)} \quad (4)$$

$$\bar{\sigma}_k^2 = \frac{\sum_{s=1}^{S}\sum_{t=1}^{T} C_{sk}(o_t)\cdot (o_t-\mu_k)(o_t-\mu_k)^t}{\sum_{s=1}^{S}\sum_{t=1}^{T} C_{sk}(o_t)} \quad (5)$$

Equation (5) provides the full covariance matrix of each Gaussian distribution, in the experiments done in this paper, as well as in most of usual speech recognition systems, just the diagonal components of the covariance are considered, what is equivalent to supposing component independence.

Results using joint re-estimation of codebooks and HMMs are better than the ones achieved using the classical method in all cases (see Fig. 1, **joint**). Error rate decreases at all CPU levels by about two points, and it is clear that the system performs faster than the classical one.

Although the used of re-trained Gaussians, with means and variances adjusted to the shape of the feature space, should make the system immune to any scaling of the components, the use of liftering carries on to have a noticeable effect, of about half a point in the error rate. We think that this is due to the difficulty of moving the centroids from their original place. Somehow, the system cannot forget the initial codebook.

2.3. The alternatives: HTK/Julius

The above mentioned baseline system is the one that we have been using at UPC as a reference in our research work during the last times. It has been used in the recognition of a variety of tasks (from isolated words to continuous speech), of recording conditions (from recording studio to noisy GSM environments), and of languages (some 8 different languages, plus several dialectal variants of Spanish). In many of these tasks, parallel independent teams carried the same task using alternative recognition systems (HTK/Julius, Sphinx, Janus, etc.) Although direct comparison is very difficult, results were in general quite similar, particularly when CPU consumption limitations appeared. For instance, an independent team carried out the recognition of the task presented in this paper, with similar parameterisation and language model, but using HTK for building the acoustic models [4], and Julius for performing the recognition [5]. Their goal was to minimise the error rate at around 1'5 times real time on a Pentium Xeon at 3GHz.

The best result, 44'5 % at 1'53 times real time, was achieved when 4,000 context dependent triphones were modelled with 32 tied Gaussians per state. This result is also plotted in Fig. 1 (**HTK/julius**), and falls between the results of the classical system and our baseline [6].

2.4. Joint Growth of Codebooks and HMMs.

An alternative to first building an Euclidean codebook, then training initial acoustic models, and finally reestimating both codebook and models together is to grow Codebooks and HMMs together. The idea is similar to the Lloyd algorithm, and has been applied in the training of continuous HMMs:

1. An initial one centroid codebook is generated.

2. Models and codebook are trained using the expectation maximisation algorithm.

3. Each centroid in the codebook, as well as the corresponding weights in the models, is splitted into two.

4. The process is iterated from step 2 until the desired codebook size is reached.

The key question is how to split codebooks and models. In our work we divide the problem in two parts: selecting the direction of splitting, and performing the splitting itself.

2.4.1. *Selection of the splitting direction*

The main objective of the splitting is to generate two new centroids out of each previous one. The usual way of doing this is selecting the direction of splitting and moving forward and backward from the mean of the old centroid along this direction. In the classical Lloyd algorithm the direction of splitting is random, because the Euclidean discrete quantisation is very robust and this initialisation is not much relevant. In the joint growth of codebooks and HMMs we wish the split to be as effective as possible, while minimising the perturbation of the models at each step. A bad election of the splitting direction is readily passed to the HMMs weights, and the expectation maximisation algorithm may not be able to move the centroids to convenient positions.

The addressed problem is that of finding the direction of splitting that provides the maximum gain of information. Different criteria may be defined to find this direction [7]. A convenient way is selecting the direction where the separation between classes is maximised. In our case, the different classes can be defined by the states of the HMMs. In a space where the variance is equal in all the directions, the variance of the means of the classes is indicative of their separability using linear classification.

For each centroid, two different covariance matrices can be built using the contributions defined in (2): the overall covariance, $\mathbf{G}_k$ is the covariance matrix of all the contributions of each centroid; and the interclass covariance, $\mathbf{I}_k$, the covariance of the means of each class:

$$\begin{aligned} \mu_k &= \frac{\sum_{s=1}^{S}\sum_{t=1}^{T} C_{sk}(o_t)\cdot o_t}{\sum_{s=1}^{S}\sum_{t=1}^{T} C_{sk}(o_t)} \\ \mathbf{G}_k &= \frac{\sum_{s=1}^{S}\sum_{t=1}^{T} C_{sk}(o_t)\cdot (o_t-\mu_k)(o_t-\mu_k)^t}{\sum_{s=1}^{S}\sum_{t=1}^{T} C_{sk}(o_t)} \end{aligned} \quad (6)$$

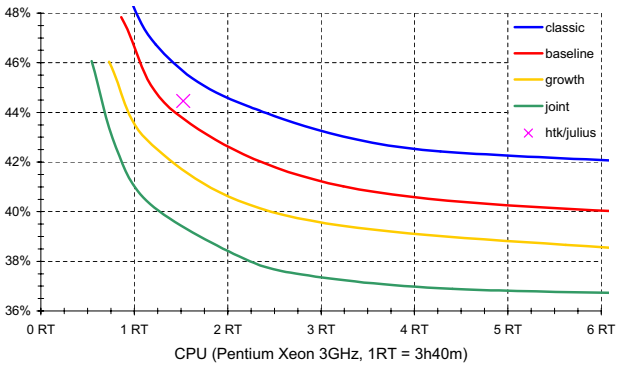


Figura 1. Word error rate in the recognition of the LC-STAR Spanish tourism retrieval task versus the CPU consumption.

$$\begin{aligned} \mu_{sk} &= \frac{\sum_{t=1}^{T} C_{sk}(o_t)\cdot o_t}{\sum_{t=1}^{T} C_{sk}(o_t)} \\ f_{sk} &= \sum_{t=1}^{T} C_{sk}(o_t) \\ \mathbf{I}_k &= \frac{\sum_{s=1}^{S} f_{sk}\cdot(\mu_{sk}-\mu_k)(\mu_{sk}-\mu_k)^t}{\sum_{s=1}^{S} f_{sk}} \end{aligned} \quad (7)$$

The product $\mathbf{G}_k^{-1/2T}\,\mathbf{I}_k\,\mathbf{G}_k^{-1/2}$, is equal to the covariance of the means in a variance-normalised space, so it reflects how much of the total variance is due to the separation between classes. In each direction, its value is comprised between zero and one, meaning zero that none of the variance in this direction is due to the separation between classes, and being close to one if most of the variance is due to it. The most effective linear split of the cluster, in the sense of reducing this variance, is following the hyperplane normal to the eigenvector of highest eigenvalue.

Notice that this product is insensitive to any linear transformation applied to the whole feature space, because it will be appear in all the terms of the product and cancel out. This includes any scaling of the coefficients, such as liftering, principal component analysis and linear discriminant analysis.

2.4.2. *Performing the splitting*

Chosen the direction of splitting, say $\mathbf{d}_k^*$, performing it is not a trivial task. In previous experiments centroids were splitted by generating two new centroids whose variance and mixture weights were the same as before, and the means were the previous one moved a certain amount forward and backward following the selected splitting direction. The problem was that, if a small movement was done, codebooks and models were too similar before and after the splitting, and the method converged to too close

centroids with little discrimination capabilities. On the contrary, if the movement was large, models and codebooks were too much disturbed at each step, and even convergence was compromised.

In order to maximally separate the centroids without perturbing too much codebooks and models, an additional step was introduced in the training procedure. In this step the contributions needed in (3) and (4) for reestimating the parameters of the grown codebooks and models are calculated as before, using the original size system. The difference is that the contribution is assigned to one of the two new centroids using linear classification along the splitting direction. In this way the separation between the two new centroids is maximised because each contribution is assigned to one or the other. At the same time, the system is minimally perturbed because the contributions are calculated using the original parameters, and the splitting of one cluster does not interfere with the rest of clusters.

$$\begin{aligned} \eta_k &= \mathbf{d}_k^{*t}\mu_k \\ \delta_{2k-1}(o_t) &= \begin{cases} 1 & \mathbf{d}_k^{*t}o_t > \eta_k \\ 0 & \text{otherwise} \end{cases} \\ \delta_{2k}(o_t) &= 1-\delta_{2k-1}(o_t) \end{aligned} \tag{8}$$

$$\begin{aligned} \bar{c}_{s2k} &= \frac{\sum_{t=1}^{T}\delta_{2k}(o_t)C_{sk}(o_t)}{\sum_{k=1}^{K}\sum_{t=1}^{T}\delta_{2k}(o_t)C_{sk}(o_t)} \\ \bar{\mu}_{2k} &= \frac{\sum_{s=1}^{S}\sum_{t=1}^{T}\delta_{2k}(o_t)C_{sk}(o_t)\cdot o_t}{\sum_{s=1}^{S}\sum_{t=1}^{T}\delta_{2k}(o_t)C_{sk}(o_t)} \\ \bar{\sigma}^2_{2k} &= \frac{\sum_{s=1}^{S}\sum_{t=1}^{T}\delta_{2k}(o_t)C_{sk}(o_t)\cdot (o_t-\mu_k)^t(o_t-\mu_k)}{\sum_{s=1}^{S}\sum_{t=1}^{T}\delta_{2k}(o_t)C_{sk}(o_t)} \end{aligned} \tag{9}$$

The formulae for $\bar{c}_{s2k-1}$, $\bar{\mu}_{2k-1}$, and $\bar{\sigma}^2_{2k-1}$, are similar as above for $\bar{c}_{s2k}$, $\bar{\mu}_{2k}$, and $\bar{\sigma}^2_{2k}$, except that $\delta_{2k-1}(o_t)$ should be used instead of $\delta_{2k}(o_t)$.

2.4.3. Experimental results using joint growth

The system built jointly growing codebooks and models achieved the lowest error rate of all the tested so far, reducing in more than one and a half points the error rate of the baseline, while also reducing even more the CPU consumption (see Fig. 1, **growth**). As expected, the system is now completely immune to liftering. Moreover, the result is almost identical whether linear discriminant analysis is used or not—we do not dispose of results using LDA on the previously described systems—.

2.5. Training the Whole Feature Vector Using Joint Growth of Codebooks and HMMs.

The fact that joint training of models and codebooks is able to increase the separability of the acoustic classes—as reflected by the improvement in both accuracy and speed—, and that it is immune to linear transformations of the feature space, encouraged us to undertake a rather difficult task: training semi-continuous models of high dimension features, namely the concatenation of the different energy and spectral features into one vector. The main implication of using this kind of feature is that different elemental features are no longer supposed to be independent.

Historically, using the whole vector with semi-continuous HMMs was impossible, or almost, because constructing a blind quantifier is increasingly difficult as the number of components grows. For instance, in order to ensure that each component in a vector of size P is split into at least two clusters, the size of the codebook should be 2^P. This is 4096 for a typical cepstral vector of size 12, but grows up to 500×10^9 when the whole vector of 39 components is used.

This problem does not appear in continuous HMMs. because they do not use a blind codebook, but phonetically guided mixtures instead. The same holds for the herein proposed algorithm—certainly, our system is very similar to a continuous system with all the Gaussians tied—. The results achieved when the whole vector was added to the former six ones are plotted on bottom of Fig. 1 (**whole**). As it stands out, it is not only possible to train such kind of information, but it still improves significantly both accuracy and speed. This is the best result we have ever achieved in this task using RAMSES or any other system.

3. DISCUSSION

The results presented in this paper clearly show that the classical ways of training semi-continuous HMMs can be improved by jointly growing models and codebooks from scratch. This technique also enables us to use high dimension feature vectors such as the concatenation of elementary features. The high performance obtained in this way is not only interesting from a recognition performance point of view. It is also interesting because the fact that a codebook of such dimension can be built opens a lot of possibilities in several other fields of speech processing such as text to speech synthesis or speech coding.

4. BIBLIOGRAFÍA

[1] A. Bonafonte et alter, "RAMSES: el sistema de reconocimiento del habla continua y gran vocabulario desarrollado por la UPC," in *VIII Jornadas de Telecom I+D*, 1998.

[2] J.B. Mariño et alter, "The demiphone: a new contextual subword unit for continuous speech recognition," *Speech Communication*, vol. 32, no. 3, pp. 187–197, October 2000.

[3] "LC-STAR: Lexicon and corpora for speech to speech translation," IST-2001-32216. http://www.lc-star.com.

[4] S. Young et alter, *The HTK Book 3.2*, http://htk.eng.cam.ac.uk, 2002.

[5] A. Lee et alter, "Julius: An open source real-time large vocabulary recognition machine," in *Proceedings of EUROSPEECH*, 2001, pp. 1691–1694.

[6] F.J. Punzano, *Sistema de Reconocimiento del Habla de Gran Vocabulario Basado en Julius. Entrenamiento mediante HTK*, Universitat Politècnica de Catalunya, 2006.

[7] T. Parsons, *Voice and Speech Processing*, McGraw-Hill, 1986.

AUTOMATIC PHONETIC SEGMENTATION AND LABELLING OF SPONTANEOUS SPEECH

Luis Pinto Coelho[1] *y Daniela Braga*[2]

[1] Instituto Politécnico do Porto, [2] Universidade da Coruña

ABSTRACT

In this paper a tool for automatic segmentation and labeling of spontaneous speech is presented. It is developed and specially tuned for the European Portuguese (EP) language but simple changes are needed to convert it to other languages. The main purpose of this system is to quickly produce a high quality output of phonetic labels and related time boundaries using as input the speech signal only. Our motivation was the development of a tool that could help to create new voices for TTS systems without and special previous text selection concerns. A quick talk or a quick reading of a randomly selected text are the targets for the use of our system. The evaluation of the presented system gave up to 88.3% of accuracy in phonetic alignment considering a 20ms temporal error.

1. INTRODUCTION

The quick development of speech systems created a demand for new and better speech databases (using new voices, new dialects, new special features to consider, etc.), often with phonetic level annotation information (and others). This trend re-enforces the importance of automatic segmentation and annotation tools because of the drastic time and cost reduction in the development of speech corpora even when some little human action is needed. For the Portuguese language, in spite of its world wide usage, the number of available resources is still low and unadjusted, justifying this way an effort on the improvement and development of databases and tools. An accurate method of automatic phonetic transcription can thus facilitate the development of TTS and ASR systems for novel material, both within and across languages, as well as increase robustness with respect to acoustic interference and variation in speaking style and pronunciation.

In section 2 an overview of the system is given including a brief description of the used corpus.

In section 3 the final results and some conclusions are presented (as a whole due to available space and esthetical considerations).

2. SYSTEM DEVELOPMENT

The system's global structure is shown in figure 1. The annotation tool uses as input a speech signal that will be decoded, on a first stage, by an HMM based recognition engine (other authors try to test several phonetic transcriptions obtained from orthography). The resulting phoneme sequence and the given audio will then be processed by a segmentation engine, also HMM based. Both engines have independent parameters but share a common phoneme inventory and a set of language and acoustic models previously trained. The use of an HMM framework gives the possibility of obtaining in a single stage, both phonetic sequence and temporal boundaries however, as will be shown, the demands of each task are different and require independent adjustments to effectively produce the desired results. Hence the use of two stages is mandatory.

The speech signal, labels and related boundaries are finally analysed as a whole in a refinement stage for producing high quality results. The duration of the phones, the acoustical/frequential positioning of time boundaries, the voiced/unvoiced behaviour, the pitch periods, among others aspects are considered in this final stage. After refinement the system outputs a list of phones with well defined temporal marks for begin and end.

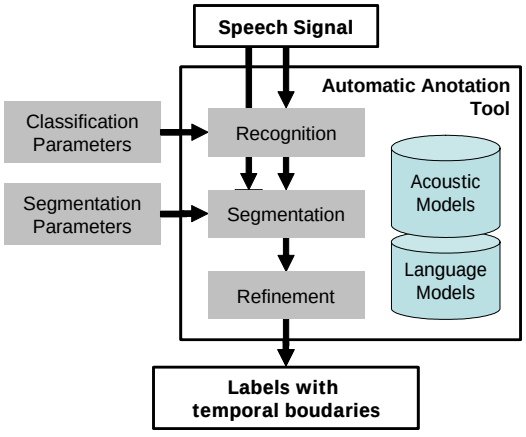


Figure 1. System overview.

2.1. Used Corpora

For system training, the FEUP/IPB speech corpus was used on a first step. This is a high quality database, recorded using professional equipment in a professional studio. It has nearly 100 minutes of speech from a single professional speaker and has phonetic level annotation [4].

On a second step, to increase robustness, the ProGmatica corpus [5] has been used. ProGmatica consists of broadcasted television materials, such as interviews, talk shows and political debates, in which natural spontaneous speech occurs. It was developed to study speech acts, using the well known John Searle's topology, and it is still an ongoing work. Around 20 minutes of phonetically hand labelled speech were selected considering the largest variety of voices and the inexistence of speech/voice overlaps.

The ProGmatica corpus has also been used for system evaluation. For this another 20 minutes were chosen using the same criteria.

2.2. HMM Framework

This system is based on a HMM framework with 50 models (20 consonants, 18 vowels, 8 diphthongs, 1 oclusion for voiced plosives, 1 occlusion for unvoiced plosives, 1 pause, 1 aspiration) with the same basic left to right topology with jumps (by quotient 9.7% better than the no-jumps topology) and 3 states in most cases. The first 38 models constitute the phonetic base for standard EP.

The errors found in vowel separation in diphthongs were significantly greater than those encountered in other situations. The soft formant frequencies transition make it hard, to humans and machines, to accurately define the inner frontier. These errors affect the overall performance for each independent phoneme occurrence and decrease significantly the global system quality because vowels represent about 50% of phoneme occurrences in EP. As EP has a large variety and high frequency of diphthongs, when compared to other languages (English for example), a study has been made in order to find the performance difference that would occur if these phonetic units were considered independently or as a whole. Adding 8 extra HMM with 5 states (9.3% improvement relative to 3 states), representing the EP main diphthongs, resulted on an increase of 3.4% in boundaries localization comparing to the independent phoneme (vowels) results and no benefits were noticed in classification.

For plosives, which represent nearly 20% of the average phonetic occurrences in EP, distinct models were created for voiced and unvoiced cases. This can give an increase of 1.1% in segmentation results and 0.3% in classification.

For non phonetic events two extra models are used to represent aspiration and pause.

2.3. Trainning

Since this approach is based on models that describe a speech signal in time, it is possible to simultaneously perform both segmentation and labeling at the same time. The pursue for an optimal configuration for the annotation system showed, by the analysis of the results, that the two implied tasks demand incompatible parameters. The variation of a parameter can improve performance on one hand (better classification) and decrease it on another hand (worst segmentation). Thus, two function oriented sets of configuration parameters were developed, one for segmentation and another for labeling.

Varying the analysis window length and the overlap time parameters several results can be obtained. Table 1 resumes this analysis showing the correctness for recognition and error rate for segmentation for a system with 12MFCC. Insertions, deletions and substitution errors (these last only considered for classifying) are included.

Overlap	Window length (ms)		
	15	20	25
2.5	**76.5** / 65.2	76.3 / 63.2	75.3 / 61.0
5.0	75.0 / **69.7**	74.7 / 67.2	73.8 / 65.7
7.5	72.3 / 64.9	72.5 / 64.5	70.0 / 63.9
10.0	68.4 / 62.9	69.3 / 62.7	68.6 / 61.1

Table 1. Recognition/Segmentation results in percentage for several windows and overlap times.

States	Class. Acc. (%)	Segment. Acc. (%)
3	0,00% baseline	0,00% baseline
4	7,01%	-9,03%
5	2,75%	-7,77%
6	0,00%	-12,46%
7	0,00%	-12,43%

Table 2: Results varying the model's number of states

Mix.	Class. Acc. (%)	Segment. Acc. (%)
3	0,00% baseline	0,00% baseline
4	15,18%	4,96%
5	19,41%	7,67%
6	22,92%	12,89%
7	23,85%	11,15%

Table 3: Results varying the state's number of mixtures

Feature	Class. Acc. (%)	Segment. Acc. (%)
MFCC	0,00% baseline	0,00% baseline
MFCC_E	31,39%	18,28%
MFCC_0	28,93%	13,79%
MFCC_ED	57,55%	39,05%
MFCC_DA	64,98%	45,61%
MFCC_EDA	70,88%	50,81%

Table 4: Results varying the feature vectors (E for Energy, D for Delta, A for Acceleration)

Tables 2, 3 and 4 show the results for others parameters variations in relation to a very simple set of initial parameters that served as a baseline (25ms window, 10ms overlap, 12MFCCs).

Finally, considering all the obtained results, two sets of parameters were defined. For segmentation, 15 ms frames with 5 ms overlap have been used. The feature vector was composed by 16 MFCCs plus Energy, delta and acceleration coefficients, for each state 7 Gaussian mixtures. For classifying, 25 ms frames with 7.5 ms overlap, 14 MFCCs plus Energy, delta coefficients and 5 Gaussian components per state have been used.

To reduce the complexity in decoding the phonetic sequence a language model was also developed. This first language model, also used in the previously described analysis, was based on simple rules that benefict from the characteristic consonant-vowel (CV) alternance of EP. The sequences CCV and VV are also common. This model was then extended to a bigram in order to include exceptions. With this, the results had an accuracy of 78.67% for classification, 77.97% for a 10 ms tolerance zone and 84.33% for a 20ms tolerance zone.

In the development of these systems, when the final accuracy values were similar for two different parameter sets, the one with more insertion errors was preferred. This kind of error is easier to be corrected. So, with this in mind, the above accuracy values have almost no deletion errors.

2.3. Training

The set of HMM was first initialized using the FEUP/IPB corpus in order to obtain a solid phoneme description. After, the speaker dependent highly constrained HMM set need to be re-trained and adapted to other voices and to non ideal studio conditions. This way, the ProGmatica corpus was used.

2.4. Refinement

In the last stage, the annotation tool analyses the results so far and tries to improve boundaries' positions and classification errors.

The given boundaries are analyzed using cues given by acoustical features extracted from the speech signal. For the selection of the most interesting feature that could give the best representation of the acoustical information, a study has been made giving a special emphasis on autoregressive moving average models (ARMA). These models are based on the generic equation 1, representing the denominator an AR model (with $b_l=0$) and the numerator a MA model ($a_k=0$).

$$H(z) = G\frac{1+\sum_{l=1}^{q} b_l z^{-1}}{1+\sum_{k=1}^{p} a_k z^{-1}} \qquad (1)$$

The number of AR and MA coefficients considered to effectively describe the speech signal has been independently analyzed. The results in table 5 have been obtained using 16KHz speech signals, 20ms windows with Hamming mask and 10ms overlapping.

Phon.	MA	AR	Avg. Error	Max. Error
m, n, J	16	4	0.53%	3.07%
	4	16	1.51%	12.13%
	16	16	0.19%	18.60%
a, p, s	16	4	2,37%	16.54%
	4	16	1,02%	9.72%
	16	16	0.18%	6.85%

Table 5. Prediction Error by Model Type

The all-pole models, commonly used, can give a very good representation of most phonemes but, when applied to nasals and some unvoiced sounds, the performance decreases a lot. The use of some MA coefficients increases the results and comes to solve this problem. The benefice of adding new poles or zeros can only be observed when the description capacity of the model is not reached. After this, some slight decrease can be noticed.

Using the described models, speech feature vectors have been composed using 4 zeros from de MA model, 16 poles from the AR model, signal energy and the dynamic related values delta and acceleration. The signal is analyzed from 20ms before the given boundary to 20ms after using 10ms segments with a Hamming window and a 2.5ms overlap. The Euclidean distance is calculated for every two successive vectors and the likelihood is observed. The phonetic boundary confidence is based on a threshold level which is calculated locally. The standard deviation is also used to create a threshold band. Instead of having only a simple frontier line that limits two phonemes in a multi-dimensional space, two limits are used, one for a raising distance and another for a decreasing distance (hysteresis classifier). This behavior has been implemented in order to reduce the number of false identified boundaries.

The final phonetic sequence is yet reviewed. The too big and too small occurrences (very rarely) are deleted and the duration of each phoneme is tested against pre-known EP knowledge. In this case the error in boundaries is calculated for each phone and compared with its reference duration. An error of 15ms in a /d/, which has an average duration of 15.7ms, would be much worst than the same error in an /a/, which has an average duration of 112.3ms. Table 6 shows the average errors and the average durations.

The log probability associated with each phoneme is also used in order to remove some errors. A problem comes up in the deletions of phonemes. A phone has two boundaries and, assuming that the phonetic sequence is composed by contiguous segments, when a phone is deleted, one of the frontiers must be moved. This can be made by readjusting the frontier of the

previous phone or by repositioning the beginning of the next phone. The use of rigid rules for the two operations has been examined. The assimilation of the phone by the previous phone was more interesting but the differences between the two did not dictate a rule that could be generalized.

Symb.	Avg. Dur. (ms)	Avg. Err. (%)	Symb.	Avg. Dur. (ms)	Avg. Err. (%)
a	112,3	0,96%	ocl	51,6	0,29%
asp	292,0	1,01%	O	94,3	2,77%
@	51,8	1,35%	p	24,3	7,96%
d	15,7	2,32%	r	35,8	0,45%
E	77,2	1,66%	R	79,8	3,40%
E	83,5	1,02%	s	99,7	3,30%
i	63,0	0,65%	S	83,1	0,45%
i_	85,3	3,96%	6	63,4	1,62%
j	32,8	5,12%	t	31,5	0,95%
k	35,5	5,49%	u	48,6	0,21%
l	49,1	1,19%	u_	77,9	2,86%
L	81,7	2,71%	v	57,4	0,70%
m	58,9	0,15%	w	25,5	1,59%
o	82,8	2,13%	Z	82,7	0,27%

Table 6. Average duration for reference phonemes and average error in segmentation.

For improving this task an algorithm has been created. For deleting phones, the duration of the two neighbor phones are analyzed. The phone to change will be the one that, after deletion, can give a set of durations closer to those in the knowledge database. The described operations can be seen on figure 2 in which are represented the two deletions' possibilities.

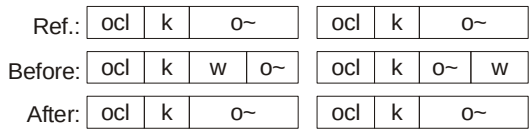


Figure 2. Phone deletion at left and right.

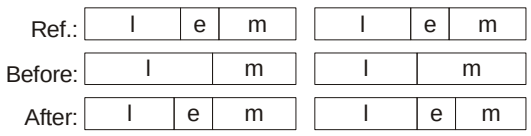


Figure 3. Phone insertion at left and right.

For insertions of phones in the phonetic sequence (corrections of deletion errors) a similar problem occurs. In this way, similar rules are used. The new label that will be inserted is obtained by analyzing context and a grammar. The durations of the phones that will be separated by the new phone are analyzed and compared with the average occurrences in the database. The frontier which is changed is the one that can give a better duration similarity with the know phones' durations. The duration of the new phone is based on the average duration of the same phones in the database and it is adjusted by a factor resulting from the comparison of the durations of the neighbor phones. This operation is shown in figure 3.

3. RESULTS AND CONCLUSION

With the described process, the obtained results were, for segmentation accuracy, 65.3% for a 10ms interval, 79.2% for a 15ms interval and 88.3% for a 20ms interval as can be seen in figure 4.

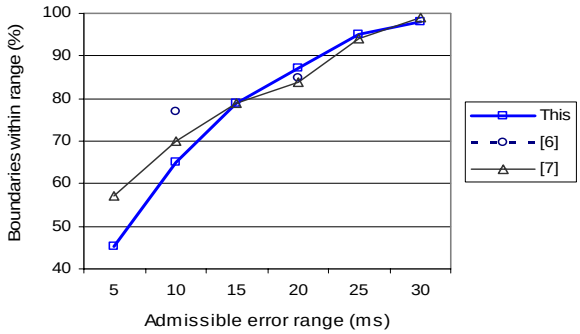


Figure 4. Final results and comparison

In this paper, an automatic annotation system has been presented. Regarding the use of natural speech recorded in real situations and trying to ease the quick creation of speech databases the system take as entry information only the speech signal. Some refinement procedures based on signal analysis can give an extra refinement to the boundaries localization. These results are close to or better than those reported by other authors that also worked with EP but not with spontaneous speech [6][7].

4. REFERENCES

[1] A. Sethy, S. Narayanan, "Refined Speech Segmentation for Concatenative Speech Synthesis", Proc.of ICSLP, Denver, 2002.

[2] A. Ljolje, M. Riley, "Automatic Segmentation and Labelling of Speech", Proc. of ICASSP, p473-476, 1991.

[3] J. Hosom, "Automatic Time Alignment of Phonemes using Acoustic-Phonetic Information", PhD Thesis, OGI, 2000

[4] J. Teixeira, D. Freitas, D. Braga, M. J. Barros, V. Latsch, "Phonetic Events from the Labeling of the European Portuguese Database for Speech Synthesis, FEUP/IPB-DB", Proc. of EuroSpeech'01, Denmark, 2001

[5] D. Braga, J. P. Teixeira, L. Coelho, D. Freitas, "ProGmatica: Uma base de dados prosódica pragmaticamente orientada em Português Europeu", Actas do XXI Encontro da Associação Portuguesa de Linguística, Lisboa, 2006

[6] P. Carvalho, "Determinacão Automática de Segmentos para Síntese de Fala por Concatenação", PhD thesis, Universidade Técnica de Lisboa - IST, 2004.

[7] L. Coelho, "Etiquetagem Automática de Sinais de Fala: Classificação e Anotação", Tese de mestrado, FEUP, 2005

DISEÑO E IMPLEMENTACIÓN DE UNA INTERFAZ GENÉRICA PARA LA ESTIMACIÓN DE AUTÓMATAS DE ESTADOS FINITOS PONDERADOS

Mikel Penagarikano, German Bordel, Luis Javier Rodriguez, Maider Zamalloa

Grupo de Trabajo en Tecnologías del Software.
Departamento de Electricidad y Electrónica.
Facultad de Ciencia y Tecnología. Universidad del País Vasco.
Barrio Sarriena s/n. 48940 Leioa. Bizkaia.
mikel.penagarikano@ehu.es
http://gtts.ehu.es

RESUMEN

En este artículo se define y desarrolla una aproximación unificada al problema de estimar autómatas de estados finitos ponderados (AEFP), mediante una interfaz genérica válida para todo tipo de autómatas, cualquiera que sea su estructura interna. La metodología propuesta se ha integrado en *Sautrela*, un paquete de software desarrollado como código abierto en Java™, altamente modular y escalable, orientado al procesamiento de señal en general aunque muy enfocado al reconocimiento del habla (véase http://sautrela.org). Una vez definida la aplicación que enlaza los estados y transiciones de un autómata con los parámetros de la interfaz, ésta permite no sólo procesar secuencias de datos y obtener la decodificación óptima, sino también estimar los parámetros del autómata de acuerdo al criterio de optimización que se establezca. Esta aproximación permite reducir drásticamente el esfuerzo necesario para añadir nuevos módulos a un sistema de procesamiento del habla basado en AEFP.

1. INTRODUCCIÓN

Desde el inicio de las investigaciones sobre reconocimiento automático del habla hasta hoy se han desarrollado muchos sistemas, con muy diferentes funcionalidades, determinadas sobre todo por la tecnología disponible [1, 2, 3]. Inicialmente esos sistemas estaban orientados a la investigación de algunos de los elementos que conforman el proceso de reconocimiento y su diseño estaba optimizado para la consecución de dicho objetivo. Las implementaciones apenas permitían la migración a nuevas tareas o la integración de nuevas tecnologías. Por lo común, cualquier innovación implicaba el desarrollo de un sistema completo desde cero. En los últimos años la flexibilidad y la escalabilidad se han impuesto como principales criterios de diseño, de modo que los motores de reconocimiento han pasado a ser plataformas modulares que por un lado incorporan el estado del arte de las tecnologías del habla, y por otro, permiten el desarrollo y la integración de nuevas áreas de investigación. Sin embargo, los motores de última generación están pensados básicamente como máquinas decodificadoras, por lo que la estimación de modelos acústicos, léxicos o de lenguaje debe realizarse mediante paquetes de software externos.

Sautrela [4] es un paquete de software de código abierto ideado para facilitar el desarrollo de aplicaciones de procesamiento de señal, especialmente de procesamiento del habla. Sautrela es altamente modular y escalable, por lo que se le pueden añadir módulos o *plugins* de manera sencilla. Se ha desarrollado mediante tecnología Java™, lo que asegura su portabilidad a una extensa variedad de plataformas. A diferencia de los sistemas mencionados más arriba, Sautrela define un único formalismo para decodificar señales y estimar modelos, es decir, no es simplemente un motor de reconocimiento alimentado con modelos externos, sino que incorpora los algoritmos y criterios de estimación necesarios para producirlos internamente.

Este trabajo se centra en el problema de la estimación de modelos que, independientemente de su estructura interna, puedan representarse como Autómatas de Estados Finitos Ponderados (AEFP), a través de una simple interfaz. Esta aproximación permite reducir drásticamente el esfuerzo necesario para añadir nuevos módulos a un sistema de procesamiento del habla basado en AEFP.

El resto del artículo se estructura como sigue. En el apartado 2 se presenta la jerarquía de interfaces necesaria para representar y manejar AEFP. En el apartado 3 se introducen las interfaces que generalizan la estimación de parámetros para cualquier modelo representado mediante AEFP. En el apartado 4 se revisan los dos criterios de estimación más habituales y se describen sus implementaciones. En el apartado 5 se muestran ejemplos de implementación de las interfaces de estimación para tres tipos diferentes de modelos. Por último, en el apartado 6 se resumen las principales aportaciones de este trabajo.

Este trabajo ha sido parcialmente financiado por el Gobierno Vasco, dentro del programa SAIOTEK, a través de los proyectos S-PE04UN26 y S-PE05UN32.

(a) Interfaz básica AEFP

```
// Devuelve el nombre del AEFP
String getName();

// Devuelve el símbolo de nombre name
Symbol getSymbolByName(String name);

// Devuelve el estado inicial del AEFP
State getIniState();

// Devuelve la probabilidad de que el estado
// state sea final
double getFinProb(State state);

// Devuelve todas las transiciones posibles
// desde el estado state
Transition[] getTrans(State state);
```

(b) Interfaz específica AEFP-d (extensión de AEFP)

```
// Devuelve la transición correspondiente al
// par (state, sy)
Transition getTrans(State state, Symbol sy);
```

(c) Interfaz específica AEFP-nd (extensión de AEFP)

```
// Devuelve el conjunto de transiciones
// correspondiente al par (state, sy)
Transition[] getTrans(State from, Symbol sy);
```

Figura 1. La interfaz básica AEFP (a) consta del conjunto mínimo de métodos que permiten manejar un autómata. Cada una de las sub-interfaces (b y c) añade un único método, de acuerdo al carácter determinista o no determinista de la función de transición.

2. INTERFACES BÁSICAS PARA LA MANIPULACIÓN DE AEFP

2.1. Jerarquía de interfaces

Sautrela utiliza una jerarquía de interfaces muy simple, que permite unificar los diferentes tipos de modelos estocásticos que se aplican habitualmente en reconocimiento automático del habla. En la parte superior de dicha jerarquía se encuentra la interfaz básica de acceso a AEFP (véase la Figura 1), más dos interfaces diferenciadas para representar AEFP deterministas (AEFP-d) y AEFP no deterministas (AEFP-nd), que constan de un único método que permite manejar las transiciones en ambos tipos de formalismos. Básicamente, un AEFP consta de un estado inicial, un alfabeto de entrada y una función de transición probabilística que asigna al par formado por un estado y un símbolo de entrada otro estado (AEFP-d) o conjunto de estados (AEFP-nd). De hecho, *Estado*, *Símbolo* y *Transición* son interfaces flexibles del sistema que permiten manejar varios tipos de objetos, como por ejemplo cadenas de caracteres (para los modelos de lenguaje) o vectores de componentes reales (para los modelos acústicos).

Sautrela es capaz de manejar cualquier modelo que pueda ser representado mediante estas interfaces, independientemente de su estructura interna. No obstante, los sistemas de reconocimiento constan de varios modelos distintos, uno por cada fuente de conocimiento, que es necesario integrar en un único formalismo. Precisamente, para dar respuesta a esta necesidad, en un trabajo previo se han propuesto los *Modelos de Markov por Capas* (MMC) [5].

2.2. Integración de distintas fuentes de conocimiento mediante MMC

Un MMC consta de varias capas, cada una correspondiente a una fuente de conocimiento. Cada capa, formada por un número finito de AEFP, modela sus unidades en términos de unidades pertenecientes a capas inferiores. Independientemente del número de capas y de los tipos de AEFP que sea necesario utilizar, un MMC es equivalente a un AEFP-nd.

No hay limitaciones en cuanto al número de capas, modelos, estados o símbolos que conforman un MMC, por lo que puede utilizarse para integrar diferentes fuentes de conocimiento en un único autómata. Tal es precisamente el problema planteado por los sistemas de reconocimiento del habla, que utilizan Modelos Ocultos de Markov (MOM), diccionarios de pronunciaciones o, en su defecto, modelos léxicos en forma de grafo, gramáticas estocásticas (n-gramas), etc. Todos estos modelos pueden ser representados como AEFP a través de las interfaces mostradas en la Figura 1, de modo que el acoplamiento *ad-hoc* presente todavía en muchos sistemas de reconocimiento, que dificulta o impide completamente la incorporación de mejoras tecnológicas, puede formalizarse de manera mucho más sencilla, flexible y eficiente mediante un MMC. Otros autores han presentado aproximaciones similares (véanse, por ejemplo, [1] y [6]), pero están diseñadas para la decodificación de señales y carecen de herramientas genéricas para la estimación de modelos.

3. LA INTERFAZ DE ENTRENAMIENTO

La interfaz AEFP, junto con las sub-interfaces AEFP-d y AEFP-nd, conforman el conjunto mínimo de métodos necesario para afrontar el problema de la decodificación. Los procedimientos de entrenamiento empleados habitualmente en los sistemas de reconocimiento comienzan con una primera fase de decodificación en la que se calcula una función objetivo; tras ella sigue una segunda fase en la que los parámetros de los modelos son reestimados con objeto de optimizar dicha función. De igual modo, los parámetros de un AEFP (probabilidades de transición y probabilidades finales) pueden reestimarse con objeto de optimizar una función objetivo. No obstante, la información de dichos parámetros debe ser transferida a la representación interna del AEFP. De esta forma, es posible reestimar externamente los parámetros de un AEFP, mientras que el propio autómata será el responsable de la conversión de los parámetros ya reestimados a su representación interna.

```
// Inicializa las cuentas de entrenamiento
void initTrCounts();

// Incrementa las cuentas asociadas a una
// transición
void incTrTransCount(Transition tr, double c);

// Incrementa las cuentas asociadas a un
// estado final.
void incTrFinalCount(State state, double c);

// Vuelca las cuentas a un modelo,
// actualizando sus parámetros
void dumpTrCounts();
```

Figura 2. La interfaz de entrenamiento permite calcular las cuentas externamente, dejando en manos del modelo transferir a su representación interna la información contenida en dichas cuentas.

Para facilitar la transferencia de los parámetros reestimados, éstos son transmitidos en forma de cuentas incrementales. Como se verá más adelante, lo que llamamos *cuenta* no es más que la probabilidad de una transición en un instante t, dada la secuencia de entrada y los parámetros actuales del autómata. El conjunto de métodos que conforman la interfaz de entrenamiento permite inicializar, transmitir y volcar las cuentas para actualizar los parámetros de un AEFP (véase la Figura 2).

4. CRITERIOS DE ESTIMACIÓN

Haciendo uso de la presente metodología, pueden aplicarse diversos criterios de estimación (y sus correspondientes transformaciones). En los siguientes dos párrafos se revisan dos de los criterios más utilizados: Máxima Verosimilitud y Máxima Información Mutua.

4.1. Máxima Verosimilitud

La Máxima Verosimilitud (MV) es el criterio de estimación más extendido. Sea $\mathbf{X}$ una variable aleatoria con función de distribución $p_\phi(x)$, donde ϕ pertenece a un espacio de parámetros Φ. Dada una secuencia de observaciones independientes $\mathbf{x} = \{x_1, ..., x_n\}$, la función de verosimilitud viene dada por:

$$L = p_\phi(\mathbf{x}) = \prod_{i=1}^{n} p_\phi(x_i)$$

La estimación MV de ϕ se obtiene maximizando L:

$$\phi_{\mathrm{ML}} = \arg\max_{\phi} p_\phi(\mathbf{x})$$

El teorema de Baum-Sell para transformaciones crecientes [7] es de aplicación en dicha maximización, pudiendo obtenerse una fórmula de reestimación para todos los parámetros. En el caso de un AEFP, los parámetros se restringen a las probabilidades de transición y finales, las cuales deben sumar 1 para todo estado origen. Sea $p_\phi(\tau)$ la probabilidad de la transición $\tau = (s, d, y)$, con estado origen s, estado destino d y símbolo de emisión y. La probabilidad de ser final de un estado puede formalizarse como una transición externa especial con emisión nula, $p_\phi(s, out, null)$, y por tanto, puede ser integrada dentro de la función de transición. Del teorema de Baum-Sell se deduce que la reestimación de $p_\phi(\tau)$ viene dada por:

$$p_{\tilde{\phi}}(\tau) = \frac{p_\phi(\tau) \cdot \left(\frac{\partial L}{\partial P(\tau)}\right)_\phi}{\sum_{\tau'=(s,d',y')} p_\phi(\tau') \cdot \left(\frac{\partial L}{\partial P(\tau)}\right)_\phi} \qquad (1)$$

Esta expresión puede ser reescrita en términos de cuentas incrementales calculadas para todas las muestras de entrenamiento. Sea $count(\tau)_{t,i}$ la probabilidad de la transición τ en el instante t y dada la observación x_i. Estas cuentas pueden ser eficientemente calculadas haciendo uso de los denominados *coeficientes forward* y *backward* (véase [8]). Finalmente, la expresión (1) puede reescribirse como sigue:

$$p_{\tilde{\phi}}(\tau) = \frac{\sum_{i=1}^{n} \sum_{t=1}^{m_i} count(\tau)_{t,i}}{\sum_{\tau'=(s,d',y')} \sum_{i=1}^{n} \sum_{t=1}^{m_i} count(\tau')_{t,i}} \qquad (2)$$

4.2. Máxima Información Mutua

Cualquier problema de reconocimiento de patrones puede ser formalizado haciendo uso de la metáfora del *canal ruidoso*. Dado un par de variables Ω y X, la Información Mutua $I(X;\Omega)$ da cuenta de la reducción de la incertidumbre de X ante el conocimiento de Ω y viceversa. Si suponemos que el vector de muestras $(\mathbf{x}, \omega) = \{(x_1, \omega_1), ..., (x_n, \omega_n)\}$ es suficientemente representativo, la Información Mutua puede aproximarse mediante la siguiente expresión:

$$I(X;\Omega) \approx \log \frac{\prod_{i=1}^{n} p_\phi(x_i \mid \omega_i)}{\prod_{i=1}^{n} p_\phi(x_i)} = \log \frac{\prod_{i=1}^{n} p_\phi^{sup}(x_i)}{\prod_{i=1}^{n} p_\phi^{nosup}(x_i)}$$

Nótese que el numerador $p_\phi^{sup}(x_i)$ hace referencia a la probabilidad de la observación x_i cuando la clase ω_i es conocida (existe una supervisión). Por otro lado, el denominador $p_\phi^{nosup}(x_i)$ da cuenta de la probabilidad de x_i en ausencia de supervisión. Dado un conjunto de clases, cada una representada por un AEFP, ambas probabilidades pueden ser calculadas integrando el conjunto de autómatas en un MMC que contará con una capa adicional que dará cuenta de la posibilidad o no de supervisión. La estimación por Máxima Información Mutua (MIM) de ϕ se obtiene de la maximización de $I(X;\Omega)$:

$$\phi_{\mathrm{MIM}} = \arg\max_{\phi} \frac{\prod_{i=1}^{n} p_\phi^{sup}(x_i)}{\prod_{i=1}^{n} p_\phi^{nosup}(x_i)}$$

El teorema de Gopalakrishnan para funciones racionales [9] demuestra que existe una constante C, tal

que la siguiente expresión es una transformación creciente:

$$p_{\tilde{\phi}}(\tau) = \frac{p_\phi(\tau) \cdot \left(\left(\frac{\partial I}{\partial P(\tau)} \right)_\phi + C \right)}{\sum_{\tau'=(s,d',y')} p_\phi(\tau') \cdot \left(\left(\frac{\partial I}{\partial P(\tau)} \right)_\phi + C \right)} \quad (3)$$

Del mismo modo que en el caso de la estimación por MV, las cuentas pueden calcularse de manera eficiente haciendo uso de los coeficientes *forward* y *backward*. En este caso, definiremos dos cuentas parciales, $count(\tau)_{t,i}^{sup}$ y $count(\tau)_{t,i}^{nosup}$, y la cuenta resultante será la diferencia de las mismas:

$$count(\tau)_{t,i} = count(\tau)_{t,i}^{sup} - count(\tau)_{t,i}^{nosup}$$

La constante C mencionada anteriormente garantiza la convergencia incluso en presencia de cuentas negativas[1]. Finalmente, la ecuación (3) puede escribirse como sigue:

$$p_{\tilde{\phi}}(\tau) = \frac{C \cdot p_\phi(\tau) + \sum_{i=1}^{n} \sum_{t=1}^{m_i} count(\tau)_{t,i}}{\sum_{\tau'=(s,d',y')} C \cdot p_\phi(\tau') + \sum_{i=1}^{n} \sum_{t=1}^{m_i} count(\tau')_{t,i}} \quad (4)$$

5. EJEMPLOS DE IMPLEMENTACIÓN DE LAS INTERFACES

5.1. Modelos de Markov por Capas

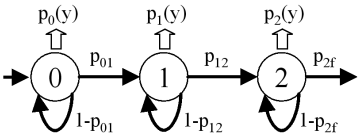
El meollo de la presente metodología de entrenamiento mediante interfaces reside en la posibilidad de reestimar un MMC compuesto por diversas capas organizadas jerárquicamente. La reestimación de un MMC implica el entrenamiento simultáneo de todos los modelos que lo componen, es decir, el entrenamiento de un conjunto de modelos. Como se ha comentado en la Sección 2, un MMC puede ser visto como un AEFP-nd. Por tanto, dado un criterio de entrenamiento, sus parámetros (desde el punto de vista de la interfaz) pueden ser reestimados. Para ello, basta calcular las cuentas para todas las muestras de entrenamiento y transmitirlas mediante la interfaz de entrenamiento mostrada en la Figura 2. Sin embargo, surge la pregunta de cómo transmitir dicha información a los modelos internos que conforman un MMC.

La respuesta es bastante simple. Una transición τ de un MMC consiste en una secuencia de transiciones $\Delta_\tau = \{\delta_1^\tau, ..., \delta_k^\tau\}$ dentro de las capas internas. Es por ello que las cuentas de dicha transición, $count(\tau)_{t,i}$, deben ser transmitidas a todas las transiciones internas que componen τ. En otras palabras, la cuenta $count(\delta)_{t,i}$ de una transición interna δ, resulta de la suma de todas las transiciones τ del MMC tales que $\delta \in \Delta_\tau$. Sea $T(\delta) = \{\tau \mid \delta \in \Delta_\tau\}$. Entonces,

$$count(\delta)_{t,i} = \sum_{\tau \in T(\delta)} count(\tau)_{t,i}$$

[1] Aunque la expresión (3) proporciona una transformación creciente para valores de C suficientemente grandes, los experimentos muestran que valores pequeños de C ofrecen una convergencia más rápida.

(a) Un MOM-d de 3 estados



(b) El AEFP equivalente

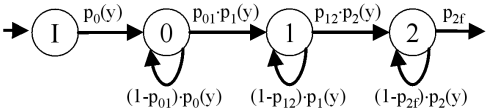


Figura 3. Cada transición en el MOM genera en el AEFP equivalente tantas transiciones como número de emisiones en el estado destino del MOM. El estado inicial adicional del modelo equivalente contiene transiciones a todos los estados iniciales del MOM.

5.2. Modelos Ocultos de Markov Discretos

En los MOM discretos las emisiones están ligadas a los estados, mientras que en los AEFP aquí descritos las emisiones ocurren en las transiciones. No obstante, el AEFP equivalente a un MOM discreto puede obtenerse fácilmente suponiendo que las emisiones ocurren en las transiciones, justo antes de llegar al estado (véase la Figura 3).

Para implementar la interfaz de entrenamiento, cada cuenta del AEFP debe ser transferida a la representación interna de los MOM discretos (cuenta inicial, de transición y de emisión para cada uno de los estados). Sea $T_A(s,d) = \bigcup_{\forall y} \{\tau = (s,d,y)\}$ el conjunto de transiciones del AEFP con estado origen s y estado destino d, $T_B(d,y) = \bigcup_{\forall s} \{\tau = (s,d,y)\}$ el conjunto de transiciones del AEFP con estado destino d y símbolo de emisión y, y s_I el estado inicial del AEFP. Entonces, las cuentas internas del MOM discreto vienen dadas por:

$$count_{init}(s)_{t,i} = \sum_{\tau \in T_A(s_I,s)} count(\tau)_{t,i}$$

$$count_{trans}(s,d)_{t,i} = \sum_{\tau \in T_A(s,d)} count(\tau)_{t,i}$$

$$count_{emit}(s,y)_{t,i} = \sum_{\tau \in T_B(s,y)} count(\tau)_{t,i}$$

Es decir, cada cuenta del AEFP es transferida bien a la cuenta inicial, bien a la cuenta de transición del MOM (dependiendo del estado origen), y siempre a la cuenta de emisión del estado destino. Cabe destacar que para el

caso del entrenamiento por MV, las cuentas así obtenidas son exactamente las mismas que se obtendrían mediante el algoritmo de Baum-Welch [8].

5.3. Modelos Ocultos de Markov Continuos

El AEFP equivalente a un MOM continuo es análogo al presentado previamente, a excepción de que existirá un conjunto continuo (es decir, infinito) de transiciones. Esto no supone problema alguno para el formalismo de los AEFP aquí descrito, ya que no impone condición alguna sobre la naturaleza continua o no de los simbolos emitidos, y por tanto, puede manejar conjuntos teóricamente infinitos de transiciones.

6. CONCLUSIONES

En este artículo se ha presentado una metodología novedosa para el entrenamiento de un AEFP. Todo modelo que implemente una sencilla interfaz puede ser entrenado independientemente de su estructura y sin conocimiento alguno de su representación interior. Para el caso del entrenamiento de MOM por MV, la reestimación es exactamente la misma que se obtendría mediante el algoritmo de Baum-Welch. Por otra parte, los MMC constituyen un formalismo extremadamente útil tanto para la decodificación como para el entrenamiento, ya que permiten la reestimación simultánea de un conjunto de AEFP. Partiendo de esta metodología, se ha desarrollado un módulo de entrenamiento para el sistema *Sautrela*, que simplifica considerablemente el esfuerzo necesario para incorporar nuevos formalismos de modelado a un sistema de procesamiento del habla basado en AEFP.

7. BIBLIOGRAFÍA

[1] J.K. Baker, "The DRAGON System - An Overview," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 23, no. 1, pp. 24–29, 1975.

[2] S. Young, "The HTK Hidden Markov Model Toolkit: Design and Philosophy," Tech. Rep. TR.153, Department of Engineering, Cambridge University, 1993.

[3] K. Lee, H. Hon, y R Reddy, "An Overview of the SPHINX Speech Recognition System," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 1, pp. 35 – 45, January 1990.

[4] M. Penagarikano y G. Bordel, "Sautrela: A Highly Modular Open Source Speech Recognition Framework," in *Proceedings of the ASRU Workshop*, 2005.

[5] M. Penagarikano y G. Bordel, "Layered Markov Models: A New Architectural Approach to Automatic Speech Recognition," in *Proceedings of the MLSP Workshop*, 2004.

[6] M. Mohri, F. Pereira, y M. Riley, "Weighted Finite-State Transducers in Speech Recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.

[7] Leonard. E. Baum y George R. Sell, "Growth Transformations for Functions on Manifolds," *Pacific Journal of Mathematics*, vol. 27, pp. 211–227, 1968.

[8] Leonard. E. Baum, "An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes," *Inequalities*, vol. 3, pp. 1–8, 1972.

[9] P. S. Gopalakrishnan, D. Kanevsky, A. Nádas, y D. Nahamoo, "An Inequality for Rational Functions with Applications to Some Statistical Estimation Problems," *IEEE Trans. Information Theory*, vol. 37, pp. 107–113, 1991.

RECONOCIMIENTO AUTOMÁTICO DEL HABLA EN EUSKERA EMPLEANDO INFORMACIÓN LINGÜÍSTICA

Víctor G. Guijarrubia, M. Inés Torres

Departamento de Electricidad y Electrónica
Universidad del País Vasco, Apartado 644, 48080 Bilbao. Spain
{vgga,manes}@we.lc.ehu.es

RESUMEN

El presente trabajo presenta experimentos preliminares en reconocimiento automático del habla en euskera haciendo uso de información lingüística, más concretamente, de lemas. El uso de esta información permite reducir el vocabulario de la tarea y agilizar el proceso de reconocimiento, manteniendo las prestaciones de un sistema estándar basado en palabras.

1. INTRODUCCIÓN

En el presente trabajo pretendemos adentrarnos en el reconocimiento automático del habla en euskera empleando un cierto conocimiento lingüístico.

El euskera es un lenguaje minoritario, pero oficial en el País Vasco. Se trata de un idioma pre-indoeuropeo de origen desconocido. Comunmente se ha asociado con el alemán, debido a que, como este, presenta una alta declinación, tanto en nombres como en verbos. Por ejemplo, *a casa* se traduce como *etxeRA*, *a las casas*, como *etxeETARA* y *a casas*, como *etxeTARA*. Es por esto que siempre se ha pensado que el uso de conocimiento lingüístico, como el proporcionado por los lemas, podría ser de gran ayuda y se hayan producido grandes estudios en el desarrollo de analizadores morfosintácticos [1] y lematizadores [2]. El empleo de este tipo de información ha reportado mejoras en otras aplicaciones relacionadas con las tecnologías del habla, como por ejemplo en sistemas de traducción estadística texto-texto de castellano a euskera [3].

En este trabajo, se muestran experimentos de reconocimiento automático del habla que hacen uso del conocimiento proporcionado por los lemas disponibles para las palabras de una base de datos meteorológica.

En lo que sigue, el resto del artículo está organizado de la siguiente manera: en la sección 2 se describen los métodos empleados para hacer uso de ese conocimiento lingüístico; en la sección 3 se describe la base de datos empleada en la experimentación, tanto la parte textual empleada para entrenar los modelos de lenguaje como la parte de voz empleada en experimentos de reconocimiento automático del habla; en la sección 4 se describe la experimentación llevada a cabo y los resultados obtenidos; en la sección 5 se encuentran los agradecimientos y por último, en la sección 6 se dan las conclusiones y las líneas de actuación futuras para completar este trabajo.

Este trabajo ha sido subvencionado por la Universidad del País Vasco bajo el proyecto 9/UPV 002224.310-15900/2004 y el CICYT bajo el proyecto TIN2005-08660-C04-03

2. HACIENDO USO DE LOS LEMAS

Un lema es cada una de las formas que, en ciertas lenguas, presenta un radical para recibir los morfemas de flexión. Para hacer uso de la información suministrada por los lemas, se han seguido dos líneas de actuación: usarla diréctamente para agrupar palabras bajo un lema común y emplearla para obtener una serie de raices y sufijos típicos en la tarea de aplicación.

Por un lado, se usaron los lemas diréctamente, haciendo un sistema de reconocimiento donde las unidades léxicas no eran las palabras sino los lemas. En este caso, las palabras se introdujeron como pronunciaciones alternativas del lema dentro del léxico. En la experimentación descrita en el presente trabajo, a cada pronunciación se le asignó la probabilidad del unigrama, es decir, el número de veces que aparece dicha palabra lematizada como un cierto lema dividido por el número total de apariciones de dicho lema.

Por otro lado, se pretendió obtener los sufijos más comunes que aparecen en la lengua vasca, restringidos a la tarea de la aplicación, para emplearlos como palabras independientes en un sistema de reconocimiento automático del habla.

Al estar la base de datos disponible tanto en forma de palabras como en forma de lemas, fue posible agrupar todas las palabras correspondientes a un mismo lema. Una vez obtenidas las agrupaciones por lema, la separación en cada agrupación se realizó tomando la raíz común. De este modo, de una sola palabra se pasó a dos: raíz y sufijo diferenciador. De este proceso se obtuvieron, por ejemplo, casos como los mostrados en las figura 1 y 2, donde se muestran los casos de las palabras agrupadas bajo los lemas *agertu* (aparecer) y *aldatu* (cambiar) y se puede observar que de ocho palabras se pasa a siete, dos raices y cinco sufijos.

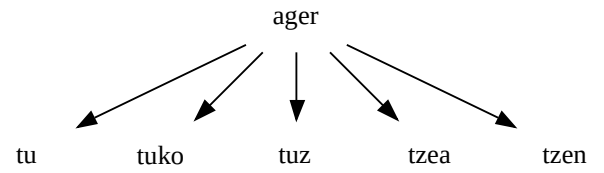


Figura 1. *Ejemplo de segmentación de las palabras cuyo lema es agertu (*aparecer*).*

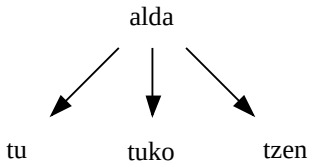


Figura 2. *Ejemplo de segmentación de las palabras cuyo lema es aldatu (*cambiar*).*

Aplicando estas separaciones, se obtuvo un nuevo corpus con el que entrenar un modelo de la aplicación.

3. BASE DE DATOS

La experimentación descrita en este trabajo se desarrolló empleando la base de datos METEUS [4], un corpus formado por predicciones meteorológicas diarias tomadas del Instituto Vasco de Meteorología durante 28 meses. Se tomaron las predicciones tanto en castellano como en euskera, obteniéndose así un corpus bilíngüe, planeado para realizar experimentación de traducción automática. Posteriormente este corpus fue dividido, de manera aleatoria, en un conjunto de entrenamiento y otro de test y se lematizo, permitiendo trabajar tanto con palabras como con lemas.

En la tabla 1 pueden verse las características más importantes del conjunto de entrenamiento en euskera de esta base de datos. Se da la información tanto para el caso de palabras como para el de lemas y raices más sufijos.

Cabe destacar especialmente el aumento significativo en la talla del vocabulario en euskera. Para esta misma tarea en castellano, el vocabulario no alcanzaba las 700 palabras. Sin embargo, para euskera es nétamente superior, debido a que, como ya se ha mencionado, el euskera es un idioma áltamente declinado. Esta alta declinación queda reflejada observando la talla del vocabulario en los otros

	Palabras	Lemas	Raices+Sufijos
Frases	14615		
Nº palabras	154778		
Vocabulario	1098	426	588
Longitud media	10.59		

Tabla 1. Características más importantes del corpus de entrenamiento en euskera de la base de datos.

Número de locutores	36 (20 hombres + 16 mujeres)
Origen	17 lengua materna 19 lengua no materna
Dialecto	22 *batua* 10 vizcaino 4 guipuzcoano
Número de elocuciones	1800
Frases diferentes	500
Tamaño	3.5 horas
Longitud media elocución	7 segundos

Tabla 2. Características más importantes de la base de datos METEUS de voz en euskera.

dos casos. Como se puede observar, el número de lemas es inferior a la mitad del número de palabras (y del mismo orden que para el castellano, con 414 lemas) y para la metodología basada en raices y sufijos comunes, se produce una disminución en la talla del vocabulario de más de 500 palabras, debido a que muchas palabras comparten los mismos sufijos debido a esta alta declinación.

Posteriormente se decidió grabar el corpus de test tanto para euskera como para castellano. En total, se emplearon 36 locutores bilingües en la grabación. Del conjunto de test se tomaron las frases distintas no contenidas en el entrenamiento, 500 en total. Estas 500 frases diferentes se dividieron en bloques de 50 frases y cada locutor grabó las frases de uno de los bloques, resultando en que los seis primeros bloques fueron grabados por cuatro locutores diferentes, mientras que los otros cuatro, únicamente por tres locutores. En total, el corpus de voz está compuesto por algo de tres horas para cada uno de los idiomas.

En la tabla 2 pueden verse las características más importantes de la parte de voz de esta base de datos, incluyendo una descripción de la distribución de los locutores participantes en el corpus de voz. Estos datos están referidos únicamente a la parte de euskera.

4. RESULTADOS EXPERIMENTALES

4.1. Condiciones experimentales

Para realizar la experimentación, la base de datos fue parametrizada en 12 coeficientes cepstrales en frecuencia Mel, con sus correspondientes derivadas y segundas derivadas, energía y derivada de la energía. Se definieron por tanto cuatro representaciones acústicas. La longitud de la ventana de análisis fue de 25 milisegundos y el desplazamiento de la ventana, de 10 milisegundos.

Cada unidad fonética se modeló mediante modelos ocultos de Markov continuos de tres estados, con arcos de izquierda a derecha de un estado al siguiente y autolazos en cada estado. Se emplearon modelos de 32 gaussianas por estado y representación acústica. Se emplearon un total de 35 unidades fonéticas [5, 6].

Para representar los modelos de lenguaje empleados

	Tasa de error
PALABRAS	5.10
LEMAS	16.31
RAICES+SUFIJOS	5.93

Tabla 3. Tasa de error obtenida para el sistema base basado en palabras, el sistema que hace uso de los lemas diréctamente y el sistema basado en las raices y los sufijos obtenidos por medio de los lemas.

	Coste temporal
PALABRAS	1
LEMAS	1.05
RAICES+SUFIJOS	0.95

Tabla 4. Coste temporal del proceso de reconocimiento.

en el sistema de reconocimiento, en lugar de emplear, como es típico, modelos de *n-gramas*, se emplearon *k-testables en sentido estricto* [7], ya que mantienen las restricciones sintácticas del lenguaje. Para los experimentos de la siguiente sección se emplearon modelos con $k = 3$.

4.2. Resultados de la experimentación

En la tabla 3 pueden verse los resultados de reconocimiento obtenidos para los tres sistemas estudiados, los dos basados en los lemas así como el sistema basado en palabras. De igual manera, la tabla 4 muestra el coste temporal del proceso de reconocimiento para los tres casos.

Se observa que el sistema basado en los lemas funciona bastante peor. No solo se produce un aumento considerable en la tasa de error, sino que además es ligéramente más lento. Este aumento en la tasa de error puede ser debido a que, al usar lemas, hay más posibles transiciones entre lemas que entre palabras, con lo que se produce una dispersión a nivel de modelo de lenguaje. Además, se pierde la relación entre las palabras y se le pide a los modelos acústicos que diferencien entre palabras generalmente similares dentro de un lema.

Para el caso basado en raices y sufijos, las tasas de error obtenidas son similares que para el caso base de usar palabras. Como contrapartida, al reducir tan drásticamente la talla del vocabulario, el tiempo de cómputo del sistema de reconocimiento se reduce en algo más de un 5 %, que aproximadamente supone ahorrarse de media unas cuatro décimas de segundo en cada elocución reconocida. Es decir, sacrificamos algo de precisión a costa de hacer el sistema más rápido. Esto puede ser de especial interés en tareas de mayor dificultad y con un vocabulario de mayor tamaño, en el que el efecto de reducción de la talla del vocabulario podría ser mucho mayor y, por consiguiente, se prodría aligerar sensiblemente el proceso de reconocimiento.

Hay que hacer mención también en otro de los problemas o ventajas que puede acarrear la metodología basada en obtener los sufijos comunes. Un sistema de reconocimiento del habla genera siempre frases formadas por palabras del vocabulario. En este caso, al emplear raices y sufijos en el vocabulario, es necesario deshacer la separación al finalizar el reconocimiento. En ese proceso, puede ocurrir que se generen palabras que no existen en el vocabulario original, ya que se puede reconocer un sufijo tras una raiz no observada en entrenamiento debido al suavizado del modelo de lenguaje. Este efecto puede ser positivo o negativo. Por un lado, podría darse el caso en que esos sufijos sirviesen para generalizar una raiz y darle opciones no vistas en un entrenamiento con palabras. Por ejemplo, siguiendo con los ejemplos de las figuras 1 y 2, podría darse el caso, como el mostrado en la figura 3, de que para el lema *gainditu* (superar) se hubiesen encontrado las palabras *gainditu* y *gainditukо*, es decir, al segmentar se obtendría la raíz *gaindi* y los sufijos *tu* y *tuko*. Debido a que para *agertu* y *aldatu* se han observado otros sufijos extra, sería posible que se reconociesen también *gaindituz*, *gainditzea* y *gainditzen*, palabras totálmente correctas, y que el sistema base basado en las palabras no podría hacer. Por otra lado, siguiendo con la misma idea, se podrían generar también palabras que no existen en el idioma, produciendo un efecto indeseado en un sistema de reconocimiento automático del habla. Concretamente, en la experimentación se observaron más de 100 palabras no presentes en el vocabulario original, de las cuales únicamente en torno a una quincena son correctas.

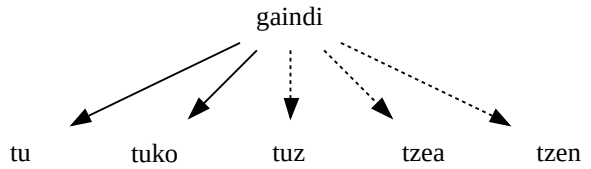


Figura 3. *Ejemplo de posible generalización que se podría obtener para otros lemas, como por ejemplo gainditu (*superar*). En trazo continuo se muestran los sufijos supuestamente encontrados en el entrenamiento y en discontinuo, los posibles sufijo extra que se podrían reconocer sin haber sido observados en entrenamiento, obteniendose palabras totálmente válidas.*

Para evitar estos efectos se siguio una aproximación sencilla preliminar en la que se impedían transiciones potencialmente problemáticas. El vocabulario de la tarea está formado por tres tipos de palabras: palabras no segmentadas, raices y sufijos. A las palabras no segmentadas se le impiden transitar a los sufijos. A las raices sólo se les permiter transitar a los sufijos y a los sufijos, tanto a las palabras no segmentadas como a las raices. Para las transiciones de raices a sufijos sólo se permitieron aquellas observadas en el entrenamiento, es decir, se impidió el tipo de generalización positiva descrita en anterioridad por medio de la figura 3. Los resultados obtenidos siguiendo

Tasa de error	Coste temporal
6.31	0.85

Tabla 5. Tasa de error y coste de procesado obtenidos cuando se impiden transiciones potencialmente problemáticas en el sistema basado en raices y sufijos.

esta aproximación se muestras en la tabla 5.

Se observa un aumento de las tasas a pesar de impedir transiciones a priori no posibles. A pesar de ello, esta modificación forzada del suavizado del modelo de lenguaje provoca un amplio descenso en los costos de procesado, que podría ser interesante en tareas de mayor dificultad. En este caso, únicamente aparecen cuatro palabras no presentes en el vocabulario original, que corresponden a raices que se reconocen al final de la elocución.

5. AGRADECIMIENTOS

Queremos agradecer al grupo *Ametzagaiña* y, en particular, a Josu Landa por proveernos con el corpus METEUS lematizado, que hizo posible este trabajo.

6. CONCLUSIONES Y TRABAJO FUTURO

En el presente trabajo se ha presentado una metodología para hacer reconocimiento automático del habla en euskera haciendo uso de información lingüística proporcionada por la base de datos lematizada. Para el caso del sistema basado en separar las palabras en raices y sufijos, se obtienen unos resultados similares que al sistema base basado en palabras, pero al reducir la talla del vocabulario se produce una disminución en el tiempo de cómputo del sistema de reconocimiento. En el sistema basado en agrupar palabras por lemas, se produce una dispersión a nivel de modelo de lenguaje y se pierde la relación entre las palabras, resultando en un aumento no sólo de la tasa de error, sino también del coste temporal.

En el futuro se desearía aplicar esta metodología a una tarea de mayor complejidad y con un vocabulario más amplio, en la que la reducción de la talla del vocabulario sería más drástica y se pudiese obtener un beneficio mayor. Este estudio permitiría además dar una valoración definitiva al uso de esta metedología. También sería interesante estudiar otros posibles métodos para impedir la aparición de palabras inexistente en el idioma, sin por ello limitar la posibilidad de generar palabras válidas por medio de uniones de raices y sufijos no vistos en entrenamiento.

7. BIBLIOGRAFÍA

[1] I. Aduriz, M. Aranzabe, J. Arriola, A. Díaz de Ilarraza, K. Gojenola, M. Oronoz, y L. Uria, "A cascaded syntactic analyser for basque," in *Computational Linguistic and Intelligent Text Processing*, Berlin, 2004, pp. 124–135, Springer-Verlag, LNCS 2945.

[2] Itziar Aduriz, Izaskun Aldezabal, Iñaki Alegria, Xabier Artola, Nerea Ezeiza, y Ruben Urizar, "Euslem: A lemmatiser/tagger for basque," in *Proceedings of EURALEX'96, Part I*, Goteborg (Sweden), 1996, pp. 17–26.

[3] A. Pérez, I. Torres, y F. Casacuberta, "Towards the improvement of statistical translation models using linguistic features," in *Proceedings of the FinTAL - 5th International Conference on Natural Language Processing. Lecture Notes in Computer Science 4139*, Turku, Finland, 23-25 August 2006, pp. 716–725.

[4] A. Pérez, I. Torres, F. Casacuberta, y V. Guijarrubia, "A Spanish-Basque weather forecast corpus for probabilistic speech translation," in *5th SALTMIL Workshop on Minority Languages*, Genoa, Italy, 23 May 2006, pp. 99–101.

[5] V. Guijarrubia, I. Torres, y L.J. Rodríguez, "Evaluation of a spoken phonetic database in basque language," in *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, 2004, vol. 6, pp. 2127–2130.

[6] Karmele López de Ipiña, Inés Torres, Lourdes Oñederra, y Luis Javier Rodriguez, "Selection of sublexical units for continuous speech recognition of basque," in *Proc. of International Conference of Spoken Language Processing*, Beijing, 2000.

[7] I. Torres y A. Varona, "k-tss language model in a speech recognition system," *Computer Speech and Language*, pp. 15(2):127–149, 2001.

SEGMENTACIÓN DE FONEMAS NO SUPERVISADA BASADA EN MÉTODOS KERNEL DE MÁXIMO MARGEN

Yago Pereiro Estevan[1*], *Vincent Wan* [2]

Universidad Carlos III de Madrid[1]
Departamento de Teoría de la
Señal y Comunicaciones
Avda. de la Universidad, 30,
28911-Leganés (Madrid), ESPAÑA

Odette Scharenborg[2], *Ascensión Gallardo Antolín* [1]

University of Sheffield [2]
Department of Computer Science
211 Portobello Street,
Sheffield S1 4DP, UK.

RESUMEN

En este artículo se desarrolla un método automático de segmentación de fonemas no supervisado. Este método utiliza el algoritmo de agrupación de máximo margen [1] para realizar segmentación de fonemas sobre habla continua sin necesidad de información a priori para el entrenamiento del sistema.

1. INTRODUCCIÓN

En los últimos años ha quedado patente que la utilización de las máquinas de vectores soporte (*Support Vector Machines*, SVMs) [2, 3] en el campo del habla es una sólida línea de futuro. Existen campos, como la verificación de locutor [4], en los que las SVMs obtienen resultados muy competitivos, como se muestra en las evaluaciones del *NIST* [5]. Sin embargo, en otros campos como el reconocimiento automático del habla, su utilización todavía no es muy común.

Un algoritmo relativamente nuevo y muy interesante en este sentido es el algoritmo de agrupamiento de máximo margen (*Maximum Margin Clustering*, MMC) [1], que representa una aproximación no supervisada de las SVMs, consiguiendo localizar fronteras de máximo margen cuando el etiquetado no está disponible. Por lo tanto, el análisis del comportamiento de este algoritmo sobre la señal de voz es una tarea interesante.

En este artículo analizaremos la segmentación a nivel fonético de la voz. Sin embargo, el objetivo último no es la segmentación fonética en sí, si no el agrupamiento de la señal de voz en unidades que posteriormente puedan ser clasificadas por SVMs.

Desarrollaremos un análisis que nos permita realizar la segmentación fonética sin necesidad de un conocimiento a priori de la secuencia de fonemas contenida en la señal de voz. No se utiliza, como en otros métodos [6], la transcripción de la frase para determinar las fronteras entre fonemas. Tampoco es necesaria una secuencia de entrenamiento anterior a la aplicación del algoritmo.

El algoritmo se divide en tres etapas: en primer lugar, se realiza un preprocesado de la señal de voz, posteriormente, sobre el conjunto de vectores de parámetros que caracterizan la señal se aplica un algoritmo de agrupación por segmentos temporales basado en [1] y finalmente se realiza un postprocesado sobre dicho algoritmo con el fin de obtener la segmentación definitiva.

A continuación en la secciones 2 y 3 explicaremos brevemente la etapa de preprocesado de la señal y el algoritmo de agrupamiento utilizados respectivamente. En la sección 4 desarrollaremos el método de análisis de la señal de voz preprocesada y finalmente en la sección 5 explicaremos el postprocesado de dicho análisis con el fin de obtener la segmentación. En las secciones 6 y 7 explicaremos los experimentos realizados, las conclusiones obtenidas y las líneas futuras de trabajo.

2. PREPROCESADO DE LA SEÑAL DE VOZ

A partir de la señal de voz inicial, se extraen una serie de vectores de parámetros que contendrán las principales características de la voz. Entre los diferentes tipos de parametrización existentes [7], para nuestro algoritmo hemos escogido la parametrización MFCC (*Mel Filtering Cepstral Coefficients*) que extrae una serie de coeficientes derivados del espectro de la señal que son pasados por un banco de filtros en escala *mel*, que aproxima la percepción auditiva humana. En la figura 1 se puede ver la etapa de parametrización de la señal de voz del estándar ETSI ES 202 050 [8].

Es crítico encontrar un compromiso entre el desplazamiento entre vectores de parámetros y el tamaño de la

*Este trabajo ha sido realizado durante el periodo de investigación de Yago Pereiro Estevan en el departamento de ciencias de la computación de la universidad de Sheffield.

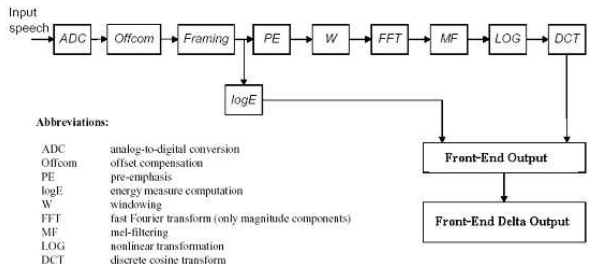


Figura 1. *Diagrama de bloques del sistema de parametrización para reconocimiento perteneciente al estándar ETSI ES* 202 050.

ventana de análisis de cada uno. Tomaremos segmentos de $15ms.$, en los cuales se puede considerar la señal casi estacionaria y, aplicando un solapamiento de $10ms.$ entre tramas, extraeremos vectores que caractericen cada segmento de señal de voz.

Inicialmente trabajaremos con vectores de 39 parámetros: 12 MFCCs más la log-energía y las correspondientes primera y segunda derivadas.

De este modo, la señal de voz quedará preprocesada obteniendo un conjunto de tramas de la forma

$$\mathbf{x}_i[n]_{i=1}^{k},$$

donde $n = 1 \dots N$ es el número de tramas y $i = 1 \dots k$ son los k coeficientes MFCCs de cada trama.

3. ALGORITMO DE AGRUPAMIENTO DE MÁXIMO MARGEN

El método de agrupamiento de máximo margen (MMC) [1] parte del principio de aprendizaje máquina de clasificación de máximo margen y lo modifica de modo que se pueda hacer una aproximación del mismo sin necesidad de que haya una etapa de aprendizaje supervisado o semi-supervisado. Esta modificación combinada con la utilización de métodos kernel consigue resultados a menudo mas eficientes que los métodos convencionales de agrupamiento espectral.

El objetivo de las máquinas de vector soporte (SVMs) es, dado un conjunto de entrenamiento etiquetado (x^1, y^1) ,. . ., (x^N, y^N) donde cada ejemplo es asignado a una clase, encontrar el discriminante lineal $f_{w,b}(x) = w^T\phi(x) + b$ que maximice el margen mínimo de no clasificación

$$\gamma^* = \max_{\omega,b,\gamma} \gamma \text{ dado } y^i(\omega^T\phi(x^i) + b) \geq \gamma, \forall_{i=1}^{N}, \|\omega\| = 1 \quad (1)$$

donde la constante de normalización euclídea sobre ω asegura que la distancia entre los datos y el hiperplano de separación (en el espacio de $\phi(\mathbf{x})$ determinado por ω^*,b^*.

El método MMC relaja las restricciones de máximo margen obteniendo una aproximación de máximo margen "blanda"de modo que a partir de un conjunto $x^1, \dots, x^N$ se consiga un etiquetado que si, posteriormente se utilizara en una SVM se consiguiese el de máximo margen entre todos los posibles.

4. ALGORITMO DE ANÁLISIS DE LA SEÑAL PARAMETRIZADA

Una vez parametrizada la señal de voz aplicaremos el algoritmo de agrupamiento sobre una ventana de análisis que iremos desplazando sobre el conjunto de vectores de parámetros, de modo que los vectores de parámetros de cada ventana queden agrupados en dos clases $y_j \in \{C_1, C_2\}$, $j = 1, \dots, m$ donde m es el número de vectores de parámetros analizado en cada ventana.

Posteriormente analizaremos la información obtenida tras dichos agrupamientos con el fin de realizar la detección de las fronteras entre fonemas. En este paso es crítica, tanto la selección del número de vectores de parámetros analizados en cada ventana como la correcta configuración de los parámetros modificables en el algoritmo de agrupamiento utilizado en la sección 3 ya que de esta selección depende el nivel en el que se analiza la señal de voz.

Tras el deslizamiento de la ventana de análisis sobre la señal parametrizada se obtendrá una matriz, en cuyas columnas se encuentra el etiquetado de cada vector de parámetros. Es decir, para cada ventana formada por los vectores de parámetros

$$\mathbf{x}_i[n]_{i=1}^{k} \quad (2)$$

con $n = N, \dots, N + m$ donde m es el tamaño de la ventana, se obtendrá un vector

$$\mathbf{y}_{j=1}^{m} \quad (3)$$

en el que cada elemento pertenecerá a una clase $y_j \in \{C_1, C_2\}$.

En la figura 2 se muestra una representación del agrupamiento realizado sobre un segmento de señal de voz parametrizada. En este caso, se ha utilizado una ventana de análisis de 19 vectores de parámetros. En cada columna de la figura se puede observar la clasificación obtenida sobre los vectores de parámetros de cada ventana analizada.

Podemos observar a partir de la figura 2 que se detectan estructuras que nos permiten distinguir donde hay un cambio significativo en la señal de voz. Estas estructuras

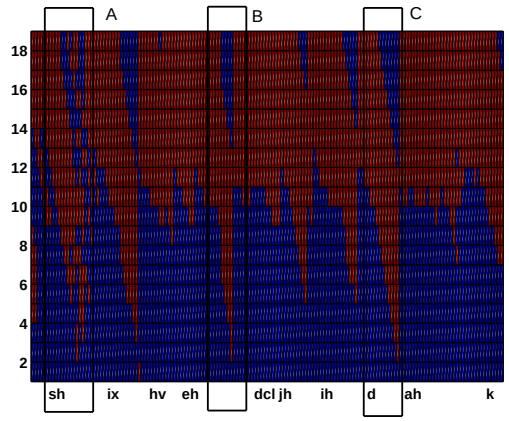


Figura 2. *Representación del agrupamiento por ventanas, en cada ventana vertical las diferencias de color indican la clasificación realizada. En el eje x se representa el comienzo de cada fonema con su nombre.*

son las que analizaremos para realizar la segmentación fonética.

5. SEGMENTACIÓN

En teoría, cuando la ventana quede centrada en una frontera el método de agrupamiento hará que la mitad anterior a la frontera quede agrupada en una clase y la mitad posterior en otra. Según nos separamos de la frontera el número de elementos pertenecientes al fonema sobre el que nos encontramos va creciendo.

Existen fronteras muy claramente definidas mientras que otras se pueden intuir pero no son tan claras. Esto puede ser debido, por ejemplo, a que el fonema sea demasiado breve o al hecho de que hay fonemas cuya separabilidad es mas compleja [9]. En la figura 2 se puede observar que hay estructuras como *C* que están claramente definidas de principio a fin. Sin embargo, hay otras estructuras como *A* que, a pesar de poder apreciarse que existe una frontera, ésta es mucho más ruidosa. En el caso concreto de *A* puede deberse a que al ser un fonema fricativo, la señal es mucho mas ruidosa. Finalmente aparecen estructuras como *B* donde no existe una frontera, lo que nos hace pensar que el algoritmo está detectando más cambios que los meramente fonéticos. Este último tipo de estructura es muy interesante ya que nos sugiere que el algoritmo MMC puede realizar un análisis a un nivel sub-fonético [10] y esto puede ser de gran utilidad para trabajos futuros.

A partir de las clasificaciones realizadas para cada ventana debemos analizar la información que nos permita encontrar la mejor segmentación posible. Como hemos visto, la información obtenida es muy compleja y muy rica, por lo tanto la obtención de un método de automatización no es un problema sencillo de resolver. A continuación presentaremos dos métodos desarrollados para realiar una aproximación a la automatización de la segmentación.

5.1. Método basado en estructuras

Una primera aproximación que se realiza es el análisis a partir de, únicamente, la matriz de vectores clasificados de modo que, a partir de las estructuras observadas y explicadas anteriormente podamos discernir donde se encuentra la frontera y realizar la correcta segmentación.

Para ello, en primer lugar se realiza una selección de aquellas ventanas en las que a cada lado de vector de parámetros central se agrupen los elementos de cada clase. Una vez seleccionadas esas ventanas se realiza un análisis basado en las ventanas siguientes con el fin de observar si aparece una estructura de deslizamiento en las clases.

Este es un problema de comparación con una máscara, es decir, comparar un segmento de ventanas siguientes con una matriz (máscara) que tenga en cada elemento las clases que esperamos. Dependiendo del tamaño de la máscara con la que comparemos, obtendremos resultados diferentes. Cuanto mayor es la máscara, menos patrones coincidirán, pero la probabilidad de que esos patrones no pertenezcan a una frontera será mucho menor. Por lo tanto hay que buscar un compromiso, y para ello se han realizado una serie de simulaciones.

Este método, como veremos en la sección de experimentos, presenta una gran fiabilidad, obteniéndose una tasa de falsa alarma muy baja, sin embargo y debido a que, como ya hemos explicado antes, hay estructuras que no aparecen de un modo claro, la probabilidad de pérdida de fronteras es demasiado elevada.

5.2. Cálculo de las distancias euclídeas

Debido a la riqueza y complejidad de las estructuras, la automatización mediante máscaras es muy compleja por lo tanto decidimos utilizar como ayuda el cálculo de la distancia euclídea entre los puntos medios de cada una de las clases, es decir, para cada trama i,

$$D_i = \sqrt{\sum_{i=1}^{k} (\bar{x}_i^{C_1} - \bar{x}_i^{C_2})^2}, \qquad (4)$$

donde

$$\bar{\mathbf{x}}^{C_i} = \frac{1}{N_{C_i}} \sum_{j}^{N_{C_i}} \mathbf{x}_j, \qquad (5)$$

siendo C_1, C_2 son las dos clases posibles y N_{C_i} es el número de elementos de la clase C_i.

Los resultados obtenidos, como se puede ver en la figura 3, nos permiten ver que la distancia es mayor en las fronteras, de modo que si aplicamos un sencillo algoritmo de detección de máximos locales obtenemos un método de segmentación automática.

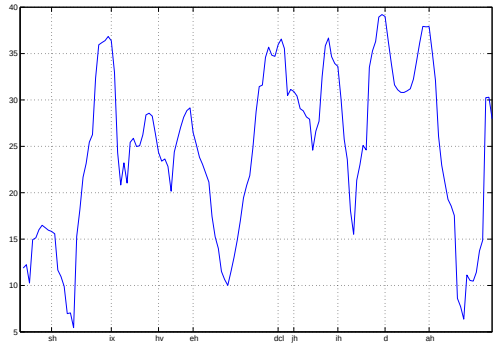


Figura 3. *Representación de las distancias euclídeas entre clases. En el eje x se representa el comienzo de cada fonema con su nombre.*

En la sección de experimentos, veremos que la cantidad de fronteras perdidas es baja, sin embargo al ser la señal de distancias, ruidosa, se introduce una mayor tasa de falsas alarmas.

5.3. Combinación

Para resolver los problemas de los métodos anteriores hemos optado por combinar ambos métodos. En la actualidad estamos desarrollando un método que nos permita combinar ambos métodos de modo que minimicemos tanto la probabilidad de falsa alarma como la de pérdida de fronteras.

6. EXPERIMENTOS

Para este estudio se utilizó un subconjunto de la base de datos Americano-Inglesa DARPA-TIMIT utilizando 4 frases pertenecientes a 4 locutores diferentes, dos masculinos y dos femeninos. Dicha base de datos está muestreada a $16\ KHz$.

El primer paso de la experimentación consistió en optimizar la selección de los diferentes parámetros utilizados tanto en la parte de preprocesado como en el algoritmo de análisis de la señal parametrizada (secciones 2 y 4). En concreto, en lo que se refiere a la parte de preprocesado de la señal de voz, se realizaron pruebas sobre el tamaño de las ventanas de parametrización, así como el solapamiento entre tramas como ya explicamos en la sección 2.

Asimismo inicialmente se realizaron simulaciones modificando el kernel utilizado en el algoritmo MMC, con el fin de optimizar la respuesta del sistema en relación al problema de segmentación de fonemas. Tras estas pruebas decidimos utilizar un kernel *RBF* con $\sigma = 200$. Finalmente realizamos pruebas sobre el tamaño de la ventana de análisis utilizada, decidiendo finalmente utilizar 9 vectores de parámetros a cada lado del vector central, es decir ventanas de 19 vectores de parámetros en total.

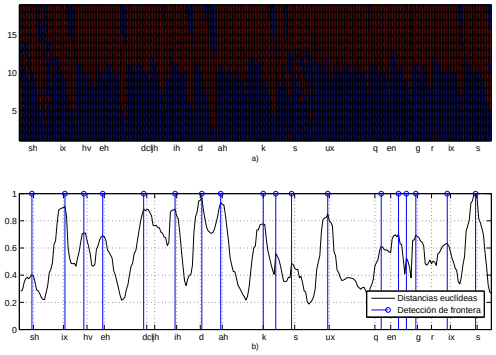


Figura 4. *a) Representación del agrupamiento por ventanas en un segmento de voz. b) Representación de las distancias euclideas entre clases para el mísmo segmento de voz que a) y segmentos detectados aplicando un método de detección de máximos locales.*

En la figura 4 se puede observar a simple vista que la combinación de ámbos métodos permite una segmentación los fonemas bastante precisa, en la parte *a)* se representa la clasificación de las diferentes ventanas, mientras que en la parte *b)* se representan las distancias euclídeas entre cada clase, para cada ventana de análisis, así como los máximos locales detectados.

En la tabla 1 se observan los resultados sobre el método de distancias euclídeas, con una distancia del etiquetado realizado por TIMIT de $\pm 20ms$. En cada columna se obtienen los resultados obtenidos para varias sensibilidades diferentes en la detección de máximos locales. Se puede observar que alcanzanzamos una detección de fronteras razonablemente alta, sin embargo la tasa de falsa alarma aumenta al aumentar la sensibilidad en la detección de máximos debido a lo ruidosa que es la señal.

En la tabla 2 se observan los resultados sobre el método de análisis de estructuras, con una distancia del etiquetado realiazdo por TIMIT de $\pm 20ms$. En cada columna se obtienen los resultados obtenidos con diferentes tamaños de la matríz máscara. Podemos ver que para máscaras muy grandes llegamos a eliminar la falsa alarma, pero debido a que aparecen muy pocas estructuras tan claras, el

	Conf. 1	**Conf.** 2	**Conf. 3**
Front. detect. (%)	84,68	78,99	71, 87
Falsas front. (%)	52,66	35,31	19,52

Tabla 1.
Valores de detección de frontera utilizando el método de las distancias euclídeas entre vectores de parámetros de diferentes clases. Cada columna muestra los resultados para una configuración en la sensibilidad de la detección de máximos locales.

	5×5	4×4	3×3
Front. detect. (%)	8,95	15,65	25,21
Falsas front. (%)	0	5,2	13,27

Tabla 2.
Valores de detección de frontera utilizando el método de detección de estructuras. Cada columna muestra los resultados para una configuración diferente de la matriz máscara.

	$Comb$
Front. detect. (%)	87,1
Falsas front. (%)	32,61

Tabla 3.
Valores de detección de frontera utilizando el método una combinación del método de distancias euclídeas y del método de detección de estructuras.

porcentaje de detección es muy bajo. Según disminuye el tamaño de la máscara, mejora la detección, pero también aumenta la falsa alarma.

Finalmente en la tabla 3 mostramos los resultados de una posible combinación de ambos métodos. Consideramos que hay una frontera donde cualquiera de los métodos haya detectado una frontera y eliminamos aquellas fronteras que estén juntas (en tramas consecutivas). Este es un método muy sencillo de combinación, pero se observa que la probabilidad de pérdida de tramas desciende y aunque se perciba un aumento de la falsa alarma, este aumento se debe a que el método introduce puede introducir en muchos casos como dos fronteras diferenciadas la misma frontera detectada en diferentes lugares (no consecutivos). Esta combinación nos permite intuir que la combinación de ambos métodos de un modo mas complejo puede mejorar los resultados ya que se pueden detectar mas fronteras (como ya veiamos en la figura 4) y la falsa alarma puede disminuirse.

7. CONCLUSIONES

Estamos comenzando una línea de investigación interesante en la que tratamos de estudiar la aplicabilidad del algoritmo de agrupamiento de máximo margen a tareas relacionadas con el procesamiento de la señal de voz. En concreto, en este artículo nos hemos centrado en la segmentación de fonemas pero como ya comentamos en la introducción, esta línea de trabajo abre un amplio abanico de posibilidades.

Los resultados obtenidos en la tarea de segmentación fonética de habla continua nos permiten ser muy optimistas. Se puede observar que tenemos la información necesaria para recuperar las fronteras entre fonemas, pero hay que mejorar la etapa de postprocesado para la extracción automática de dicha información. Podemos decir que, aunque estamos en una fase inicial, los resultados son relativamente buenos dada la sencillez de los algoritmos presentados. En la actualidad, estamos trabajando en la mejora del algoritmo de extracción de las marcas entre fonemas basado en la combinación entre el método de análisis de estructuras y el basado en las distancias euclídeas.

Respecto a otras líneas de trabajo futuro, y puesto que en la experimentación realizada hasta ahora hemos observado que el algoritmo parece capaz de realizar un análisis de voz a nivel subfonético, tenemos planeado su aplicación al análisis de características articulatorias (AF) [11]. Este es un campo menos explorado, pero muy prometedor en el ámbito de reconocimiento automático del habla, en el que la técnica AF aparece como una alternativa más flexible para la modelización de la variación de la voz [12]. Además, en este contexto, todavía no se ha desarrollado un algoritmo de etiquetado automático.

8. BIBLIOGRAFÍA

[1] L. Xu, J.Ñeufeld, B. Larson, y D. Schuurmans, "Maximum margin clustering," *in Advances in Neural Information Processing Systems*, vol. 17, pp. 1537–1544, 2005.

[2] B. Schölkopf y A. Smola, *Learning with kernels*, MIT Press, Cambridge, MA, 2002.

[3] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.

[4] K. Daoudi y J. Louradour, "Conceiving a new sequence kernel and applying it to svm speaker verification," *Proc. Interspeech*, pp. 3101–3104, 2005.

[5] National Institute of Standards y Technology, "The nist year 2006 speaker recognition evaluation plan," 2006.

[6] Bryan L. Pellom y John H. L. Hansen, "Automatic segmentation of speech recorded in unknown noisy channel characteristics," *Speech Communication*, vol. 25, pp. 97–116, 1998.

[7] Christophe Lévy, Georges Linarès, Pascal Nocera, y Jean-François Bonastre., "Reducing computational and memory cost for cellular phone embedded speech recognition system," *IEEE International Conference on Acoustic, Speech and Signal Processieng*, 2004.

[8] ETSI Standard:ETSI ES 202 050., "Speech processing, transmission and quality aspects (stq); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms dsr advanced front end," 2002.

[9] Maria-Barbara Wesenick y Andreas Kipp, "Estimating the quality of phonetic transcriptions and segmentations of speech signals," *Internacional Conference on Spoken Language Processing (ICSLP)*, 1996.

[10] O. Scharenborg, V. Wan, y R. Moore, "Capturing fine-phonetic variation in speech through automatic classification of articulatory features," *ITRW Workshop on Speech Recognition and Intrinsic Variation*, 2006.

[11] P. Taylor y S. King, "Detection of phonological features in continuous speech usin neural networks," *Computer Speech and language*, vol. 14, pp. 333–353, 2000.

[12] M.Ostendorf, "Moving beyond the 'beads-on-a-string model of speech," *Proceedings of the IEEE Automatic Speech Recognition and Understandig Workshop, Keystone, CO*, pp. 79–84, 1999.

ON THE USE OF MLP-DISTANCE TO ESTIMATE POSTERIOR PROBABILITIES BY KNN FOR SPEECH RECOGNITION

Ana I. García Moral, Carmen Peláez Moreno

EPS-Universidad Carlos III
Departamento de Teoría de la Señal
y Comunicaciones
Avd. de la Universidad 30,
28912 Leganés, Madrid

Hervé Bourlard

IDIAP Research Institute
CP 592, rue du Simplon 4,
1920 Martigny, Switzerland

ABSTRACT

In this work we try to estimate a posteriori probabilities needed for speech recognition by the K-Nearest-Neighbors rule (KNN), using a Multi-Layered Perceptrom (MLP) to obtain the distances between neighbors. Thus, we can distinguish two different works: on the one hand, we estimate a posteriori probabilities using KNN with the aim of using them in a speech recognition system [1][2]; and, on the other hand, we propose a new distance measure, MLP-distance.

1. INTRODUCTION

Typical Automatic Speech Recognition (ASR) systems use features obtained from short-term spectrum, like Mel-Frequency Cepstral Coefficients (MFCC) or Perceptual Linear Prediction (PLP). Phoneme posterior probabilities can also be used as features, being more stable and robust [1][2].

There are several types of non-parametric methods of interest in pattern recognition. Some of them estimate the density functions $p(x|\omega_j)$ - *the class-contidional probability density* function (probability density function for x given that the state of nature is ω_j)- from sample patterns.

Some other alternatives directly estimate the a posteriori probabilities $P(\omega_j|x)$ - the probability of the state of nature being ω_j given that feature value x has been measured. This is closely related to non-parametric design procedures, such as the nearest-neighbor rule, which bypasses explicit probability estimation and goes directly to decision functions. Besides, there are also non-parametric procedures for transforming the feature space in the hope that it may be possible to employ parametric methods in the transformed space.

The KNN classifier is a very simple non-parametric method for classification. Despite the simplicity of the algorithm, it performs very well and is an important benchmark method. The KNN classifier, as described by [3], requires a distance metric d, a positive integer k, and the reference templates X_n of n labeled patterns.

Euclidean or Mahalanobis distances have been typically used as local distance between vectors. In this work we investigate the use of MLP-distance as a measure of local similarity between two vectors since the a posteriori probabilities vector can be seen as a distribution over the phoneme space.

This work must be considered as an experiment to evaluate the effectiveness of the KNN algorithm for estimating a posteriori probabilities, but about all, it must be considered as a preliminary experiment to evaluate the potential usefulness of an MLP as distance measurer between vectors. So, we are going to train an MLP with only two input vectors and one output, predicting whether or not (0/1 output) the two input vectors belong to the same class. Using the softmax output of the MLP, we will obtain some kind of non-linear 'distance' between input vectors, which can be used to replace the usual Euclidean distance used in KNN.

In Figure 1 is shown a block diagram of the system, which can be divided into 3 main blocks. The first, feature extraction (for obtaining a posteriori probabilities), in which we do not get into details, can be replaced by other or even omitted, using for example directly PLP features. And the second and third blocks, proposed MLP$_d$ (an MLP as distance measurer) and KNN probabilities a posteriori estimation, are described in detail in sections 3 and 2, respectively.

The document is organized as follows: Section 2 describes the KNN rule and its application as posterior probabilities estimator, Section 3 introduces the proposed MLP-distance, Section 4 shows experiments and results and finally, Section 5 gives conclusions and some ideas for future work.

2. K_N-NEAREST-NEIGHBORS ESTIMATION

To estimate $p(x)$ from n training vectors or prototypes, we can center a cell about x and let it grow until it captures k_n samples, where k_n is some specified function of n (distance function). These samples are the k_n nearest neighbors of x. It the density is high near x, the cell will

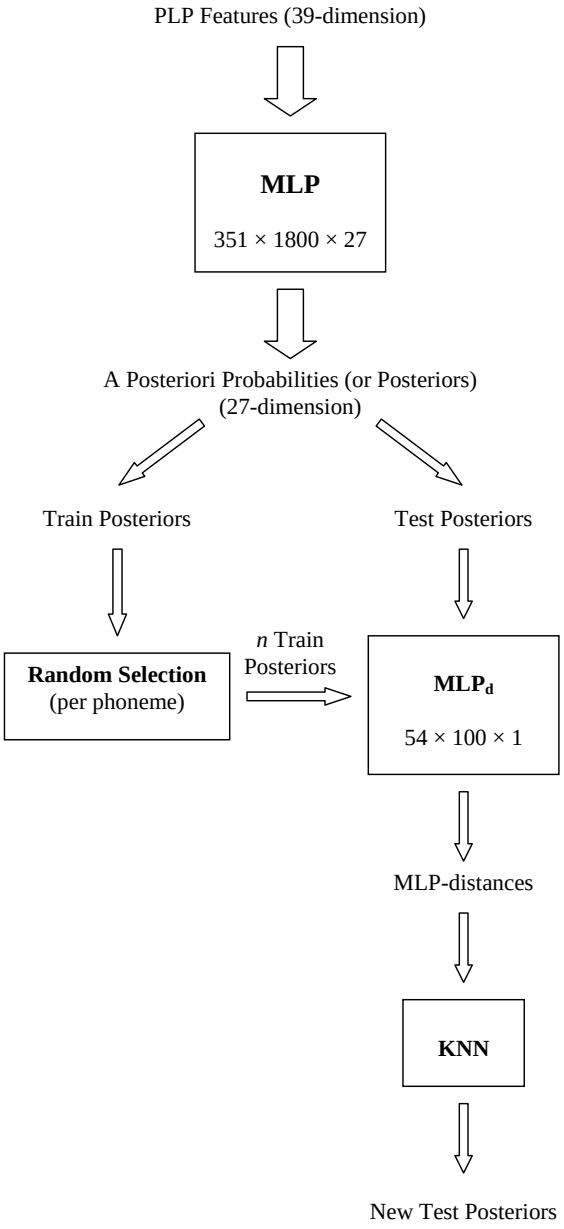


Figure 1. *Block diagram of the system.*

be relatively small, which leads to good resolution. If the density is low, it is true that he cell will grow large, but it will stop soon after it enters regions of higher density. In either case, if we take

$$p_n(x) = \frac{k_n/n}{V_n} \tag{1}$$

We want k_n to go to infinity as n goes to infinity, since this assures us that $\frac{k_n}{n}$ will be a good estimate of the probability that a point will fall in the cell of volume V_n. However, we also want k_n to grow sufficiently slowly that the size of the cell needed to capture k_n training samples will shrink to zero. Thus, it is clear from (1) that the radio must go to zero. Although we shall not supply a proof, it can be shown that the conditions $\lim_{n\to\infty} k_n = \infty$ and $\lim_{n\to\infty} k_n/n = 0$ are necessary and sufficient for $p_n(x)$ to converge to $p(x)$ in probability at all points where $p(x)$ is continuous. If we take $k_n = \sqrt{n}$ and assume that $p_n(x)$ is a reasonably good approximation to $p(x)$ we see from (1) that $V_n \approx \frac{1}{\sqrt{n}p(x)}$.

2.1. Estimation of a posterior probabilities

The technique discussed in the previous section can be used to estimate the a posteriori probabilities $P(\omega_j|x)$ from a set of n labeled samples by using the samples to estimate the densities involved. Suppose that we place a cell of volume V around x and capture k samples, k_i of which turn out to be labeled ω_i. Then the obvious estimate for the joint probability $p(x, \omega_i)$ is

$$p_n(x, \omega_i) = \frac{k_i/n}{V} \tag{2}$$

Thus, a reasonable estimate for $P(\omega_i|x)$ is

$$P(\omega_i|x) = \frac{p_n(x, \omega_i)}{\sum_{j=1}^{c} p_n(x, \omega_j)} = \frac{k_i}{k} \tag{3}$$

That is, the estimate of the a posteriori probability that ω_i is the state of nature is merely the fraction of the samples within the cell that are labeled ω_i. For minimum error rate, we select the category most frequently represented within the cell. If there are enough samples and if the cell is sufficiently small, it can be shown that this will yield performance approaching the best possible.

When it comes to choosing the size of the cell we can use k_n-*nearest-neighbor* approach. V_n would be expanded until some specified number of samples, were captured, such as $k_n = \sqrt{n}$. As n goes to infinity an infinite number of samples will fall within the infinitely small cell. The fact that the cell volume could become arbitrarily small and yet contain an arbitrarily large number of samples would allow us to learn the unknown probabilities with virtual certainty and thus eventually obtain optimum performance. Interestingly enough, we shall now see that we can obtain comparable performance if we base our decision solely on the label of the single nearest neighbor of x [3].

3. MLP-DISTANCE

In KNN algorithm described in Section 2, k_n is some specified function of n. Usually, we calculate the Euclidean distance between the vector we want to classify and the n prototypes (or training vectors), with the aim of selecting only the k prototypes closest (with a smaller distance) to it. Once we have selected these k *nearest* vectors, we only need to vote according to the their targets, obtaining in this way the final posterior probabilities.

Now, instead of using Euclidean distance, we propose to implement a very simple MLP which decides if two input vectors belong to the same class or not. In fact, softmax output of the MLP will give us a number between 0 and 1, that is, if two inputs belong to the same class MLP-distance will be near to 0, and near to 1 if they belong to different classes. Thus, we use 0.5 as threshold.

4. EXPERIMENTS AND RESULTS

This work must be considered as a first experiment to evaluate the effectiveness of the MLP-distance as local distance between vectors. But also, we want to evaluate the a posteriori probabilities estimated by KNN using this distance.

4.1. Database

In the following sections, the results are reported on the Numbers95 [4] conected digit task. Numbers95 contains digit strings spoken in US English over a telephone channel and is a small vocabulary database. There are 30 word types in the database modelled by 27 context-independent phones. The training set consists of 3330 sentences (2996 sentences were used for training and 334 for cross-validation) and 2250 sentences were used for testing (see Table 1). Training and test were only performed on clean speech. The lexicon has 12 different words (from *zero* to *nine* plus *oh* and *silence*) with a single pronunciation per word.

4.2. Features

Short-term spectral-based features, such as MFCC or PLP, are traditionally used in ASR. They can be modeled by a mixture of Gaussians (a typical function used to estimate the emission distribution of a standard HMM system) and this is the reason why they have been successfully applied. Nevertheless, spectral-based features not only contain lexical information, but also knowledge about the speaker or environmental noise. This extra information is cause of unnecessary variability in the feature vector, which may decrease the performance of the ASR system. Thus, we can use a transformation of traditional acoustic vectors as features for ASR, i.e. a posteriori probabilites. A multi-layer perceptron can be trained to estimate the phone posterior probabilities based on spectral-based features. In this case, the MLP performs a non-linear transformation. Because of this discriminant projection, posteriors are known to be more stable [1] and more robust to noise (chapter 6 of [5]). Moreover, the databases for training the MLP and for testing do not have to be the same so it is possible to train the MLP on a general-purpose database and use this posterior estimator to obtain features for more specific tasks [6]. Also, phone posterior probabilities can be seen as phone detectors [7], making them a very suitable set of features for speech recognition systems since words are formed by phones. Despite their good properties, posterior features cannot be easily modeled by a mixture of Gaussians. In the what is called the Tandem approach [1], posteriors are used as input features for a standard HMM/GMM system. However, a PCA transform on the logarithm of the a posteriori probabilities has to be done previously to make them more Gaussian like and decorrelate the feature vector.

SET	♯ sentences	♯ PLP frames	♯ Posterior frames
Train	2996	449.877	425.909
Cross-Validation	334	48.720	46.048
Test	2250	336.433	318.433

Table 1.
Feature data sets.

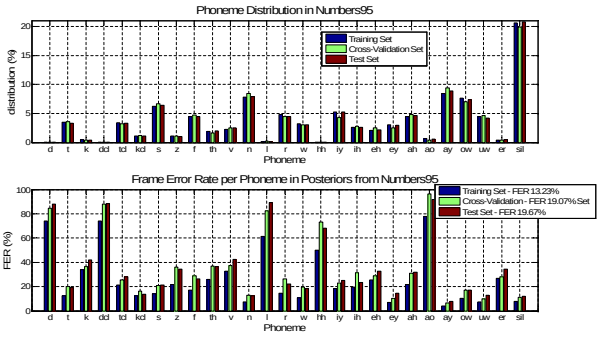


Figure 2. *Analysis in Posterior sets: a) Phoneme Distribution and b) Frame Error Rate per Phoneme in a posteriori probabilities.*

We work with 39 dimensional acoustic vectors, 13 static features (PLP) extracted using a window length of $32ms$ and a window shift of $12.5ms$, plus their delta and aceleration features. Posterior features were obtained using a MLP trained on Numbers95 (see Section 4.1). This MLP has 351 input nodes corresponding to the concatenation of 9 frames of 39 dimensional acoustic vectors (this is the reason why we lose 8 frames per sentence when we calculate a posteriori probabilities, as we can see in Table 1), one hidden layer with 1800 units, and 27 output units in output layer, each of them corresponding to a different monophone. After training MLP, train and cross-validation sets are also passed through the MLP (forward pass) to generate their a posteriori probabilities.

If we make a brief analysis in all a posteriori probability sets, we obtain Figure 2

4.3. Training MLP$_d$

The first design problem we face is to decide which n posterior probabilities from training set should be used for training MLP$_d$ (the one we use to calculate distances between posterior probabilities). To train MLP$_d$ we would

SET	♯ frames
Train	3273^2
Cross-Validation	299^2
Test	6662^2

Table 2.
MLP_d data sets.

Hidden size	FER (%)
50	1.85
100	1.98
500	5.92

Table 3.
Frame Error Rates (%)in MLP_d test (for a subset of 6662 frames).

have to confront every posterior vector (27 dimension) with itself and with the rest of posterior probabilities, therefore we would have N^2 training frames, being N the number of a posteriori probabilities in the training set (in this case 425.909, see Table 2). This quantity of data was too large for our purposes, so we have randomly selected a subset: approximately 120 frames per phoneme (more than 3 thousand posterior probabilities which will be transformed in more than 9 millions of training frames). We have also selected a subset for the test and cross-validation sets, in the same way (see Table 2).

The second design problem was to select the correct hidden size for the MLP_d. In Table 3 we show Frame Error Rates (FER) obtained for different hidden sizes. Our final decision was 100 hidden units. Although, at first sight, the use of only 50 hidden units can be considered the best choice, the better results obtained in the next step (KNN estimation described in section 4.4) justified our decision.

Analyzing the results showed in Table 3, we can see that MLP-distance is a very good method to compute local distances between vectors.

4.4. KNN

We began the posterior estimation via KNN, by selecting n vectors from the training set (n prototypes). Thus, we use as prototypes the 3273 training vectors used in MLP_d training (see Table 2). Once we have calculated the distances for all test vectors (between them and all the n train ones) using MLP_d, we try to fix k_n. As this process is very slow, we have only used 500 utterances from the test set to carry out this selection. In Figure 3 we can see the evolution of mean test frame error rate when k_n varies from 1 to 200. Thus, we can notice that better results are obtained considering only the 3 nearest training frames (being results for $k_n = 1$ and $k_n = 3$ very close to it).

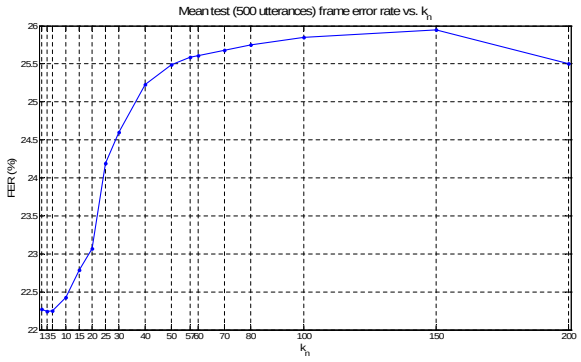


Figure 3. *Variation of mean test frame error rate with k_n.*

Experiment	Training frames	FER (%)
Baseline (Posteriors before KNN)	449.877	19.67
3-NN (Posteriors after KNN)	3.273	21.82

Table 4.
Frame Error Rate comparative.

Once the best k_n was selected, we applied KNN to obtain new posteriors as it was explained in sections 2 and 3. In Table 4 we show FER obtained before and after KNN (really, 3NN: 3-nearest neighbors) and, in Figure 4 is showed the confusion matrix of KNN-posteriors. So, we have an increase of approximately 2% in FER, but using only a 0.77% (see Tables 2 and 1) of the training samples which is a promising result.

5. CONCLUSIONS AND FURTHER WORK

As we mentioned before, this is a preliminary experiment to evaluate the effectiveness of both MLP as distance measurer (MLP-distance) and KNN as posterior estimator.

For the MLP-distance, we obtained very good results using a posteriori probabilities as features. But, it is true, that we must carry out experiments with other features (e.g. PLPs) and also, with other distance measures, as the Kullback-Leibler (KL) divergence [8] or the classical Euclidean distance.

On the other hand, we also obtained very good results using KNN as posterior estimator, mainly considering the quantity of data (less than 1% total training frames) used in KNN estimation, as we can see in Table 4. But, as we have already said in previous sections, this is only a preliminary experiment.

Further work includes the following:

- Use PLP features to train MLP_d and also to ob-

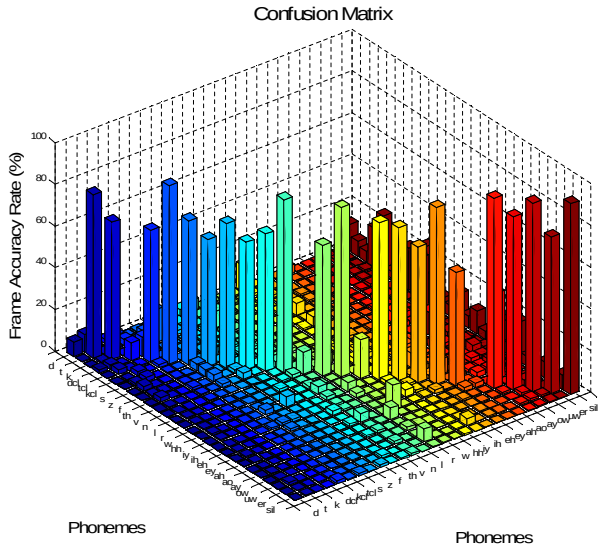


Figure 4. *Confusion Matrix for KNN-posteriors.*

tain posteriors with KNN, and compare results with those which use posterior features.

- Analyze other distances, as KL or Euclidean, and compare them with MLP-distance.
- Increase the number of prototypes n in KNN.
- Include these a posterior probabilites in a complete Tandem system to obtain word error rates (WER).

6. REFERENCES

[1] H. Hermansky, D. Ellis, and S. Sharma, "Tandem Connectionist Feature Extraction for Conventional HMM Systems," *Proccedings of ICASSP*, 2000.

[2] Q. Zhu, "On Using MLP features in LVCSR," *Proccedings of ICSLP*, 2004.

[3] R. O. Duda, P.E. Hart, and D. G. Stork, *Pattern Classification*, Wiley, 2nd edition, 2001.

[4] J. Cole, M. Noel, T. Lander, and T. Durham, "New telephone speech corpora at CSLU," *Proccedings of European Conference on Speech Communication and TEchnology*, vol. 1, pp. 821–824, 1995.

[5] S. Ikbal, *Nonlinear Feature Transformations for Noise Robust Speech Recognition*, Ph.D. thesis, Ecole Polytechnique Federal de Lausanne, 2004.

[6] H. Hermansky and S. Sivadas, "On the Use of Task Independent Training Data in Tandem Feature Extraction," *Proceedings of ICASSP*, vol. 1, 2004.

[7] P. Niyogi and M. M. Sondhi, "Detecting Stop Consonants in Continuous Speech," *The Journal of the Acoustic Society of America*, vol. 111, no. 2, pp. 1063–76, 2002.

[8] G. Aradilla, J. Vepa, and H. Bourlard, "Using Posterior-Based Features in Template Matching for Speech Recognition," *Proceedings of ICSLP*, June 2006.

STUDY OF MAXIMUM A POSTERIORI SPEAKER ADAPTATION FOR AUTOMATIC SPEECH RECOGNITION OF PATHOLOGICAL SPEECH

Oscar Saz, Carlos Vaquero, Eduardo Lleida *

Communications Technology Group (GTC)
I3A, University of Zaragoza
{oskarsaz,cvaquero,lleida}@unizar.es

José Manuel Marcos, César Canalís

Colegio Público de Educación
Especial "Alborada", Zaragoza
cpeealborada@gmail.com

ABSTRACT

This paper shows the results achieved by the Maximum A Posteriori (MAP) speaker adaptation method in a task of isolated word Automatic Speech Recognition (ASR) system for 6 young speakers with different types of speech pathologies. In this work, the influence of two important variables in speech recognition and speaker adaptation are studied. On one hand, the performance of the ASR system depending on the acustic unit used by the system (word level unit and context-dependent unit) is analized. On the other hand, how the size of the training set influences the results when using MAP adaptation is studied. The results of this work with the first subset of the Alborada-GTC corpus show that context-dependent units are the most reliable acoustic unit to use, while the use of 2 utterances of the vocabulary recorded in the corpus is enough for getting the best results in speaker adaptation.

1. INTRODUCTION

In the last years, the performance of Automatic Speech Recognition (ASR) systems has improved to very high levels of word accuracy when being used in controlled conditions (noise-free, known acoustic environment and collaborative user). But the performance decays sharply when the system is out of these conditions. This is the case when a user who is suffering a pathology in his/her speech is confronted to an ASR system. Some studies about how the speech is affected when the speaker is suffering any kind of pathology have been made [1]. These studies have pointed mayor variations in the time and frequency domain of the speech as a cause of the failure of the ASR systems when the user suffers any speech pathology. But, patients of any kind of speech pathology are a very likely group of people to use any kind of assistive technology based on speech technologies.

Speaker adaptation has been widely used to introduce ASR systems in our daily life. Several algorithms for speaker adaptation have been developed, Maximum Likelihood (ML) algorithms [2], Maximum A Posteriori (MAP) algorithms [3] and Maximum Likelihood Linear Regression (MLLR) algorithms [4]. MAP adaptation is a good and reliable method for speaker adaptation when sparse data from the speaker is available.

When dealing with speech pathologies, speaker adaptation is the first and basic step to take, but the patients of these pathologies usually find difficult and exhausting to talk in a continous way for a long time. This difficulty to record great amounts of speech for these users forces us to use a quick and reliable algorithm for speaker adaptation like MAP. However, studies on the use of MAP are necessary. The first question to answer is which is the best topology for the acoustic modelling, as it has been studied that the usual topologies may not be the best ones when dealing with pathological speech [5]. The second question to answer is which is the quantity of speech necessary to make an accurate adaptation to the speech of the speaker.

This paper is organized as follows. In Section 2, the Alborada-GTC corpus is introduced. Section 3 is a review of the basis of Maximum A Posteriori (MAP) speaker adaptation. In Section 4 the set of experiments is explained, while the results on MAP adaptation are shown in section 5. Finally, in section 6, the conclusions to this work are given.

2. THE ALBORADA-GTC CORPUS

The acquisition of the Alborada-GTC Corpus is the result of the joint work of the Communications Technology Group of the University of Zaragoza and the teachers and staff of the Public School for Special Education “Alborada”. This work, initial result of further collaborations, was carried away during the first half of year 2006 and finished with the development of the application “Vocaliza”, application for computer-aided speech therapy.

At the end of the process of development of “Vocaliza”, several recordings of children in the school were made, until the completion of the first part of the Alborada-GTC corpus, that it is introduced in this work.

2.1. Speaker composition of the corpus

This first part of the corpus contains recordings from six speakers, whose pathology and situation is explained next:

*This work has been supported by the national project TIN 2005-08660-C04-01

- Speaker BE: Girl, in the range of 10-12 years old, she suffers often and continous dyslalias altogether with sporadic guttural speech. Her speech may be hard to understand for the unused listener.
- Speaker EN: Boy, in the range of 19-21 years old, he suffers slight dyslalias and distorts the pronunciation of some phonemes.
- Speaker ES: Girl, in the range of 10-12 years old, she suffers severe dyslalias and a very weak speech, which does not allow her to properly vocalize. When she is speaking in a continous way, the unused listener might not understand some of the words.
- Speaker JO: Boy, in the range of 16-18 years old, he suffers slight dyslalias, his speech is understable although his speaking rate is very high and utters a high number of disfluencies.
- Speaker RB: Girl, in the range of 13-15 years old, she suffers severe dyslalias and a certain degree of dysphasia. She has problems concerning all levels of language.
- Speaker RG: Girl, in the range of 16-18 years old, she suffers very remarkable dyslalias and she has a non-standard Spanish accent. Her continous speech is hard to understand.

2.2. Phonological composition of the corpus

The set of words chosen for the recordings is the Induced Phonological Register [6]. These 57 words, arranged in terms of their difficulty is a very well-known set of words for the speech therapists in Spain. The set of words contains a rich selection of all the Spanish phonemes in several situations of phonemes boundaries and neighborhood relations.

2.3. Recording parameters

The recordings were made in the classrooms of the "Alborada" School, under the supervision of at least one of the members of the GTC and the surveillance of at least one of the staff of the school. The recordings were made in series of the whole Induce Phonological Register. The tiredness of the children during the process forced to a constant change of the speaker until every one of the 6 speakers recorded 4 series of the Register, which was consider enough speech material for the preliminary work exposed in this paper.

The corpus was recorded with a close-talk wireless microphone (AKG C444L). The use of a close-talk microphone gave us a noise-free recordings, where the average Signal to Noise Ratio is 23.29 dB. The use of a wireless microphone is to avoid that the children were attached to the recording system, so they could feel more comfortable during the recording sessions. Signals were recorded with a 16 kHz sampling frequency and a depth of 16 bits.

3. MAP ADAPTATION

The MAP adaptation [3], though not being the only method available for speaker adaptation, is a well known and reliable method to estimate speaker dependent HMM. This method is based on Bayes theory, and its main feature is the fact that it enables to take advantage of prior information in the training process, thus reducing the data needed to obtain a good speaker dependent acoustic model.

The MAP estimation can be seen as a Bayesian estimation: given a set of n speech feature vectors $X = (x_1, ..., x_n)$, if θ is the parameter vector to be estimated from X, with probability densisty function (pdf) given by $f(X|\theta)$, and g is the prior pdf of θ, it is possible to estimate θ_{MAP} as:

$$\theta_{MAP} = \arg\max_{\theta} g(\theta|X) = \arg\max_{\theta} f(X|\theta)g(\theta) \quad (1)$$

If θ is not known, it can be assumed that $g(\theta)$ gives no information, and (1) then reduces to the Maximum Likelihood (ML) expression.

This estimation method can be applied to estimate multivariate gaussian density functions as well as HMM vector parameters, so it makes possible to estimate speaker dependent or speaker independent acoustic models. To obtain a speaker dependent acoustic model, MAP adaptation usually takes as input a speaker independent model and some utterances from the speaker. The prior information provided by the speaker independent model enables MAP adaptation to obtain good speaker dependent acoustic models using sparse training data. Moreover, even if there is enough training data to use ML estimation, it has been proved that MAP adapted models obtain the same ASR performance in terms of WER as ML adapted models.

4. EXPERIMENTS

For these experiments, a 37 MFCC (Mel Frequency Cepstral Coefficient) parametrization was made, using 12 static parameters, 12 delta parameters, 12 delta-delta parameters and the delta-log-energy. The signals were windowed with a Hamming window of 25 ms. length, with an overlap of 15 ms. Two different acoustic models were used for the baseline, a set of 822 context-dependent units, trained with the whole 16108 utterances of the noise-free training subset of the Spanish SpeechDat-Car [7], and a word model with 57 units, generated by linking the context-dependent units to form whole words. Both models also included two units to model silence and inter-word silence.

Speaker	Word units	Context-dependent units
BE	42.18%	40.79%
EN	34.65%	33.77%
ES	31.14%	31.14%
JO	12.72%	11.40%
RB	58.77%	59.65%
RG	33.33%	32.89%
Average	35.47%	34.94%

Table 1. *Baseline WER results.*

4.1. Baseline per speaker

All the baseline results for the 6 speakers are shown in the Table 1. An average Word Error Rate (WER) of 35% for both models is obtained. Speaker RB clearly exceeds the WER of the rest of speakers, while speaker JO has a much lower WER than the rest.

5. RESULTS ON SPEAKER ADAPTATION

To compute the results of speaker adaptation, a set of 28 different adapted models were estimated for every speaker: 14 word unit based models, and 14 context-dependent unit based models. Each 14 models set was estimated as follows:

- Obtaining 4 different models trained with only one of the series recorded for every speaker. The testing will be made over the other 3 series not used for training.
- Obtaining 6 different models trained with the 6 possible combinations of the 4 series recorded for every speaker, taken 2 by 2. The testing will be made over the other 2 series not used for training.
- Obtaining 4 different models trained with the 4 possible combinations of the 4 series recorded series recorded for every speaker, taken 3 by 3. The testing will be made over the other serie not used for training.

This way, the independence between training and testing data is kept; and the results for the MAP adaptation can be obtained considering training with 1,2 or 3 series as the average of the results of all the models calculated with 1,2 or 3 series.

5.1. Word units

The results obtained considering word unit based models are shown in Table 2. These results are shown for every one of the six speakers of the Alborada-GTC corpus.

The results show that in the best situation, there is an improvement of the 53.82% of reduction of the Word Error Rate. The speaker who gets the highest improvement is BE with a 69.73% reduction of the WER, while the speaker with the poorer performance of the speaker adaptation is JO with a 21.84% reduction of the WER. All the speakers obtain a WER in the gap of 10% to 20%, except the speaker RB, who suffers the most severe pathologies, whose WER is 32.02%.

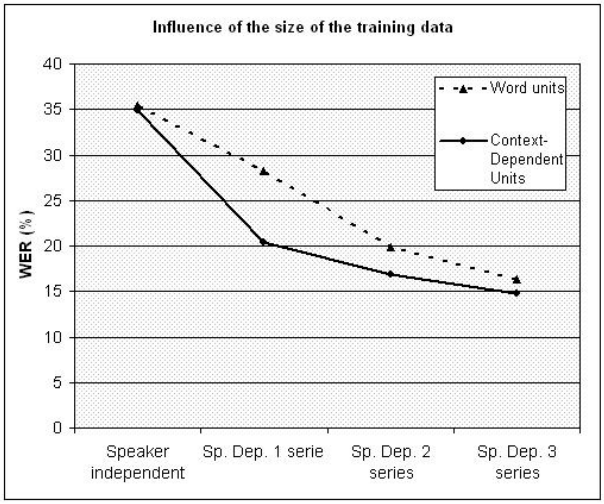


Figure 1. *Influence of the size of the training data.*

5.2. Context-dependent units

The results obtained considering the set of context-dependent units are shown in Table 3.These results are shown for every one of the six speakers of the Alborada-GTC corpus.

The results show that in the best situation, there is an improvement of the 57.73% of reduction of the Word Error Rate, a 4% more reduction than the reduction achived with word units. The speaker who gets the highest improvement is again BE with a 81.72% reduction of the WER, while the speaker with the poorer performance of the speaker adaptation is RB with a 25.73% reduction of the WER. All the speakers obtain a WER lower than 10%, except the speakers RG (12.72%) and RB (44.30%).

Of special interest is the case of the speaker RG, the speaker with the most severe and distinctive pathology all over the set of speakers of the Alborada-GTC corpus. This speaker achieves better results with word models, when the set of training data is big enough (2 or 3 series), than with the context-dependent units: 32.02% of WER with word units versus 44.30% of WER with context-dependent units for the case of the maximum size of training data.

5.3. Influence of the size of the training data

The evaluation of the influence of the size of the training data on the performance of the MAP algorithm for the speaker adaptation is made through the evaluation of the average results over all the speakers for the three possible situations of training (using 1, 2 or 3 series for training). These results are shown on the Figure 1.

The Figure clearly shows that word models are more sensitive to the size of the training data, as the decrease of the WER with this kind of acoustic units is slower than with the context-dependent units. But, with the maximum amount of data available (3 series), the word units obtain only less than 2% of more WER than the context-dependent units (16.38% versus 14.77%).

Speaker	Sp. independent	Sp. Dep. 1 serie	Sp. Dep. 2 series	Sp. Dep. 3 series
BE	42.18%	22.66%	13.60%	12.75%
EN	34.65%	23.69%	19.01%	14.48%
ES	31.14%	27.49%	18.57%	13.16%
JO	12.72%	11.97%	9.94%	10.09%
RB	58.77%	52.05%	36.99%	32.02%
RG	33.33%	31.43%	21.49%	15.79%
Average	35.47%	28.21%	19.93%	16.38%

Table 2. *Speaker adaptation WER results for word units.*

Speaker	Sp. independent	Sp. Dep. 1 serie	Sp. Dep. 2 series	Sp. Dep. 3 series
BE	40.79%	12.58%	9.51%	7.46%
EN	33.77%	17.98%	11.99%	9.65%
ES	31.14%	15.35%	12.57%	8.77%
JO	11.40%	9.79%	7.31%	5.70%
RB	59.65%	47.22%	45.47%	44.30%
RG	32.89%	19.74%	14.48%	12.72%
Average	34.94%	20.44%	16.89%	14.77%

Table 3. *Speaker adaptation WER results for context-dependent units.*

6. CONCLUSIONS

The results exposed in this work show the convenience of using speaker adaptation methods as a way to improve the performance of ASR systems for users with speech pathologies. The results show that MAP adaptation algorithm is as good as to reach a 60% improvement in the reduction of the WER. However, further work should be carried out to make possible a comparison in this task with other speaker adaptation algorithms, like MLLR, that have also shown good performance in several tasks.

As a conclusion to this work about the use of MAP adaptation for pathological speech it is shown that context-dependent units achieve better results than word units. The better performance of the context-dependent units is due to their higher precision to represent the borders between phonemes. A exception to this is the speaker RB, by far the one who is achieving worst results, for whom word units are the best adapted unit. Due to the special characterictics of her speech (dyslalia plus dysphasia), it should be studied if word units are the best units for her kind of speech.

About the size of the training set, good results are obtained when using 2 utterances of every word in the vocabulary (for a total of 114 words), while there is not a much major gain when using 3 utterances of every word. Word units show to be the most sensitive units to the lack of training data, as they achieve much worst results than context-dependent units with only 1 utterance of every word to train; while this difference is reduced when using a greater amount of data for training. This is because the context-dependent units may appear in different words, so they have more data for train with only one serie of the vocabulary of the corpus

7. BIBLIOGRAPHY

[1] Saz O., Miguel A., Lleida E., Ortega A., Buera L., “Study of Time and Frequency Variabilities in Pathological Speech and Error Reduction Methods for Automatic Speech Recognition”, In Proceedings of the 9th International Conference on Spoken Language Processing (ICSLP), Pittsburgh, 2006.

[2] Dempster A. P., Laird N. M., Rubin, D. B., “Maximum likelihood from incomplete data via the EM algorithm”, Journal of the Royal Statistical Society, Series B, vol. 39(1),pp. 1–38, 1977.

[3] Gauvain J.L., Lee C.-H., “ Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains” IEEE Transactions on Speech and Audio Processing, vol.2 (2), pp. 291–298, April 1994.

[4] Legetter C.J., Woodland P.C., “Maximum likelihood linear regression for speaker adaptation of the parameters of continuous density hidden markov models”, Computer Speech and Language, vol. 9, pp. 171–185, 1995.

[5] Deller J.R., Hsu D., Ferrier L.J., “On the use of Hidden Markov Modelling for recognition of dysarthric speech”, Computer Methods and Programs in Biomedicine, 35:125–139, 1991.

[6] Monfort M., Juárez-Sánchez A., “Registro Fonológico Inducido (Trajetas Gráficas)”, Ed. Cepe, Madrid, 1989.

[7] Moreno A., Nogueira A., Sesma A., “Speechdatcar: Spanish” Technical Report SpeechDat.

Author Index

Álvarez, A. 337

Abad, A. 271
Acero, Alex 37
Aizpuru, O. 297
Alías, Francesc 183, 219, 237
Alabau, Vicente 197
Alonso, E. 297
Alonso, Jesús B. 203
Amaral, R. 123
Anguita, J. 327

Banchs, Rafael E. 55
Banga, Eduardo R. 111, 167
Barra, R. 117, 191
Barros, Maria João 177
Bauman, Timo 231
Benítez, Carmen 25, 31
Bordel, Germán 349, 373
Bourlard, Hervé 389
Braga, Daniela 171, 369
Buera, Luis 7, 297

Córdoba, R. 45
Caballero, Mónica 225, 363
Callejas, Zoraida 105, 135
Campillo, Francisco 111, 167
Canalís, César 321, 395
Cardeñoso-Payo, Valentín 149
Carmona, José L. 249
Casacuberta, Francisco 49, 65, 71, 93, 99
Casar, Marta 225, 243
Caseiro, D. 123
Castro, María José 59, 141
Claudi, J. 183, 237
Coelho, Luís 171, 369
Colás, José 255
Crego, Josep M. 55
Cubel, Elsa 315
Curto, Natalia 77

D'Haro, L. F. 45, 117, 191
Díaz, David 39

Eliodoro, L. 93
Erro, Daniel 155
España, Salvador 141
Esquerra, Ignasi 161

Félix, V. 93
Falé, Isabel 265
Faria, Isabel Hub 265
Farrus, M. 327
Faundez-Zanuy, Marcos 261
Fernández, F. 45, 117, 191
Fernández, R. 337
Fernández, Rubén 343
Ferrés, Daniel 309
Ferreiros, J. 191
Ferrer, Miguel A. 203
Fombella, Carlos 343
Fonollosa, José A. R 83, 225, 243
Formiga, Lluís 183, 219

Gómez, Ángel M. 231, 249
Gómez, P. 337
Górriz, J. M. 31
Gallardo, Ascensión 383
García, Ana I. 389
García, Fernando 145
García, Luz 25, 31
García-Varea, Ismael 65
Garmendia, J. 297
Giménez, A. 93
Godino, J. I. 337
González, Jesús 93
González, Jorge 93
González, María-Teresa 71
González-Domínguez, Javier 355
González-Ferreras, César 149
González-Rodríguez, Joaquín 355
Gonzalvo, Xavier 183
Gorbe, Jorge 141
Grau, Sergio 145
Griol, David 129
Guijarrubia, Víctor 49, 297, 379

Hansen, John H.L. 189
Haya, Pablo 135
Hernáez, Inmaculada 13, 333
Hernández, Álvaro 39
Hernández, Luis A. 39, 343
Hernández-Pérez, E. 209
Hernando, Javier 271, 309, 327
Huerta, A. 117
Hurtado, Lluís F. 89, 129, 145

Ibáñez, E. 117
Iglesias, Gonzalo 111
Iriondo, Ignasi 183

Justo, Raquel 89

Khalilov, Maxim 83

López, Beatriz 39
López, Eduardo 39, 343
López-Cózar, Ramón 105, 135

López-Moreno, Ignacio 355
Lagarda, A. L. 93, 277
Lambert, Patrik 55
León, José de 203
Lleida, Eduardo 7, 291, 297, 321, 395
Lorenzo-García, F. D. 209
Luengo, Iker 13, 333
Luque, Jordi 309, 327

Méndez, Francisco 111
Macías-Guarasa, J. 45, 191, 291
Macho, Dušan 1, 19, 327
Malkin, Robert 1
María, Fernando Díaz de 213
Marcos, José Manuel 321, 395
Mariño, José B. 55, 285, 309, 363
Marqués, F. 327
Martín-González, S. I. 209
Martínez, C. 327
Martínez-Hinarejos, Carlos 197, 277, 303
Mazaira, L. M. 337
Meinedo, H. 123
Miguel, Antonio 7, 297
Miralles, Pere 183
Mishra, Taniya 167
Montero, J. M. 191
Montero-Asenjo, Alberto 355
Montoro, Germán 135
Morales, Juan A. 231
Moreno, Antonio 77
Moreno, Asunción 155
Morros, R. 327

Nadeu, Climent 1, 19, 271
Navarro, J. R. 93, 277
Navarro-Mesa, J. L. 209
Navas, Eva 13, 333
Neto, J. 123
Nogueiras, Albino 225, 363

Omologo, Maurizio 1
Ortega, Alfonso 7, 297
Ortiz-Martínez, Daniel 65

Pérez, Alicia 49
Pérez, José L. 231, 249
Pérez-Cruz, Fernando 213
Peñagarikano, Mikel 349, 373
Peinado, Antonio M. 231, 249
Peláez, Carmen 389
Pereiro, Yago 383
Peris, P. 93
Pieraccini, Roberto 283
Plaza, J. C. 45

Quintana-Morales, P. J. 209

Ramírez, Javier 25, 31
Ramos-Castro, Daniel 355
Ravelo-García, A.G. 209
Rodellar, V. 337
Rodríguez, Horacio 309
Rodríguez, Leandro 111
Rodríguez, Luis 277
Rodríguez, Luis Javier 349, 373
Rose, Richard 361
Rubio, Antonio J. 25

Sánchez, Jon 13, 333
Sánchez, Victoria 249
Sainz, Iñaki 13, 333
San-Segundo, R. 45, 117
Sanchís, Alberto 315
Sanchís, Emilio 129, 145
Sanchis, Germán 99
Santen, Jan van 167
Saratxaga, Ibon 13, 333
Satué-Villar, Antonio 261
Saz, Óscar 7, 297, 321, 395
Scharenborg, Odette 383
Segarra, Encarna 129, 145, 291
Segura, C. 271
Segura, José C. 25, 31
Solera, Rubén 213

Tamarit, Vicent 303
Tejedor, Javier 255
Temko, Andrey 1, 19
Torre, Ángel de la 25, 31
Torre, Doroteo 39, 77, 343, 355
Torre, Raúl de la 77
Torres, M. Inés 49, 89, 291, 297, 379
Tortajada, Salvador 59
Toselli, Alejandro H. 315
Trancoso, I. 123
Travieso, Carlos M. 203

Uría, A. 297
Uribe, Juan Pedro 349

Vaquero, Carlos 321, 395
Varona, Amparo 297
Vilaplana, V. 327

Wan, Vicent 383
Weiss, Christian 177

Zamalloa, Maider 349, 373
Zamora, Francisco 141
Zieger, Christian 1