UAM
UNIVERSIDAD AUTONOMA
DE MADRID

Contributions to Keyword Spotting and Spoken Term Detection For Information Retrieval in Audio Mining

by

Javier Tejedor Noguerales

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Escuela Politécnica Superior
Departamento de Ingeniería Informática
Universidad Autónoma de Madrid

Thesis advisor
Dr. José Colás Pasamontes

March 2009

TITLE: Contributions to Keyword Spotting and Spoken Term Detection For Information Retrieval in Audio Mining

AUTHOR: Javier Tejedor Noguerales

ADVISOR: Dr. José Colás Pasamontes

The committee for the defense of the thesis mentioned above is composed by:

PRESIDENT: Dr. Joaquín González Rodríguez

MEMBERS: Dr. Javier Macías Guarasa

Dra. Ascensión Gallardo Antolín

Dr. Simon King

SECRETARY: Dr. Doroteo Torre Toledano

Madrid, 14th April 2009

Dr. Doroteo Torre Toledano

Abstract

Nowadays, the number of real-world applications which get benefit from speech recognition techniques continues growing dramatically. Within such applications, Speech Information Retrieval is a very important activity in the world. Speech-based techniques such as continuous speech recognition have been widely used to develop such applications, commonly by means of Large Vocabulary Continuous Speech Recognition (LVCSR) systems. However, LVCSR systems in isolation are not well sited to deal with the search in the audio content due to this main reason: In Speech Information Retrieval, the common set of words used to access to the relevant information stored in huge audio repositories often includes proper names, acronyms, foreign words, which do not usually appear in the vocabulary of the LVCSR systems (i.e., they are Out-Of-Vocabulary (OOV) words). It causes that new approaches must be used to access to such information. Keyword Spotting and Spoken Term Detection (STD) are two approaches that try to solve the OOV problem within Speech Information Retrieval systems. This thesis concerns the development of new methods and solutions to be applied within the Keyword Spotting and STD framework.

For Keyword Spotting, we have followed a two-level based strategy. The first level makes use of a standard Hidden Markov Model (HMM)-based keyword spotting process, from which several filler models have been explored: phones, phonemes, broad classes and a single filler model. The second one presents four different types of confidence measures with the goal of improving the rates achieved by the first level in isolation. Two of them make use of an additional phone speech recognition, while the two others employ an additional isolated word speech recognition. Both the phone- and the word-based speech recognition compute the parameters used in the *Decision stage* according to each confidence measure to accept or reject each keyword proposed by the first level. Experimental results have shown that the confidence measure which makes use of a phone-based speech recognition and computes a modified Levenshtein distance from the sequence of phones according to the time intervals of the keywords proposed by the first level achieves the best system performance, with a reduction of about 43% relative in the *False Acceptance Rate (FAR)*, which causes a slight reduction of about 1% relative in the *Recognition Accuracy (RA)* on the Spanish ALBAYZIN database.

For STD, we have presented a comparison between phone- and grapheme-based acoustic units for Spanish language. Experimental results have shown that by using only the acoustic information in the way of the HMMs which represent both sets of units for a Spanish STD system, grapheme-based acoustic units outperform phone-based ones. In addition to this, the combination of the output of each system from each kind of acoustic unit to form the final output was shown to outperform each system in isolation. Two novel applications of the MultiLayer Perceptron (MLP)-based techniques and decision tree-based techniques have been also presented as follows: A posterior probability computed by means of the MLP training, along with the language model, were used to estimate the confidence score for each occurrence proposed by the STD system. It was shown that such approach improved in about 44% relative the performance achieved with standard HMM-based techniques on the Spanish ALBAYZIN database. The decision tree-based approach was applied over an English meetings domain and consisted of the classification of the occurrences proposed by the STD system in *hit* or *False Alarm (FA)* by means of the decision tree and the rejection of those classified as FA. To build the decision tree, several prosodic and lexical features have been used as input features. It has been shown that such approach achieves a slight better STD system performance than the absence of it of about 5% relative.

Resumen

En la actualidad, el número de aplicaciones que usan las técnicas basadas en reconocimiento de voz crece de forma imparable. Dentro de tales aplicaciones, la extracción de información en voz es una actividad de reconocida importancia. Las técnicas basadas en voz, como los reconocedores de habla continua, han sido ampliamente usadas para desarrollar tales aplicaciones, por medio de los reconocedores de habla continua de gran vocabulario. Sin embargo, los sistemas de reconocimiento de habla continua de gran vocabulario por sí mismos no son suficientes a la hora de realizar búsquedas en el contenido de audio por la siguiente razón: en la extracción de información en voz, el conjunto de palabras que se suele usar para acceder a la información almacenada en grandes repositorios de audio incluye nombres propios, acrónimos, extranjerismos, que no suelen aparecer en el vocabulario de los reconocedores de habla continua de gran vocabulario (es decir, son palabras de fuera del vocabulario). Esto obliga a buscar y desarrollar nuevas técnicas que permitan acceder a dicha información: el "Reconocimiento de Palabras Clave" y la "Detección de Términos Hablados". Estas dos técnicas intentan solucionar el problema causado por las palabras fuera de vocabulario dentro de los sistemas de extracción de información en voz. Esta tesis está enfocada al desarrollo de nuevos métodos y soluciones que son aplicados para el "Reconocimiento de Palabras Clave" y la "Detección de Términos Hablados".

Para el "Reconocimiento de Palabras Clave", hemos seguido una estrategia basada en 2 niveles. El primer nivel hace uso de un estándar "Reconocimiento de Palabras Clave" basado en Modelos Ocultos de Markov, en el cual varios modelos de relleno han sido explorados: alófonos, fonemas, clases amplias y un modelo genérico. El segundo nivel presenta 4 medidas de confianza diferentes con el objetivo de mejorar el resultado que ofrece el primer nivel. Dos de ellas hacen uso de un proceso adicional de reconocimiento de voz basado en alófonos, mientras que las otras dos emplean un reconocedor de palabras aisladas. Tanto el reconocedor de alófonos como el de palabras aisladas, calculan los parámetros necesarios para que la medida de confianza decida la aceptación o el rechazo de cada palabra clave propuesta por el primer nivel. Los experimentos han demostrado que la medida de confianza que hace uso de un reconocimiento de voz basado en alófonos, y calcula una distancia de Levenshtein modificada a partir de la secuencia de alófonos reconocida correspondiente a los intervalos temporales de las palabras clave propuestas por el primer nivel, logra los mejores resultados, con una reducción relativa de la tasa de falsas aceptaciones de un 43% y una mínima reducción relativa del 1% en la tasa de palabras clave detectadas de forma correcta sobre la base de datos española ALBAYZIN.

Para la "Detección de Términos Hablados", hemos presentado una comparación entre las unidades acústicas basadas en alófono y en grafema para el español. Los experimentos han demostrado que, usando únicamente la información contenida en los Modelos Ocultos de Markov que representan a cada unidad acústica para el sistema de "Detección de Términos Hablados" en español, las unidades acústicas basadas en grafema mejoran a las basadas en alófono. Además, la combinación de cada sistema a partir de cada conjunto de unidades acústicas para presentar la salida final del sistema se ha demostrado que mejora a cada sistema por separado. Dos nuevas aplicaciones de las técnicas basadas en perceptrones multicapa y árboles de decisión han sido también presentadas de la siguiente forma: la probabilidad a posteriori calculada a partir del entrenamiento de un perceptrón multi-capa, junto con el modelo de lenguaje, fueron usados para calcular la puntuación (confianza) de cada palabra clave propuesta por el sistema de "Detección de Términos Hablados". Se ha demostrado que esta técnica mejoró en un 44% relativo el resultado obtenido con las técnicas basadas en Modelos Ocultos de Markov sobre la base de datos española ALBAYZIN. La técnica basada en árboles de decisión fue aplicada sobre un dominio de reuniones en inglés y consistía en la clasificación de las palabras clave propuestas por el sistema de "Detección de Términos Hablados" en acierto o falsa aceptación usando el árbol de decisión y el rechazo de las clasificadas como falsa aceptación. En la construcción del árbol de decisión, características léxicas y prosódicas han sido usadas como características de entrada. Se ha demostrado que dicha técnica logra unos mínimos mejores resultados comparado con la ausencia de la misma de alrededor de un 5% relativo.

Acknowledgements

First of all, I would like to thank to my thesis advisor, José Colás Pasamontes, for his help and support during this thesis work. José has always given to me valuable comments during this work and has also guided me at every stage of my PhD studies. Thanks also to allow me to belong to the HCTLab group in University Autónoma of Madrid, where this thesis has been mainly developed.

I would also like to thank to the people belonging to the HCTLab group during these years, especially to Dani, Nico and Víctor with whom I have shared most of the moments and have helped me in my understanding of many aspects necessary for this thesis work. Thanks to Fernando, Javier, Ricardo and Sergio as well as Ana, Angel, Edu and Guillermo. They provided me with the perfect environment to achieve my goals.

A very special thank for the people belonging to the CSTR group in the University of Edinburgh, with whom I have shared one year, and gave me the chance of sharing and developing new ideas and work for this thesis. Special thanks to Dong, Joe, Simon and Steve for their valuable comments about my work there. Thanks also to Bela, Blaise, Gregor, Heriberto, Iván, John, Leonardo, Matthew, Michael, Mónica, Partha, Paula, Roberto, Sebastian, Volker, Yolanda and so on, who made me feel at home.

Finally, I would like to thank to my family, my mum María Isabel, my dad Dalmiro and my sister Laura and my closest friends (Ana, Carlos, César, Cristina H., Cristina S., Fernando, Héctor, Javi A., Javi S.G., Javi S.P., Jorge, Laura, Mar, Pedro and Sara) who have been always there for me. Without them, this work had not been possible.

And thanks to all people who, as much as they possibly could, have supported and guided me through this thesis.

Contents

Abstract v

Resumen vii

Acknowledgements ix

List of Figures xvii

List of Tables xix

1 Introduction 1
- 1.1 Introduction . . . 1
- 1.2 Thesis goals . . . 5
- 1.3 Outline . . . 6

2 Experimental background 9
- 2.1 Introduction . . . 9
- 2.2 The HTK recognition system . . . 9
  - 2.2.1 Feature extraction . . . 11
    - 2.2.1.1 Tandem Features . . . 11
  - 2.2.2 Acoustic modelling . . . 13
    - 2.2.2.1 Phoneme-based units for Spanish . . . 14
    - 2.2.2.2 Grapheme-based units for Spanish . . . 14
    - 2.2.2.3 Phoneme-based units for English . . . 15
  - 2.2.3 Lexical Modelling . . . 15
  - 2.2.4 Language Modelling . . . 16
  - 2.2.5 Recognition . . . 16
- 2.3 The speech corpora . . . 17
  - 2.3.1 ALBAYZIN database . . . 17
  - 2.3.2 Meetings speech data . . . 18
    - 2.3.2.1 Meetings 2005 data . . . 18

2.3.2.2 2004 Spring NIST Rich Transcription (RT-04S) Development data 18
2.3.2.3 2004 Spring NIST Rich Transcription (RT-04S) Evaluation data 19
2.3.2.4 2005 Spring NIST Rich Transcription (RT-05S) Evaluation data 19
2.3.3 Summary 20

3 Prior research and State-of-the-art 21
3.1 Introduction 21
3.2 Approaches to model OOV words in continuous speech recognition . 21
3.2.1 Vocabulary Optimization 22
3.2.2 Confidence Scoring 22
3.2.3 Multi-stage Subword Recognition 22
3.2.4 Filler Models 23
3.3 Approaches to access the OOV words in continuous speech recognition: Keyword Spotting and Spoken Term Detection 24
3.3.1 Keyword Spotting 24
3.3.1.1 HMM-based approaches for Keyword Spotting . . . 24
3.3.1.2 Keyword Spotting without filler models 26
3.3.1.3 Hybrid Keyword Spotting 26
3.3.1.4 General approaches for Keyword Spotting 27
3.3.1.5 Filler and Language Models for HMM-based Keyword Spotting 28
3.3.2 Confidence Measures for Keyword Spotting 32
3.3.2.1 Confidence measures for HMM-based Keyword Spotting 32
3.3.2.2 Confidence measures for LVCSR-based Keyword Spotting 34
3.3.2.3 Confidence measures for Keyword Spotting without filler models 35
3.3.3 Spoken Term Detection 36
3.3.3.1 1-Best word-to-phoneme conversion plus lexical access for STD 36
3.3.3.2 Phone-Ngrams and Lattice based STD 37
3.3.3.3 A phone-state based matrix for STD 40
3.3.3.4 Probabilistic pronunciation model for STD 41
3.4 Summary 43

4 Contributions to Keyword Spotting 45
4.1 Introduction 45
4.2 First level: HMM-based keyword spotting process 46
4.3 Second level: Confidence Measures 48
4.3.1 Exact Match 49
4.3.1.1 Motivation 49

4.3.1.2 System architecture 50
4.3.2 Likelihood 51
4.3.2.1 Motivation 51
4.3.2.2 System architecture 51
4.3.3 Heuristic Rules 52
4.3.3.1 Motivation 52
4.3.3.2 System architecture 53
4.3.4 Lexical Access 55
4.3.4.1 Motivation 55
4.3.4.2 System architecture 55
4.4 Experimental setup 58
4.4.1 Feature extraction 58
4.4.2 Acoustic modelling 59
4.4.3 Language modelling 59
4.4.4 Lexicon 59
4.4.5 System tuning 59
4.5 Results and discussion 61
4.5.1 CI results 61
4.5.2 CD results 65
4.5.3 Keyword-length based analysis 67
4.6 Conclusions 69
4.7 Summary 70

5 Contributions to Spoken Term Detection 73

5.1 Introduction 73
5.2 STD approach 74
5.3 Confidence scoring 74
5.3.1 Motivation and prior research 74
5.3.2 HMM-based confidence scoring 75
5.3.3 MLP-based confidence scoring 78
5.3.3.1 Direct LM integration 79
5.3.3.2 Baum-Welch LM integration 80
5.3.4 Experimental setup 82
5.3.4.1 Feature extraction 82
5.3.4.2 Acoustic Modelling 82
5.3.4.3 Language Modelling 82
5.3.4.4 System tuning 82
5.3.5 Results and discussion 83
5.3.6 Conclusions 84
5.4 Decision tree-based confidence measure 85
5.4.1 Preliminary work 85
5.4.2 Decision tree-based approach 88
5.4.2.1 Motivation and prior research 88
5.4.2.2 Decision tree and selected features 89

5.4.2.3 Experimental setup 93
5.4.2.4 Results and discussion 93
5.4.2.5 Conclusions 93
5.5 Summary 95

6 Phone- versus Grapheme-based systems for Keyword Spotting and Spoken Term Detection in Spanish 97
6.1 Introduction 97
6.2 Motivation and prior research 98
6.3 Comparison of phone- and grapheme-based units on MFCC-based feature extraction 99
6.3.1 Architecture *1-Best*: 1-Best sub-word unit decoding + lexical access 100
6.3.2 Arhitecture *Lattice*: sub-word unit lattice + exact word matching 101
6.3.3 Architecture *Hybrid*: hybrid word + sub-word system 103
6.3.4 Experimental setup 103
6.3.4.1 Feature extraction 103
6.3.4.2 Acoustic Modelling 103
6.3.4.3 Language Modelling 104
6.3.4.4 System tuning 104
6.3.5 Results and discussion 105
6.3.5.1 Recognition accuracy 105
6.3.5.2 Spoken term detection and keyword spotting results 106
6.3.6 Conclusions 109
6.4 The influence of the feature extraction for phone- and grapheme-based units for Spanish STD 112
6.4.1 Results and discussion 114
6.4.1.1 Recognition accuracy 114
6.4.1.2 STD results 115
6.4.2 Conclusions 119
6.5 Combination of phone- and grapheme-based units for Spanish STD 120
6.5.1 Results and discussion 121
6.5.2 Analysis of the complementary behaviour of phone and grapheme acoustic units 126
6.5.3 Conclusions 127
6.6 Summary 129

7 Summary, contributions and future work 131
7.1 Summary 131
7.1.1 Prior research and State-of-the-art 131
7.1.2 Keyword Spotting approaches 132
7.1.3 Spoken Term Detection approaches 133
7.1.4 Phone- and grapheme-based units for Spanish Keyword Spotting and STD 133

7.1.5 Experimental results 133
7.2 Contributions and publications 136
7.3 Future work 138

A Inventory of Spanish phones 139

B Inventory of English phonemes 141

C Broad class model as filler model 143

D Keywords from the Albayzin geographical domain 145

E Evaluation metrics 147

F Conclusiones 151

Bibliography 155

List of Figures

1.1 The framework for speech recognition in an Audio Mining system. . 3
1.2 The framework of a keyword spotting system. 3
1.3 The framework of an STD system. 4

2.1 The framework of an ASR system. 10
2.2 The MLP network for the Tandem Features extraction. 13

3.1 The probabilistic pronunciation model based on HMMs. 42

4.1 The framework of the keyword spotting system. 46
4.2 The recognition network in the HMM-based keyword spotting process. 48
4.3 The system architecture for the *Exact Match* confidence measure in the keyword spotting system. 50
4.4 The system architecture for the *Likelihood* confidence measure in the keyword spotting system. 52
4.5 The system architecture for the *Heuristic Rules* confidence measure in the keyword spotting system. 54
4.6 The system architecture for the *Lexical Access* confidence measure in the keyword spotting system. 58

5.1 The MLP network for the MLP-based confidence scoring. 78
5.2 The phone-dependency representation for the posterior confidence computation. 79
5.3 The DET curves for the confidence scoring approaches for Spanish STD. 85
5.4 The decision tree for CER-based tuning. 92
5.5 The decision tree for the *biased towards hits* tuning. 92
5.6 The DET curves for the development set without the decision tree-based approach and with it when tuning for biased towards hits and for CER. 94
5.7 The DET curves for the test set without the decision tree-based approach and with it when tuning for biased towards hits and for CER. 95

6.1 The Architecture *1-Best*. 100
6.2 Example of a lattice of graphemes. 102
6.3 The Architecture *Lattice*. 102

6.4 The DET curves for *MFCC* feature vectors for the CI and CD acoustic models for the Architecture *1-Best*. 109

6.5 The DET curves for *MFCC* feature vectors for the CI and CD acoustic models for the Architecture *Lattice*. 110

6.6 The DET curves for *MFCC* feature vectors for the CI and CD acoustic models for the Architecture *Hybrid*. 110

6.7 The DET curves for the CD acoustic models for *MFCC* and *MFCC+Tandem Features* configurations for the Architecture *1-Best*. 118

6.8 The DET curves for the CD acoustic models for *MFCC* and *MFCC+Tandem Features* configurations for the Architecture *Lattice*. 119

6.9 The DET curves for *MFCC* feature vectors for the three confidence scoring computations for the CD phone and grapheme acoustic models and their combination. 124

6.10 The DET curves for *MFCC+Tandem Features* configuration for the three confidence scoring computations for the CD phone and grapheme acoustic models and their combination. 125

6.11 The FOM value for the phone- and grapheme-based systems for each keyword in the *SKG* (Short-length keywords group). 128

6.12 The FOM value for the phone- and grapheme-based systems for each keyword in the *MKG* (Medium-length keywords group). . . . 128

6.13 The FOM value for the phone- and grapheme-based systems for each keyword in the *LKG* (Long-length keywords group). 129

7.1 The comparison of the keyword spotting and STD approaches for OOV words in information retrieval. 136

List of Tables

2.1 Text resources to train the LM for the phoneme-based system on the English meetings domain. 17

2.2 Specification of the sub-corpora for the ALBAYZIN database. 18

2.3 Division by recording institute for the meetings data. 20

3.1 Approaches for OOV modelling capable of Keyword Spotting and STD. 24

3.2 Methods applied to HMM-based Keyword Spotting. 25

4.1 Results in terms of RA/FAR for CI acoustic models for the confidence measures in the development set, i.e., the ***geographic training set***. 62

4.2 Results in terms of RA/FAR for CI acoustic models for the confidence measures in the test set, i.e., the ***geographic test set***. . . . 62

4.3 Results in terms of FOM for CI acoustic models for the confidence measures in the development set, i.e., the ***geographic training set***. 63

4.4 Results in terms of FOM for CI acoustic models for the confidence measures in the test set, i.e., the ***geographic test set***. 63

4.5 Results in terms of FOM, RA/FAR and the reduction of FAR compared with No CM for CD acoustic models for the confidence measures in the development set, i.e., the ***geographic training set***. . 66

4.6 Results in terms of FOM, RA/FAR and the reduction of FAR compared with No CM for CD acoustic models for the confidence measures in the test set, i.e., the ***geographic test set***. 66

4.7 Results in terms of FOM for the *SKG*, *MKG* and *LKG* in the test set, i.e., the ***geographic test set***. 68

4.8 Results in terms of *RA/FAR* for the SKG, MKG and LKG in the test set, i.e., the ***geographic test set*** 68

5.1 Results in terms of FOM and ATWV for the CD phone models for the confidence scoring approaches for the ***geographic training set***. 83

5.2 Results in terms of FOM and ATWV for the CD phone models for the confidence scoring approaches for the ***geographic test set***. . . 83

5.3 Results in terms of FOM and ATWV for the RT-04 evaluation data. 87

5.4 Results in terms of FOM and ATWV for the RT-05 evaluation data. 87

5.5 Results in terms of ATWV without the decision tree-based approach and with it when it is tuned for biased towards hits and for CER for the ***geographic training set***. 94
5.6 Results in terms of ATWV without the decision tree-based approach and with it when it is tuned for biased towards hits and for CER for the ***geographic test set***. 94

6.1 Phone and grapheme accuracy for *MFCC* feature vectors for both CI and CD acoustic models on the ***phonetic test set***. 106
6.2 Phone and grapheme accuracy for *MFCC* feature vectors for both CI and CD acoustic models on the ***geographic training set***. . . . 106
6.3 Phone and grapheme accuracy for *MFCC* feature vectors for both CI and CD acoustic models on the ***geographic test set***. 106
6.4 Evaluation of the *HMM-based keyword spotting* process of Architecture *Hybrid* in isolation for the ***geographic training set***. 106
6.5 Evaluation of the *HMM-based keyword spoting* process of Architecture *Hybrid* in isolation for the ***geographic test set***. 107
6.6 Results in terms of FOM and OCC for the three architectures for CI and CD phone and grapheme acoustic models for the ***geographic training set***. 107
6.7 Results in terms of FOM and OCC for the three architectures for CI and CD phone and grapheme acoustic models for the ***geographic test set***. 108
6.8 Phone and grapheme accuracy for *MFCC* and *MFCC+Tandem Features* configurations for CD phone and grapheme acoustic models on the ***phonetic test set***. 114
6.9 Phone and grapheme accuracy for *MFCC* and *MFCC+Tandem Features* configurations for CD phone and grapheme acoustic models on the ***geographic training set***. 115
6.10 Phone and grapheme accuracy for *MFCC* and *MFCC+Tandem Features* configurations for CD phone and grapheme acoustic models on the ***geographic test set***. 115
6.11 Results in terms of FOM and OCC for *MFCC* and *MFCC+Tandem Features* configurations for the *1-Best* architecture for CD phone and grapheme acoustic models for the ***geographic training set***. . 115
6.12 Results in terms of FOM and OCC for *MFCC* and *MFCC+Tandem Features* configurations for the *1-Best* architecture for CD phone and grapheme acoustic models for the ***geographic test set***. 116
6.13 Results in terms of FOM and OCC for *MFCC* and *MFCC+Tandem Features* configurations for the *Lattice* architecture for CD phone and grapheme acoustic models for the ***geographic training set***. . 116
6.14 Results in terms of FOM and OCC for *MFCC* and *MFCC+Tandem Features* configurations for the *Lattice* architecture for CD phone and grapheme acoustic models for the ***geographic test set***. 117

6.15 Results in terms of FOM and ATWV for *MFCC* feature vectors with the three confidence scoring computations for CD phone and grapheme acoustic models and their combination for the ***geographic training set***. 121

6.16 Results in terms of FOM and ATWV for *MFCC* feature vectors with the three confidence scoring computations for CD phone and grapheme acoustic models and their combination for the ***geographic test set***. 122

6.17 Results in terms of FOM and ATWV for *MFCC+Tandem Features* configuration for the three confidence scoring computations for CD phone and grapheme acoustic models and their combination for the ***geographic training set***. 122

6.18 Results in terms of FOM and ATWV for *MFCC+Tandem Features* configuration for the three confidence scoring computations for CD phone and grapheme acoustic models and their combination for the ***geographic test set***. 123

6.19 Results in terms of FOM for the *SKG*, *MKG* and *LKG* on the ***geographic test set***. 127

A.1 Inventory of Spanish phones along with an example of each. 140

B.1 Inventory of English phonemes along with an example of each. . . . 142

C.1 Eight broad class models to build the broad class filler model. . . . 143

D.1 List of keywords from the ALBAYZIN geographical domain along with the number of occurrences (Occ.) in development (dev) and test sets for each and the total (TOTAL) number of them in both sets. 146

Chapter 1

Introduction

1.1 Introduction

Speech recognition is the process of converting an input acoustic signal to a set of words. Applications such as voice dialing, call routing, domotic control, etc are involved within this technology. In recent years, the ever-increasing volume of audio data available online through the World Wide Web (WWW) means that automatic methods for indexing and search are becoming essential. This last issue is related to the search over audio-based content (Audio Mining). Broadcast News (BN), Conversational Telephone Speech (CTS) and meetings are domains over which these methods are widely applied to access to relevant information.

Garofolo claimed that the information extraction in large audio repositories was a solved problem by means of the well trained and tuned Large Vocabulary Continuous Speech Recognition (LVCSR) systems [1]. In this way, a search within the output of such systems for the words required in the application would be enough. Therefore, it was the main approach used in the past [2, 3, 4, 5, 6, 7]. However, these systems suffer from three main drawbacks:

- Vocabulary coverage: the words that do not appear in the vocabulary of the LVCSR system cannot be recognized, which leads to important errors in the final output of the system.
- Computational cost: They are very expensive computationally.

- Amount of training data: To train robust, accurate and useful LVCSR systems, a lot of data are required.

The two last drawbacks have been solved in the last years by means of the huge amount of data available for many domains and the fast and the big storage capacity of the machines used for this task. But the first one (Vocabulary coverage) still remains in these days. It is accepted that if all of the words to search are in the vocabulary of the LVCSR system, this approach achieves the best results. However, the words to search in the audio content are usually proper names, named entities, acronyms, etc which are often Out-Of-Vocabulary (OOV) words because they do not appear in the vocabulary of the LVCSR systems. This is considered to be the main drawback of the LVCSR-based techniques applied over the search in audio content. No matter how large the vocabulary is, speech recognizers always have to deal with OOV words. In his study, Logan [3] claimed that over 10% of the user queries contain OOV words in an information retrieval system. And OOV words also tend to cause an error in neighbouring words, which degradates dramatically the performance of the information retrieval system. When an OOV word appears, a recognizer may hypothesize similar word or words from the vocabulary instead, causing the neighbouring words to be mis-recognized. Therefore, to deal with the OOV words, Keyword Spotting and, more recently, Spoken Term Detection (STD) approximations, are used with the objective of addressing and solving the OOV problem. Figure 1.1 represents the common framework related to speech recognition in an information retrieval system, i.e., Audio Mining system.

Keyword Spotting deals with the identification of a reduced set of keywords in utterances. The most common approach is the Hidden Markov Model (HMM)-based keyword spotting, where the keywords are represented by their phonetic transcription whereas the non-keywords are represented by means of filler (garbage) models (fillers), which can vary from sub-word units, such as phones, syllables, graphemes, etc to whole words. However, it mantains the drawback that a single change in the vocabulary of the application makes necessary run the recognition process again, which is the most time-consuming task in the whole process. Confidence measures are also used to increase the performance of the final system. The common framework of a keyword spotting system is depicted in Figure 1.2.

On the other hand, STD, defined by the National Institute of Standards and Technology (NIST) in 2006 [8] makes the recognition process be independent of

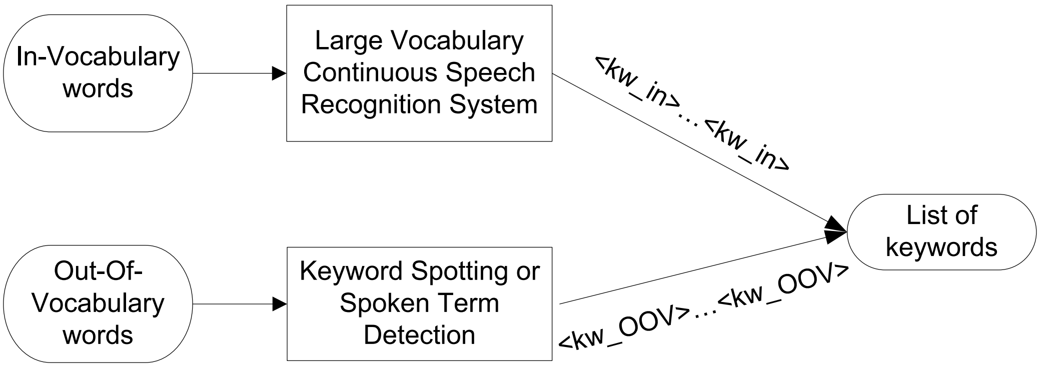

Figure 1.1: The framework for speech recognition in an Audio Mining system. <kw_in> denotes a keyword which appears in the vocabulary of the LVCSR system. <kw_OOV> denotes a keyword which does not appear in the vocabulary of the LVCSR system.

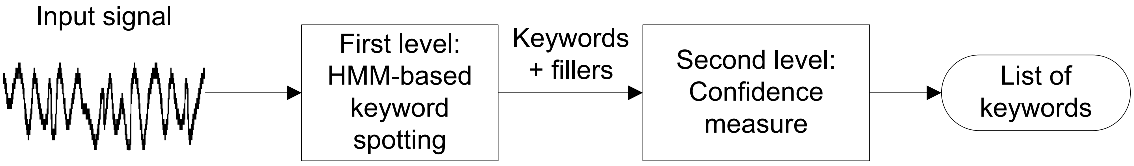
Input signal
First level: HMM-based keyword spotting
Keywords + fillers
Second level: Confidence measure
List of keywords

Figure 1.2: The framework of a keyword spotting system.

the list of keywords to search due to it is unknown during the recognition process, contrary to Keyword Spotting where the list of keywords is known beforehand. It causes that approaches to STD must be addressed in two different steps as it is shown in Figure 1.3. The first step indexes the audio by means of sub-word units (typically phonemes) commonly in the way of a lattice or 1-Best. The second step employs a *Keywords search algorithm* to build the keywords from this index. Although NIST considers optional the use of the audio in a step different from the first one, as the speed is considered as important as the accuracy in STD, we have only made use of the audio in the first step. In addition, state-of-the-art STD systems also follow such decision. It is well-known that this kind of approaches achieve poorer results than the keyword spotting ones, due to the absence of the word-level lexical information during the decoding process, but it allows to search any keyword within huge audio repositories faster. Nowadays, a combination of an LVCSR system to search for in-vocabulary (INV) words with such approaches for

OOV words is the main technique applied to build efficient information retrieval systems.

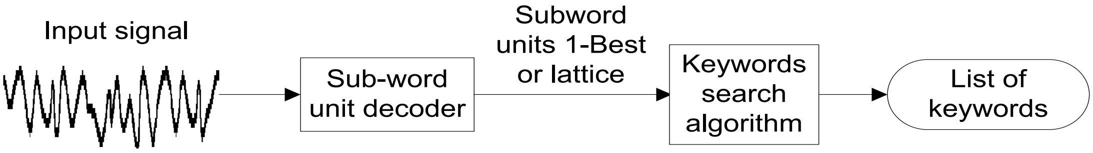

FIGURE 1.3: The framework of an STD system.

In this thesis we face the problem of solving the OOV problem of a traditional LVCSR system. For such purpose, we propose novel approaches to deal with Keyword Spotting and STD tasks. The best one presented for Keyword Spotting combines two different levels. The first level integrates the common and widely used HMM-based keyword spotting method. The second level uses a phone-based approach (as confidence measure) with the intention of reducing the errors produced in the first level. This approach exploits all of the benefits of two decoding processes in parallel, the knowledge of the errors appearing during the phone-based decoding and the knowledge of the list of keywords to search. A comparison between this new approach and another based on the widely used likelihood computed from the Viterbi algorithm during the decoding process is presented.

Our contribution to STD focuses on the comparison and combination of two different acoustic models (phoneme-based units and grapheme-based units) for the Spanish language. Due to the relationship between its set of phonemes and graphemes is very close and the near Word Error Rate (WER) of the grapheme- and phoneme-based LVCSR systems in Spanish, we hypothesize that grapheme-based units can achieve at least similar performance to the phoneme-based units for a vocabulary independent STD system for Spanish. The same comparison has been done for the best architecture for Keyword Spotting. On the other hand, the performance of the STD systems greatly relies on the confidence score computed for each occurrence. Therefore, a novel approach based on a Multi-Layer Perceptron (MLP) is presented. It is trained from the sequence of feature vectors extracted from the input acoustic signal and computes a posterior probability for each sub-word unit in the keyword according to the feature vectors. Such posterior probability and the language model (LM) component are used to calculate the confidence score for each occurrence. This approach has been evaluated over the Spanish language for both phoneme- and grapheme-based units.

We also present a novel approach for the STD task over meetings domain. It is based on Classification And Regression Trees (CARTs), widely used for classification tasks and already used for other tasks such as hot spot in meetings, sentence boundary detection and finding disfluencies in conversational speech. Since the occurrences hyphothesized in the STD system are classified as hit (if the putative occurrence is correct) or false alarm (FA) (if the putative occurrence is incorrect), we make use of this approach to classify the putative list of occurrences as one of these two classes with the final purpose of rejecting those classified as FA.

1.2 Thesis goals

There are three main approaches that can be used to retrieve a set of keywords from the audio content. The first approach is to use an LVCSR system and after that a simple search of the keywords within the output (1-Best or lattice) of the recognition step. This output is a string composed of the words defined in the lexicon or vocabulary of such system. A posterior linear search within this output (1-Best or lattice of words) of the relevant keywords presents the final output of the system. The second approach is to use an HMM-based keyword spotting process which outputs a string composed of the keywords plus the filler models to absorb the non-keyword intervals in the audio content. Such keywords constitute the final output of the system, while filler models are rejected. Finally, the third approach can be used for the STD task explained before, where the output of the recognition step is composed of sub-word units without making use of the audio in subsequent steps prior to build the relevant keywords from such units.

The main goal of this thesis is to develop several approaches to deal with the Keyword Spotting and STD tasks to solve the OOV problem of an LVCSR system. Therefore, the approach consisting of an LVCSR system is beyond the scope of this thesis. Read speech data and meetings domain have been used in the experimental task. In achieving our main goal, several issues are addressed:

- Which approach does achieve the best results for such tasks?

- Which confidence measure does improve the final performance for each architecture?
- Which kind of sub-word units should be used in the acoustic modelling for these tasks?
- Which type of units should be used as filler models in the keyword spotting task?
- Which type of features should be used to represent the input acoustic signal?

In this thesis, we make the following contributions to the Keyword Spotting and STD tasks:

- The development of new approaches and confidence measures on the keyword spotting task.
- The comparison of two types of sub-word units (phones and graphemes) for Spanish language for both tasks and the combination of them for Spanish STD.
- The comparison of different techniques to calculate the final confidence score for the keywords in the STD task.
- The development of a new confidence measure to deal with the STD task based on CARTs.

1.3 Outline

The remainder of this thesis is organized into seven chapters. Following is a brief description of each:

- Chapter 2: Experimental background

  This chapter provides the basic background needed throughout the thesis. It briefly describes the HTK tool used for the recognition system, the acoustic models and the feature extraction as well as the lexical models and LMs used throughout the tesis. It also provides an overview of the corpora used in the thesis.

- Chapter 3: Prior research and State-of-the-art

  This chapter describes the main approaches used to deal with the Keyword Spotting and STD tasks to solve the OOV problem so far. Previously, it also presents the approaches used to model the OOV words in LVCSR systems and analyzes which of them can be applied on Keyword Spotting and STD.

- Chapter 4: Contributions to Keyword Spotting

  This chapter describes the architectures developed for the keyword spotting task along with the confidence measures proposed for this task. We also describe the filler models used in this task. The second half of the chapter presents the experiments over the geographical domain in the Spanish ALBAYZIN database for these architectures. We compare the results achieved by the architectures using the *Recognition Accuracy (RA)* and the *False Acceptance Rate (FAR)* along with the standard Figure-Of-Merit (FOM) metric. We also make an analysis of the performance of the different architectures presented according to the length (number of phones) of the keywords.

- Chapter 5: Contributions to Spoken Term Detection

  This chapter describes the approaches developed for the STD task. It presents the experiments over the same geographical domain as the keyword spotting task and over data recorded for the meetings domain. We evaluate the results using the FOM metric and the Actual Term Weighted Value (ATWV) metric, and present the Detection Error Tradeoff (DET) curve according to the ATWV metric as well.

- Chapter 6: Phone- versus Grapheme-based systems for Keyword Spotting and Spoken Term Detection in Spanish

  This chapter describes the different behaviour of the phone and grapheme acoustic models when are applied over Keyword Spotting and STD in Spanish. One architecture for Keyword Spotting and two architectures for STD have been used to evaluate both types of acoustic models. It compares the results presented over the geographical domain in the Spanish ALBAYZIN database for both types of acoustic models using two different feature extraction processes. The FOM metric, the Occurrence-weighted value (OCC) metric and the DET curve according to the ATWV metric have been used to evaluate them. We also present the powerful combination of phone and grapheme acoustic models in the STD task. For such combination, we present the results in terms of FOM and ATWV metrics and plot the DET curve got from the ATWV metric.

- Chapter 7: Summary, contributions and future work

  This chapter presents a summary of the innovations presented in the thesis. It also reports the main contributions presented in this thesis work. It concludes with a discussion on possible future work.

Chapter 2

Experimental background

2.1 Introduction

This chapter provides the general background directly relevant to the content of this thesis. In the first part, we review the main components of the HTK tool used throughout this thesis for the speech recognition work along with the data used for each. The second part includes a description of the corpora used in this thesis: the read speech Spanish Albayzin database and data recorded in English meetings (Meetings speech data) from several institutes: the International Computer Science Institute (ICSI) at Berkeley, the Interactive Systems Laboratories (ISL) at Carnegie Mellon University (CMU), the National Institute for Standards and Technology (NIST), the Linguistic Data Consortium (LDC), Virginia Polytechnic and State University (VT) and partners of the Augmented Multiparty Interaction (AMI) project.

2.2 The HTK recognition system

HTK [9] is a toolkit which is mainly designed for building HMM-based speech processing tools, in particular Automatic Speech Recognition (ASR) systems. It offers a set of tools to carry out the basic functions related to the speech recognition: The feature extraction, from which the input acoustic signal is transformed into a sequence of vectors (Mel-Frequency Cepstral Coefficient (MFCC), Perceptual Linear Predictive (PLP), Linear Predictive Coding Coefficient (LPCC), etc), used during the training of the acoustic models and the recognition (decoding) process. The training of the acoustic models which consists of the building of the HMMs by means of the Baum-Welch algorithm, from the vectors extracted previously. And finally the recognition process from which, by means of the Viterbi algorithm, the input signal is transformed into a string composed of the required recognized units (phonemes, words, etc). In addition to this, this tool also provides components from which LMs to be used during the recognition process can be built. The set of units to be recognized is defined in the lexicon of the ASR system and typically consists of phones, graphemes, syllables or words. Figure 2.1 represents the common framework of an ASR system. In the next sections a brief overview of these components is presented, along with the data used for each. Readers are referred to [9] to get a full knowledge of these standard components. This tool has been used throughout the thesis for the speech recognition work.

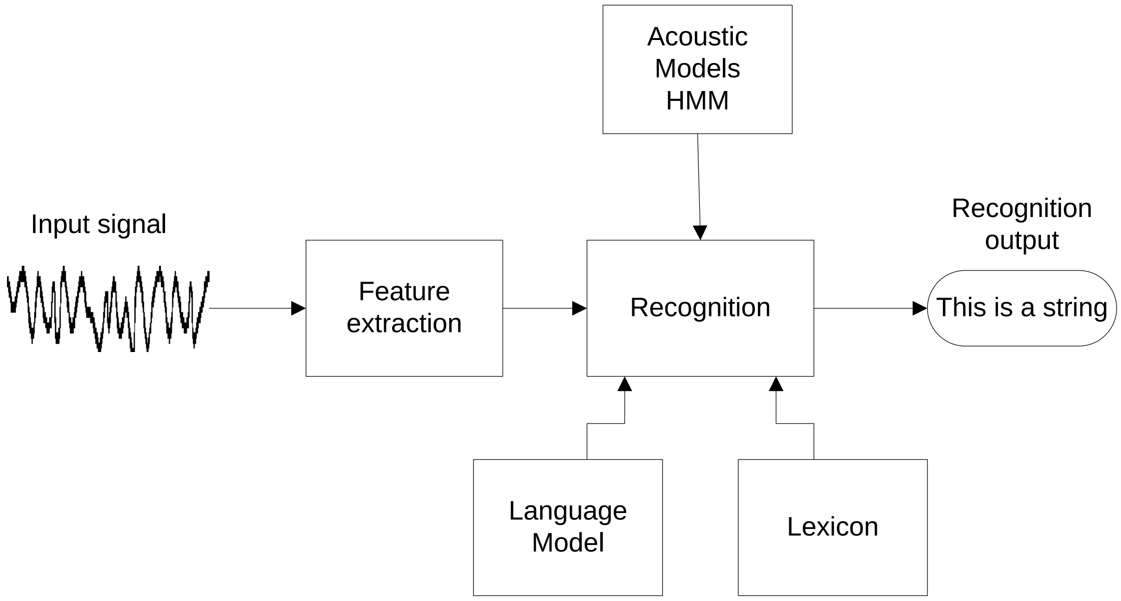

Figure 2.1: The basic steps in the framework of a standard ASR system.

Theoretically, the speech recognition problem is defined as follows: From a sequence of feature vectors used as observations which represent the acoustic signal $O = \{o_1, o_2, ..., o_n\}$, the goal is to find the best sequence of units $W = \{w_1, w_2, ..., w_m\}$ presented in the input acoustic signal. It can be expressed as follows:

$$W = argmax_w P(w|O) \tag{2.1}$$

This equation is easily transformed by means of Bayes' rule in this one:

$$W = argmax_w \frac{P(O|w)P(w)}{P(O)} \tag{2.2}$$

And assuming that the denominator in Equation 2.2 is constant, the solution to the speech recognition problem is equivalent to solve the following final formulation:

$$W = argmax_w P(O|w)P(w) \tag{2.3}$$

where $P(w)$ is computed from the LM and $P(O|w)$ is computed from the HMMs which represent the sequence of units defined in the lexicon.

2.2.1 Feature extraction

The aim of this step is to get a set of parameters from each frame of the input acoustic signal which represents the most relevant information of the signal. Two types of features have been used throughout the experiments of the thesis. The first one is the standard feature extraction based on the MFCCs which are widely used to represent the audio signal. The second one is the combination of these MFCCs with the more recently defined *Tandem Features*, built from the training of an MLP from the PLP coefficients or MFCCs of the audio signal, which outputs a posterior probability for each unit defined in the acoustic models according to each frame of the audio signal.

2.2.1.1 Tandem Features

Recently, new acoustic features have been proposed to replace or augment the standard MFCC, PLP or LPCC, which are commonly known as standard features. They are based on the training of an MLP whose input is the standard MFCC, PLP or LPCC features and whose output is the posterior probability of each class (unit in the acoustic models) given the features. In 2000, Hermansky et al. [10]

proposed the use of the MLP outputs as observations (in an identical way as the standard features) for a Gaussian Mixture Models (GMM)-HMM system. Zhu et al. [11] proposed to merge the MLP outputs with the standard features to get the observations used as input for the GMM-HMM system. And this work and other related works have also proved that the combination of both types of features (standard features and the outputs of the MLP) achieves better performance in ASR systems [12, 13, 14, 15, 16, 17, 18, 19] compared with using the standard features in isolation.

The training process followed to extract such tandem features and the combination with the standard MFCCs consists of the following steps:

1. The standard PLP coefficients are extracted from the input acoustic signal, in an identical fashion as the MFCCs are.

2. The MLP is trained (weights are computed) using the Quicknet software [20] from the PLP coefficients computed previously. This MLP contains 3 layers: the input layer, a single hidden layer and an output layer. The input layer is a window of 2W + 1 frames of acoustic features. $W = 4$ in our case, so a 9-frame input window is used. Each of these frames contains 39 PLP coefficients, so 351 units are used for the input layer. The number of units in the hidden layer, along with the number of epocs and the learning factor in the MLP training, are computed during the training of different MLPs varying these numbers to maximize the cross-validation accuracy. The output layer contains as many units as the number of acoustic models. The MLP was trained using a softmax output activation, which can be used to estimate the class posterior probabilities for a classification task. Figure 2.2 shows the structure of the MLP along with the 3 layers.

3. The posterior probability for each 9-frame input window for each of the units in the output layer is computed. Each posterior is considered to be a new coefficient in the future new feature vector composed of the MFCCs and the tandem features.

4. A global decorrelation by using a Karhunen-Loeve (KL) transform is applied over the posterior probabilities computed in the step 3. Therefore, there will be less coefficients in the tandem features vector after this step. This transform is applied according to the property of that such posterior probabilities contain one large value (corresponding to the current acoustic unit) whereas all of the other values are much smaller.

5. Merge the MFCCs and the posterior probabilities (i.e, coefficients) resulting from the step 4 (i.e., after the KL transform).

The step 2 is just required to be applied over the training data used to build the HMMs from the *MFCC+Tandem features* combination. The rest of data sets made use of such MLP and all of the steps except 2 are computed. For them, the posterior probabilites were computed from the set of weights calculated during the MLP training stage.

In addition to this, it is necessary to train the number of posterior probabilities (i.e., coefficients), that remains after the KL transform in the step 4, along with the matrix used in it. It has been done from the training data as well. Next, such matrix has been applied in the KL transform for the rest of data sets.

The steps followed to get the *MFCC+Tandem features* have been taken from other approaches proposed in the literature [10, 11].

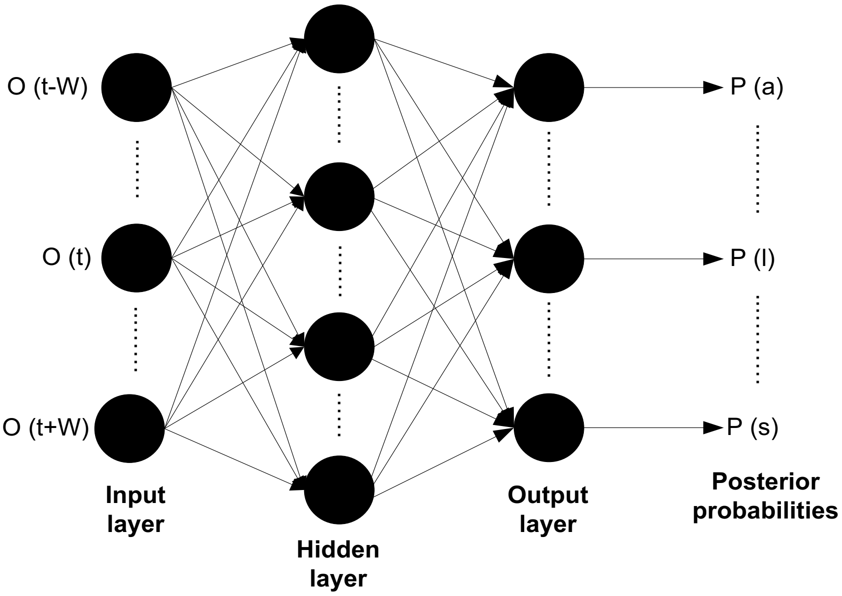

FIGURE 2.2: The MLP network for the Tandem Features extraction.

2.2.2 Acoustic modelling

HMMs have been used as acoustic models throughout the thesis. Both context-independent and context-dependent sub-word units have been built using the Baum-Welch algorithm based on the maximum likelihood criterion provided within the HTK tool [9]. At the beginning, all context-independent models were made to be equally and subsequent iterations of such algorithm estimated the final values for each acoustic model. When moving from context-independent to context-dependent models, HTK's standard decision tree method was used.

The acoustic models make use of two different types of sub-word units for the Keyword Spotting and STD tasks for Spanish and one for English. The first one consists of the set of allophones (phones) [21] in Spanish language. The second one is the set of graphemes [22] in Spanish language. The next two sections describe these two types of sub-word units in more detail. For the English experiments we have used the standard set of phonemes defined for English (*ftp://ftp.cs.cmu.edu/project/speech/dict/phoneset.0.6*).

2.2.2.1 Phoneme-based units for Spanish

The well-defined set of 47 allophones (phones) proposed by Quilis [21] was chosen for the phoneme-based systems throughout the thesis. This set differs from the standard set of phonemes for Spanish, composed commonly by 24 phonemes, due to it presents a slight better phoneme recognition accuracy (77.2% against 76.6%) for the 24-phoneme based speech recognition and it represents the different sounds in Spanish more accurately. Moreover, it allows us to make a different model for couple of words which only differ in the stressed vowel, e.g. "cuadro" (picture) and "cuadró" (the past tense of the verb to balance). In this case, more different words can be recognized, and therefore more complex systems can be built in using such set. In addition to this, previous works in Spanish speech recognition have shown its good performance on isolated and continuous speech recognition tasks [23, 24]. A full inventory of this set of allophones can be found in Appendix A, with an example of a word containing each.

2.2.2.2 Grapheme-based units for Spanish

Although there is a simple relationship between spelling and sound in Spanish, care must be taken in defining the inventory of graphemes [22]. We will use the term "grapheme" to mean a single unit, which is a sequence of one or more letters, to be used for acoustic modelling. This may not be precisely match the alphabet used for writing because we can expect better performance if we account for a small number of language-specific special cases.

The letter ”h” only affects the phonetic realisation when it appears in the combination ”ch”, as in ”chaqueta” (”jacket”) or ”Pancho” (a proper name). ”ch” is always pronounced [tʃ]. Therefore ”ch” is considered to be a grapheme (digrapheme in this case) and the letter ”h” can be removed everywhere else. The only exceptions are in loanwords, such as ”Sáhara” (borrowed from Arabic) or ”hall” (borrowed from English) where the ”h” is pronounced somewhere along a [h] - [χ] continuum, depending on the speaker. In this thesis, we have ignored the pronunciation of ”h” in loanwords, because the Spanish corpus used for experimentation contains no loanwords.

The combination ”ll” is pronounced [dʒ] or [y], depending on context, and so is also considered a grapheme (digrapheme in this case) because its pronunciation is not related to that of its constituent letters. ”ñ” is also considered a grapheme for the same reason (it is *not* an ”n” plus a ”˜”). It is always pronounced [ɲ].

There are therefore a total of 28 grapheme units in our systems: a, b, c, ch, d, e, f, g, i, j, k, l, ll, m, n, ñ, o, p, q, r, s, t, u, v, w, x, y and z.

There are, of course, other letter combinations that could be considered as single graphemes, such as ”rr”, but a balance must be struck between capturing these special cases of letter-to-sound relationships, and keeping the grapheme inventory size small for statistical modelling reasons.

2.2.2.3 Phoneme-based units for English

The standard set of phonemes in English taken from the CMU dictionary was used for the English STD system presented in this thesis. It is composed of 39 English phonemes, which are specified in Appendix B along with an example of a word containing each.

2.2.3 Lexical Modelling

For both Keyword Spotting and STD, the final objective is to detect a set of keywords presented in the input acoustic signal. For Keyword Spotting, where the list of keywords is known prior to the recognition process, the lexical modelling consists of those keywords plus filler models to deal with the non-keyword segments of the audio signal. To compare phone- and grapheme-based units as acoustic models, in the former case, the keywords are modeled by a sequence of phones, extracted for each keyword using a grapheme-to-sound module. In the latter case, the keywords are modeled by their ortographic form, according to the grapheme-based units explained in the Section 2.2.2.2. For STD, where the recognition is forced to be performed by sub-word units in all this thesis work, the lexical models were composed of phones and graphemes respectively for each kind of acoustic model for Spanish and they were composed of phonemes for English.

2.2.4 Language Modelling

It is very common in continuous speech recognition systems that the language modelling is defined by means of N-grams, commonly in the way of word bi-grams or tri-grams. However, for systems where the vocabulary of the application is very likely to change, such LMs should be recalculated each time a new keyword is added. For such reason, language modelling in Keyword Spotting and especially in STD (in which the recognition process followed in this thesis work is made by sub-word units) should differ from such word-based N-grams. In our work, we have defined an LM in the keyword spotting systems consisting of a uni-gram where the probability assigned to a keyword model and filler model is different and a bi-gram trained from sub-word units for Spanish and English STD systems. Table 2.1 lists the data used to build the LM for the English STD system.

2.2.5 Recognition

The aim of the recognition, also known as decoding process, is to find the best path through the labelled segmented network, with the lexicon (pronunciation models) and the LM serving as constraints. The Viterbi algorithm is used to perform the decoding process and to hypothesize the final (1-Best or lattice) output corresponding to the audio signal. The aim of this algorithm is to find the most likely sequence of units (words or sub-word units) corresponding to the acoustic feature vectors got in the feature extraction process. A whole description of this algorithm can be found in [25].

Corpus	millions of words
Switchboard/CHE	3.5
Fisher	10.5
Web (Switchboard)	163
Web (Fisher)	484
Web (Fisher topics)	156
BBC-THISL	33
HUB4-LM96	152
SDSR99-Newswire	39
ICSI/ISL/NIST/AMI	1.5
Web (ICSI)	128
Web (AMI)	100
Web (CHIL)	70
Total	1355.5

Table 2.1: Text resources to train the LM for the phoneme-based system on the English meetings domain.

2.3 The speech corpora

Two different sets of data were used in the experiments of the thesis. The Spanish Albayzin database and data recorded in an English meetings domain. The next two sections describe each set of data in more detail.

2.3.1 ALBAYZIN database

The Spanish Albayzin database [26] is a read speech database of about 10.2 hours which contains two separate sub-corpora: a phonetically rich component and a geographic corpus. Each of these is divided into training and test sets. Therefore, there exist four distinct, non-overlapping portions of the data as described by Table 2.2.

The geographic corpus was used throughout the experimental work in the Keyword Spotting and STD tasks for Spanish. It contains sentences and questions related to the Spanish geography such as: *"dime donde nace el río más corto que pasa por Barcelona"*, *"¿a qué altura se encuentra el pico más alto del sistema penibético?"* and *"todas las ciudades con población superior a un millón de habitantes"*.

	Phonetic corpus (orthographically transcribed and phonetically labelled)	Geographic corpus (orthographically transcribed)
Train set	NAME: ***phonetic training set*** CONTAINS: 4800 phonetically balanced sentences from 164 speakers: 3.3 hours.	NAME: ***geographic training set*** CONTAINS: 4400 sentences from 88 speakers: 3.3 hours.
Test set	NAME: ***phonetic test set*** CONTAINS: 2000 phonetically balanced sentences from 40 speakers: 1.6 hours.	NAME: ***geographic test set*** CONTAINS: 2400 sentences from 48 speakers: 2 hours.

TABLE 2.2: Specification of the sub-corpora for the ALBAYZIN database.

2.3.2 Meetings speech data

2.3.2.1 Meetings 2005 data

This set of data (meetings-05) includes speech collected and transcribed by ICSI at Berkeley, ISL at CMU, NIST and AMI partners. It includes 73 hours of speech from 30 meetings at ICSI [27], 13 hours of speech from 15 meetings at NIST, 10 hours from 18 meetings at ISL [28] and 16 hours from 35 meetings by AMI partners [29]. Totally, these data contain 104 hours of speech excluding the silence regions.

2.3.2.2 2004 Spring NIST Rich Transcription (RT-04S) Development data

The RT-04S Development data (RT-04Sdev) is the development set of data (containing meeting speech and reference transcripts) used in the RT-04S evaluation provided by NIST.

This set of data contains speech collected and/or transcribed by ICSI at Berkeley, ISL at CMU, NIST and LDC. It consists of 8 meetings which contain 1.4 hours of speech in total. Each meeting was recorded using lapel microphones, head-mounted microphones (IHMs) and at least one distant microphone, although in this thesis we have only used the speech recorded from the IHMs. Table 2.3 summarizes the portion of data recorded in each institute.

In using this set of data, we have partitioned the original files into short segments, each of which represents one utterance and have excluded all of the segments that just contain silence to complete the 1758 utterances.

More information about this database can be found in [30].

2.3.2.3 2004 Spring NIST Rich Transcription (RT-04S) Evaluation data

The RT-04S Evaluation data (RT-04Seval) contains the test material (both meeting speech and reference transcripts) used in the RT-04S evaluation provided by NIST.

This set of data contains speech collected and/or transcribed by ICSI at Berkeley, ISL at CMU, NIST and LDC. It consists of 8 meetings which contain 1.7 hours of speech in total. Each meeting was recorded using an IHM per person and at least one distant microphone, although in this thesis we have only used the speech recorded from the IHMs. Table 2.3 summarizes the portion of data recorded in each institute.

As for the RT-04Sdev data, we have partitioned the original files into short segments, each of which represents one utterance and have excluded all of the segments that just contain silence to complete the 2501 utterances.

More information about this database can be found in [30].

2.3.2.4 2005 Spring NIST Rich Transcription (RT-05S) Evaluation data

The RT-05S Evaluation data (RT-05Seval) contains the test material (both meeting speech and reference transcripts) used in the RT-05S evaluation provided by NIST.

This set of data contains speech collected and/or transcribed by ICSI at Berkeley, ISL at CMU, AMI partners and VT. It consists of 10 meetings which contain 2.1 hours. One IHM per person and several distant microphones were used in recording the meetings. As in the RT-04S Evaluation, we have only selected the data from the IHMs. Table 2.3 summarizes the portion of data recorded in each institute.

As before, the original audio files were divided into short segments, to complete the 3130 utterances used as evaluation data. Again, the silence segments were excluded.

More information of this database can be found in [31].

	utt/hrs			
	meetings-05	RT-04Sdev	RT-04Seval	RT-05Seval
ICSI	101136/66.7	509/0.35	604/0.42	596/0.44
NIST	11767/12.8	377/0.35	560/0.40	638/0.41
ISL	10476/8.9	366/0.32	694/0.45	749/0.46
LDC	–	506/0.38	643/0.40	–
AMI	13443/15.5	–	–	570/0.39
VT	–	–	–	577/0.35
TOTAL	136822/103.9	1758/1.4	2501/1.7	3130/2.1

Table 2.3: Division by recording institute for the meetings data. utt refers to the number of utterances recorded in each institute and hrs refers to the number of hours of speech recorded in each institute.

2.3.3 Summary

In this section we have reported a short background related to this thesis. In doing, we have reported a brief overview of the HTK tool used throughout the thesis for the speech recognition work. We have also presented the data used within each component of the tool for Keyword Spotting and STD. Finally, we have presented an overview of the corpora used for the experiments in this thesis.

Chapter 3

Prior research and State-of-the-art

3.1 Introduction

In this chapter we present an overview of the Keyword Spotting and STD techniques used for years in both tasks. First, we describe the techniques used to model the OOV words in continuous speech recognition. We later describe some related work developed for Keyword Spotting and STD (without making use of the audio in any step except the indexing one) to solve the OOV problem.

3.2 Approaches to model OOV words in continuous speech recognition

According to Bazzi [32], there are four common categories in which the approaches used to model the OOV words in LVCSR systems can be classified. Table 3.1 presents the approaches that can be used for Keyword Spotting (different from the LVCSR-based approach) and STD (without making use of the audio in any step except the indexing one) to solve the OOV problem.

3.2.1 Vocabulary Optimization

The first attempt is to reduce the OOV rate of the LVCSR systems as much as possible. It can be done in two different ways: On the one hand, it is possible to augment the vocabulary size for large vocabulary-independent recognizers. On the other hand, it is also possible to select those words belonging to a specific domain and to incorporate them to the final vocabulary of the application. In both cases, the OOV problem is not totally solved due to there will still be some words belonging to the domain that are not going to be selected, mainly proper names and foreign words. Another drawback is that increasing the vocabulary makes the recognition process slower and more expensive computationally. And sometimes it could also degrade the final performance due to more words are involved during the recognition process. In addition to this, the LM needs to be retrained as well.

No definitive improvement can be done to solve the OOV problem in the LVCSR systems so Keyword Spotting and STD approaches are used for it.

3.2.2 Confidence Scoring

The second strategy tries to predict if a recognized word is actually a substitution of an OOV word in the output of the LVCSR system. These types of approaches only try to recover from some errors occuring in the recognition process caused by the absence in the vocabulary of the actual word presented in the speech signal. Due to there is no possibility of retrieving such OOV words because they do not appear in the vocabulary of the LVCSR system, approaches in this category cannot be used for Keyword Spotting and STD tasks.

3.2.3 Multi-stage Subword Recognition

The third strategy is the multi-stage recognition approach. This strategy splits the whole process into two or more steps. In the first step, a sub-word unit decoding is performed to retrieve the most likely sequence of sub-word units (phones, syllables, etc), commonly in the way of a lattice, and to store them as an index. The second step uses that index to build the words according to the vocabulary of the application. It is well-known that this strategy performs worse than the LVCSR-based approach if we only focus on the INV words, due to the word-level lexical knowledge is not used during the decoding process. However, as Keyword Spotting deals with the OOV words, this strategy is well-sited to face it. In addition to this, since this strategy does not make any a priori knowledge of those OOV words during the recognition process, it is the only able to STD.

3.2.4 Filler Models

Filler models have been the most common approach to handle OOV words. And in fact, these models have been widely used in keyword spotting tasks for years, presenting better performance than the multi-stage subword recognition approach. Typically, a filler model acts as a generic word or a garbage model. Here, it is important to decide how this filler model is integrated into the LM component, how many filler models are used and which data are used to train them. The main difference between the filler model in OOV modelling and in Keyword Spotting is that in Keyword Spotting, its main purpose is to absorb the non-keyword part of speech, while in OOV modelling is used to detect the OOV words, which are possibly the most important in the utterance. The main drawback of the filler models is that they are highly unconstrained and they may absorb some parts of the speech corresponding to relevant keywords.

Therefore, this strategy can be used in Keyword Spotting, but not in STD, due to the list of keywords must be known when the recognition process is run. It must be noted that, as commented in Chapter 1, all of the approaches used for the STD task in this thesis work do not make use of the audio in any step except the recognition, i.e., indexing one, so the list of keywords cannot deal with the audio files directly. In this strategy, both the lexicon and the LM component are composed by the list of keywords plus the filler models.

	Keyword Spotting	Spoken Term Detection (*)
Vocabulary Optimization	NO	NO
Confidence Scoring	NO	NO
Multi-stage Subword Recognition	*YES*	*YES*
Filler Models	*YES*	NO

Table 3.1: Approaches for OOV modelling capable of Keyword Spotting and STD (*) (without making use of the audio in any step except the indexing one)

3.3 Approaches to access the OOV words in continuous speech recognition: Keyword Spotting and Spoken Term Detection

For years, Keyword Spotting and more recently STD, defined by NIST, have been widely used to access to the OOV words (i.e., to solve the OOV problem) of the LVCSR systems with the final objective of getting relevant information within the audio content. The following two sections describe the techniques and approaches developed for them.

3.3.1 Keyword Spotting

3.3.1.1 HMM-based approaches for Keyword Spotting

There are two basic methods adopted for Keyword Spotting for years, apart from the well-known search within the output (1-Best or lattice) of an LVCSR system, but impractical for the applications which deal with OOV words.

The most widely method used has been the HMM-based keyword spotting introduced by Rose and Paul [33] where filler models are used to absorb the non-keyword intervals of the speech and the keywords are built from their sequence of phones [34, 35, 36, 37, 38, 39, 40]. It outputs the sequence of keywords and filler models resulting from the HMM-based decoding process. The next sections describe the variety of filler models and confidence measures used in these works along with the results in more detail.

The other method proposes the use of a Finite State Grammar (FSG) [41] instead of the filler models. This method suffers from the substantial limitation of its inability of covering all of the possible words appearing in the speech data, i.e., the same limitation that the LVCSR systems. Such methods make use of some specific words (I, need, want, please, find, etc) instead of the filler models to deal with the non-keyword intervals. Guo et al. [41] showed that in case the test set contains data that the FSG can recognize (i.e., is composed by these specific words plus the keywords), this method gets a better performance than the filler model, due to a whole-word level information is used. However, when the test set is composed by data that are not defined in the FSG, such method presents a worse system performance. It means that this method is only valid when the domain over which the keyword spotting system is developed is very restricted and the words appearing in the speech data are very well defined, e.g. auto-attendant systems, and generally speaking, in systems where the user is only allowed to speak a well-defined set of words related to a very specific domain (flight booking, account bank management, etc). For their experiments, they selected 602 utterances which only contained a single name, 298 sentences that can be recognized by the FSG and 880 sentences that were not defined in the FSG.

An approach which combines both methods has been also investigated by Yining et al. [42], achieving a better result than the one based on filler models. Such approach generates a Mixed Grammar Model (MGM) where both filler models and those specific words (seven words in their work) are merged within the LM component prior to the decoding process. Experiments for such approach selected 498 utterances which only contained a single keyword and 272 utterances with only one keyword in each along with other non-relevant words. Table 3.2 summarizes the comparison of each method and the combination of both.

	Filler models	FSG	MGM
Performance	Fair	Excellent	Good
Robustness	Good	Poor	Good
Adapting a prior knowledge	No	Yes	Yes
Covering all the possibilities	Yes	No	Yes

Table 3.2: Methods applied to HMM-based Keyword Spotting. This table has been taken from Yining et al.[42]

3.3.1.2 Keyword Spotting without filler models

Modifications to the widely used Viterbi algorithm during the decoding process to avoid the use of filler or garbage models for Keyword Spotting have been also proposed. Silaghi and Bourlard [43] presented an iterating Viterbi decoding algorithm, in which instead of trying all of the possible start and end points for a putative keyword according to the input signal, their iterative algorithm finds for each time stamp t, the optimal path from the beginning to t and updates the confidence value from which compute if it has fallen below a threshold and decide that there is a keyword in such path. Such confidence value is calculated from the local posterior probability (output values assigned to each frame for each phoneme) from the training of an MLP used in a hybrid HMM/Artificial Neural Network (ANN) system [44]. They only considered the case of detecting one keyword per utterance, with non-keyword segments at the left and right of the putative keyword. They chose 100 keywords from the BREF database [45] and showed that the system performance is comparable to other alternative approaches.

3.3.1.3 Hybrid Keyword Spotting

Yu and Seide [46] presented a hybrid word/phoneme-based approach consisting of a *prior combination* and a *posterior combination*. The phoneme-based approach is a lattice-based word spotting where a phonetic word-fragment lattice is generated. It contains word fragments which may vary from syllables to whole words, along with phonemes. A posterior step to build the keywords from such fragments in the lattice is required. The word-based approach generates a word lattice over which find the set of relevant keywords. A trigram for the word-based approach and a bigram for the phoneme-based approach were used as LMs. They showed that the phoneme-based approach achieved only a little worse accuracy than the word-based approach for INV words and it still maintained similar performance for OOV words (which cannot be found by the word-based approach). The hybrid word/phoneme approach is composed of two different combinations. The *posterior combination* takes the keywords hypothesized by the word- and phoneme-based approaches separately and merged them into a single output. The new confidence score for each keyword merged is a linear combination of the two posterior probabilities, with a weight applied over each. The *prior combination* integrated both phonemic and graphemic versions of a word in a single decoding process (thus the dictionary contains both phonemic and graphemic representations of each word). In this case, both LMs and vocabularies are combined prior the decoding process. The easiest way to do such combination is based on utterance-level which is similar to posterior combination. Within it, transitions between words and phonemes are not allowed, so such combination is done by merging the word and phoneme lattices and by connecting the start and end times. The main drawback of such method is that it still needs two different recognition processes, each of these generating each set of lattices. For this reason, a *word-level prior combination* was presented, which combined a hybrid LM by doing a simple linear interpolation between the word- and the phoneme-level LMs. Experiments were performed over the LDC Voicemail Corpus [47]. To test the system over two different domains, they presented two versions for the word and hybrid approaches, one *in domain* and another *out of domain*. The first one uses the same corpus (although a non-overlapping set of data) for both training the LMs and testing the system, while the second used a different corpus (Switchboard) to build the word-based LM and to test the system (Voicemail). The list of keywords consisted of 2049 entries, with 620 OOV entries for the *in domain* scenario and 530 OOV entries for the *out of domain* one. They concluded that the phoneme-based approach achieved better performance than the word-based approach for OOV words in the *in domain* scenario and *out of domain* scenario for both OOV and INV words. They also showed that the *posterior combination* achieved better performance in all the cases (for both INV and OOV words and for the two scenarios) compared with each approach (phoneme-based or word-based) in isolation. And similar performance was achieved when using the *word-level prior combination* and the *posterior combination*, with a single recognition pass in the former and a double recognition pass in the latter.

3.3.1.4 General approaches for Keyword Spotting

Szoke et al. [39] presented several approaches for keyword spotting systems (KWS). Such approaches relate the well-known state-of-the-art for such task. In this way, they presented a LVCSR-based lattice keyword spotting system, where they search the keywords within the output (word lattice) of the LVCSR system. They also presented an acoustic KWS where keyword models are composed by phoneme models and the OOV words are absorbed by filler models represented by a loop of the same models. And finally they presented a third approach based on a prior phoneme-lattice based decoding and a posterior step which searchs for the exact sequence of phonemes and is able to handle substitutions and insertions in such lattice to hypothesize if a keyword was spotted or was not. For the experiments, they used the ICSI meetings database, selecting 17 keywords appearing more than 95 times each in the test set. All these words were INV, so it is expected that the LVCSR-based KWS system achieves the best performance. Context-dependent acoustic models and a trigram LM were used to generate the word lattices in the LVCSR system. A recognition system based on temporal patterns (TRAPs) and neural networks (NNs) [48] was used for both the acoustic and phoneme lattice-based approaches, due to its better performance over phoneme decoding than the traditional systems based on HMM/GMM. They even showed better performance with the TRAP-based acoustic KWS using monophones as acoustic models than using triphones with the HMM/GMM-based one. No LM was used for both acoustic KWS and phoneme-lattice based KWS. They conclude in their study that the LVCSR-based lattice KWS achieved the best rate (it must be noted that all of the keywords are INV), followed by the acoustic KWS from the TRAP-NNs system and finally the phoneme lattice KWS. However, due to no LM was used in the acoustic KWS and the rate (64.46 as FOM), is very near to the one achieved by the LVCSR-based KWS (66.95 as FOM), such result is very promising. The phoneme-lattice based approach, which achieved the worst rate (58.9 as FOM) should be use instead for a fast search over huge speech data, where the accuracy is as important as the speed.

3.3.1.5 Filler and Language Models for HMM-based Keyword Spotting

For years, the most common approach for Keyword Spotting is the HMM-based keyword spotting. It augments the keyword models with filler (garbage) models or fillers to deal with the non-keyword intervals of the speech.

Rose and Paul [33] proposed the use of the following types of filler models: (1) word models, where 80 non-keywords represented the filler models, (2) sub-word models where both context-independent and context-dependent phoneme models were used and (3) unsupervised clustering to form 128 single-state filler models as the cluster centroids of the Kmeans algorithm using a Mahalanobis distance. All these filler models, except the ones derived from the unsupervised clustering were trained from the non-keyword training data. Keyword models were formed from context-dependent phonemes trained on the whole training data. Both keyword and filler models, except the 128 filler models derived from the unsupervised clustering, were trained from clean speech. Those 128 fillers were trained from conversational speech. They used a parallel network of keywords and fillers as LM where the interword transition weight assigned to each keyword and to each filler model was chosen to achieve a desired tradeoff between misses and false alarms. As test set, they used conversational speech, selecting 20 keywords, with 353 keyword occurrences in this set. They showed that the use of context-dependent phoneme models as filler models achieved the best performance, followed by the context-independent phoneme models and the word models. Much worse performance achieved the 128 filler models from the unsupervised clustering. In all of these cases, the score for each putative keyword is computed by the likelihood given by the Viterbi-based decoder divided by the length of the keyword time interval. In their work, they showed that, for the context-independent phoneme filler models, the system performance is improved when the final confidence score is computed by substracting that keyword score minus the score computed by applying the same Viterbi-based decoder over the regions of speech corresponding to the keyword against a background network composed of the same filler models.

Manos and Zue [34] proposed several filler models such as context-independent phones and the clustering of the context-independent phones into broad phonetic classes (nasals, closures, stops, etc). They tried 18, 12 and 1 models in this latter configuration, which were built from the clustering of the context-independent phones. All of these filler models were trained from the non-keyword speech. Two different sets of keyword models were used. The first one is the context-independent phones, trained from the whole training data. The second is the word-dependent phones, trained only from the keyword instances. The ortographic transcriptions of the training data were used to perform forced alignments which produced transcriptions composed of phones for the non-keyword words and whole words for the keywords. For the broad phonetic class filler model, the phones are replaced by the corresponding cluster label, keeping the keywords as before. Such transcriptions were also used to build the bigram used in the LM component of the decoding process. Experiments used the Air Travel Information Service (ATIS) [49] and the task was to detect 61 keywords including city names, airlines, etc. They showed that the context-independent phones as filler models achieve an acceptable compromise in terms of system performance and time computation.

It improved the three configurations of the broad phonetic classes. It was only improved by an LVCSR system. They also showed that, as expected, the use of word-dependent phones for the keyword models improved the rate achieved by the context-independent phones due to only the keywords presented in the speech are used to build the acoustic models.

El Méliani and O'Shaughnessy [50] presented a comparison between acoustic and strict lexical filler models. In the latter, the distinction between keywords and non-keywords was made only at lexical level. Therefore, both keywords and non-keywords were represented by the same set of context-dependent models trained from the whole training corpus. They proposed two different types of lexical filler models: The first one was composed of a different lexical filler model for each phoneme, which leads to the best performance although the LM needs more memory and more storage capacity. The second one was composed by the set of syllables. On the other hand, for the acoustic-phonetic filler models, they trained two different sets of context-dependent phonemes, one for the keywords and another for the fillers. In this case, the lexicon is completed by the phonetic forms of the filler models (phonemes and syllables depending on the filler model), apart from the keywords. As LM they used a bigram from the keywords and filler (acoustic and lexical) models. It was trained for both filler models from the OOV words of the speech corpus. In their experiments, they chose the Wall Street Journal database and showed that the strict lexical fillers gave a slight worse detection than the corresponding acoustic-phonetic ones. However, it is not necessary to retrain the context-dependent phonemes when the list of keywords changes, which makes the system dramatically dependent of the list of keywords, and impractical in most of the cases. Nevertheless, the LM still needs to be retrained.

Cuayahuitl and Serridge [35] also proposed phonemes as filler models, but investigated the use of syllables and common words as fillers as well. They used the Spanish language for their experiments and chose a set of 24 phonemes for the phoneme-based filler model, a set of 49 common syllables for Spanish to build the syllable-based filler model and a combination of this filler model with a set of 30 common words to build the word-based filler model. The keyword models were built from the set of phonemes in Spanish. For each wordspotter, a bigram LM was used. This bigram was trained from data containing transcriptions composed of whole-words for the keywords and phonemes, syllables and/or one of the 30 common words depending on the filler model used in each wordspotter for the non-keyword intervals. For such non-keyword intervals, they used a set of 92009 words in Spanish. They restricted their system to appear a single keyword or no keyword in each utterance. Experiments used the auto attendant system (CONMAT) developed in the Universidad de las Américas in México, selecting 2288 words from it. They showed that the phoneme-based filler model achieved the best result, although a posterior second step in their work proposed a confidence measure that allows the syllable-based filler model to improve slightly the final rate of the two-stage system paying a considerable price in computational cost.

Xin and Wang [37] reported their keyword spotting system over Mandarin language and used the syllable as basic unit, due to the Chinese is a monosyllable language and to achieve a reasonable compromise between robustness and flexibility. Each keyword model was built from the concatenation of two or three syllables. As filler model they used the combination of an anti-syllable model for each syllable, trained from the data of all syllables but that of syllable, with a general acoustic model trained from the non-keyword speech.

OU et al. [38] also presented a keyword spotting system for Mandarin and made a clustering of the 21 initial consonants of such language in a single filler model and the 153 tonal finals into six different filler models. A whole-word model for OOV words trained from the non-keyword speech was incorporated within the filler models to complete the 8 filler models used in the system. The keyword models were a concatenation of the original set of phones. All of the models, both keyword models and filler models were context independent. The experiments were performed with utterances that contain a single keyword or a single OOV word, so a simple unigram with the same probability for each keyword and filler model was used as LM.

Kim et al. [36] also proposed the use of phonemes as filler models. But they presented a new approach focusing on the LM component when the decoding process is run in the keyword spotting system. They proposed a uni-gram as LM (pseudo N-gram), instead of high-order LMs. In so, they varied the probability of retrieving a keyword or a filler model of such uni-gram. In fact, it was found to be substantial in those keyword spotting systems where the keyword models are built from the same acoustic units than the fillers and no more complex LMs such as bigrams, trigrams, etc are used. It must be noted that, contrary to Paul and Rose [33], the same weight (probability) is assigned to all the keywords, and the same weight (probability) is assigned to all the filler models. They used the 445-database of the Electronics and Telecommunications Research Institute (ETRI) and the YNU-database in Yeungnam University, selecting 45 keywords from them. They showed that when the uni-gram probability of keyword and filler models is set to 0.8 and 0.2 respectively, the system performs the best.

3.3.2 Confidence Measures for Keyword Spotting

Confidence measures have been demonstrated to be a powerful method to increase the performance of the ASR systems, i.e., to reduce the WER [51, 52]. Therefore, they also play a very important role in Keyword Spotting, with the main objective of increasing the final system performance.

3.3.2.1 Confidence measures for HMM-based Keyword Spotting

Cuayahuitl and Serridge [35] proposed a confidence measure based on the confidence score provided by the Speech-Works recognizer when the decoding process is run. They divided the confidence score range into three different parts: the *rejection region* containing the scores from 0 to LT (Low Threshold), the *confirmation region* containing the scores from LT to HT (High Threshold) and the *acceptance region* with scores higher than HT. They gave a different weight α_i for all of the possible results of a recognition event (correct acceptance INV, correct confirmation INV, false acceptance INV, false rejection INV, false confirmation INV, correct rejection OOV, false acceptance OOV and false confirmation OOV) and tried to minimize the Discriminative Error Rate (DER) in the following equation for all of the possible combinations of HT and LT with HT > LT:

$$DER = \frac{\sum_i \alpha_i \beta_i}{N} \tag{3.1}$$

where β_i is the number of occurrences of each type of recognition event and N is the number of total occurrences. The lowest DER according to their baseline was reduced from 0.132 to 0.101 after applying the confidence measure.

Xing and Wang [37] in their syllable-based-HMM keyword spotting system explained in Section 3.3.1.5 proposed a utterance-based confidence measure combining the confidence score (log likelihood score) of all of the syllables in the keyword and their corresponding anti-syllables calculated during the syllable-based decoding process. If such confidence score is above a preset threshold, the keyword is accepted; otherwise, the keyword is rejected. First, they computed the score for each syllable as the substraction of its log likelihood score minus the log likelihood score for its anti-syllable model, divided by the duration of the speech segment. They found that a normalization of such score by substracting the mean and dividing by the variance assuming a Gaussian distribution for each syllable in the keyword decreased the FAR, from 12% to 7%, with no decrease in the hits rate (87.5% of detection rate) compared with the absence of such confidence measure. They used a set of 20 city names over a CTS database, with 205 utterances responding to these 20 city names.

OU et al. [38], in their Mandarin place name recognition keyword spotting system, presented a NN-based confidence measure to confirm or reject the keywords hypothesized by the HMM-based keyword spotting in the first level, consisting of filler models and an anti-keyword model. The training and test of the NN were made from a five-dimension vector estimated for each keyword, and contained the following information: the highest and the second highest likelihood scores produced by the keyword models, the average likelihood score of the top N likelihood scores produced by keyword models and the posterior probability produced by the filler models and the anti-keyword model. The NN accepts a putative hit (true hit) if the difference between its two output nodes (corresponding to hit and FA) remains above a certain threshold. For the experiments, 464 utterances contained one of the 144 names used as keywords, while 879 utterances were OOV words. They selected as baseline the confidence score built from the highest likelihood score produced by the keyword models averaged over the rest top N likelihood scores produced by the keyword models. Therefore, in the baseline, a keyword hypothesized is accepted if the confidence score remains above a certain threshold. In the NN approach, the keywords classified as true hit remain in the final output. Comparing several NNs (feed-forward propagation, Elman backpropagation and cascade-forward backpropagation), they showed that the NN-based approach can reduce the average error rate in 54.4%.

3.3.2.2 Confidence measures for LVCSR-based Keyword Spotting

Hazen and Bazzi [40] presented a word confidence scoring prior to define a threshold-based confidence measure from which keywords with a score below it are automatically rejected. To produce such confidence, they used ten different features (the average normalized likelihood score over all of the observations in a word, the minimun normalized likelihood score for a word, the fraction of the N-best hypothesis in which a keyword hypothesized appears, etc) with which they built a single confidence feature vector. To calculate a final score from such vector, they used a simple linear discrimination projection vector. This vector was trained using a minimum classification error (MCE) training strategy. A latter computation from this vector and the vector containing the features reduces the multi-dimensional confidence vector composed of the features to a single confidence value. They applied this confidence measure as a post-processing of an LVCSR system, whose lexicon is composed by keywords and non-keywords with the final objective of rejecting the OOV words (i.e., words that do not appear in such lexicon). To detect OOV words of the speech signal, comparing this confidence measure with the filler model-based approach to absorb the non-keyword segments, the latter works better due to the filler model is built specifically for this task, while the former is designed to detect any type of recognition error. However, to detect recognition errors, when they only focused on those relevant keywords (937 proper names from the geographical Jupiter domain), these two methods perform almost the same. An approach that combines both methods (filler model to detect OOV words in the recognition process and the post-processing confidence measure) outperformed each method in isolation in the final set of those 937 keywords.

Ben Ayed et al. [53] proposed the use of SVMs as confidence measure for a post-processing of the keywords extracted from the output of an LVCSR system. The features computed for each of those keywords contained parameters such as the total number of frames, the number of frames of the first and the last phone, the minimum and maximum phone posterior probability, the number of phones, the average per-frame phone posterior probability, etc according to the phones and frames of each keyword hypothesized. They used these parameters as input features for the SVM, using the package in *http://www.kernel-machines.org/*. The final objective was to reject those keywords classified as incorrect by the SVM. They used linear and Radial Basis Function (RBF) kernels in the SVM training and experimented with 10 keywords extracted from the French BREF80 database.

This approach showed promising results rejecting the incorrect keywords proposed by the LVCSR system. More advances made in such address by Ben Ayed et al. [54] showed that the use of the linear and RBF kernels in the SVM training, when using as input feature vector the arithmetic mean, the harmonic mean and the geometric mean for each word computed from the posterior probability of each phone in the keyword, outperformed the rate achieved by the best mean-based confidence measure (harmonic mean) in classifying the keywords output by the LVCSR system as correct and incorrect. For these experiments, they selected 20 keywords from the French SPEECHDAT database.

3.3.2.3 Confidence measures for Keyword Spotting without filler models

Ferrer and Estienne [55] presented a confidence measure to form a two-level keyword spotting system, which improved the final system performance. The first level is based on a modification of the Viterbi algorithm which proposed putative keywords at several time positions of the input signal without using filler models [56]. It takes the sequence of features according to the utterance, the set of HMMs which represent the keywords and generates a set of score signals C_1 for each keyword. This signal can be seen as the distance from the optimal sequence of states over each keyword model between a time interval, to the best sequence of states in the same time interval. The best state for each time is the one with the highest emission probability given the feature vector at that time. The decision stage generates a new detection each time the score signal C_1 for each keyword remains under a preset threshold. It must be noted that a different threshold for each keyword was used in their system. The final list of keywords with the beginning and end points and the score signals are taken by the second level. With such data, a vector Vk of length L is generated by the Viterbi algorithm when is run against the keyword model. This vector contains the emission probabilities for the optimal state sequence obtained by maximum likelihood according to the keyword model. A posterior phoneme-based recognition process, using the Viterbi algorithm, over the segments which represent the keyword, stored a similar vector Vp composed of the emission probabilities according to the optimal state sequence corresponding to the sequence of phonemes output by the phoneme-based decoding. In this way, when the keyword hypothesized is very likely to occur, both vectors Vk and Vp will be very similar. In case the sequence of phonemes produced by the phoneme-based recognition process matches exactly the correct transcription of the keyword, both vectors will be the same. The confidence score generated from these two vectors was based in the mean square rate as indicated in the following equation:

$$C_2 = \frac{\sum_{i=0}^{L}(Vk_i - Vp_i)^2}{L} \tag{3.2}$$

The final confidence score, from which they decided if the keyword was spoken in the utterance or was not, was computed by means of a linear combination of both scores C_1 and C_2, with a different weight for each. If such final confidence score is below a preset threshold, the keyword is accepted; otherwise, the keyword is rejected. The Spanish speech database at SRI was used for the experiments. 18 keywords were selected as the list of terms to test the system, with 882 occurrences of them. They showed that the combination of both scores outperformed the use of each in isolation. Moreover, when it is compared with a medium vocabulary continuous speech recognition system, composed of 700 words, its performance is 64.2 as FOM, whereas a 73.5 as FOM was achieved by the continuous speech recognition system.

3.3.3 Spoken Term Detection

Although methods dealing with STD can be also applied for Keyword Spotting, this section describes methods just able to STD. Altough STD itself emerged in an iniciative proposed by NIST at the end of 2006, methods used for this task have been developed for years, due to its capacity to be applied on Keyword Spotting and, therefore, on applications for audio information retrieval.

3.3.3.1 1-Best word-to-phoneme conversion plus lexical access for STD

Amir et al. [57] generated the word transcriptions using the IBM speech recognition system with a LM trained from BN data. The equivalent sequence of phonemes is generated from such word transcriptions. A posterior method based on the Minimum Edit Distance (MED) plus a likelihood ratio thresholding is used to hypothesize the final list of query words. They built a phoneme-based confusion matrix, which represented the probability of substitution, deletion and insertion for each phoneme. Based on that confusion matrix, they identified the phonemes that are more likely to be confused with each other. They formed groups of seven *metaphones*, each of these containing between two and ten similar phonemes and added them to the final confusion matrix. Based on the MED criterion and on dynamic programming (DP), they computed the necessary transformation which converts the sequence of hypothesized phonemes H into the correct sequence of phonemes C for each query word, using a sequence of single phoneme operations (substitution, deletion and insertion). Such transformation computes the maximum likelihood (or mininum cost) that allows to convert H into C. Query words with a likelihood above a preset threshold are hypothesized in the final output of the system. In the test of the system, they previously converted the sequence of phonemes of the query word into several three-phoneme keys, each of them composed of three consecutive phonemes. Each key that contains one or more phonemes in the *metaphones* group is indexed using this representation, and therefore using the same *metaphones* data in the confusion matrix, by replacing each phone by the corresponding metaphone. They reported their experiments over the spoken document retrieval field using data from HUB4 [58] and showed that the combination of the LVCSR-based approach and this approach improved between 5-15% the system performance for INV words compared with the use of the LVCSR-based approach in isolation.

3.3.3.2 Phone-Ngrams and Lattice based STD

Dharanipragada and Roukos [59] presented a new algorithm to spot words in speech which complements a standard LVCSR system for OOV words. They presented a method composed of a phoneme-based recognition and a search with two different steps which aimed to maximize the final system performance. The phoneme-based recognition consisted of a phone-Ngram representation at all time locations, where the speech was converted, by means of a time-synchronous Viterbi-beam search, into a table composed of phone-trigrams with their times (beginning and end) of occurrence and their normalized likelihood scores (acoustic scores). To produce such phone-trigrams, the standard HMMs, typically composed of three states, were substituted by a single-state model consisting of frame-triplets, that worked on one-third the frame-rate. It causes that each phoneme has a minimum duration of three frames (as the standard way) but only allows durations in multiple of three frames. The phone-trigrams were extracted from a graph built during the Viterbi-based decoding. A prefix tree was used as LM, consisting of arbitrary sequences of words within the vocabulary, due to its better phoneme accuracy than a trigram-based LM. Each of the nodes in the tree corresponds to a phone of the word in the dictionary and each of the leaves corresponds to an end of a pronunciation of each word. Self-loop probabilities and uniform probabilities to the transitions between each node and the following were assigned in the LM. For OOV words, the prefix tree is smoothed to allow for unknown phone-trigrams to be indexed in the search step. The search stage is composed by two different steps: the first one is the *"coarse acoustic match for putative hits"*, where two different parameters were found to be relevant to decide if the keyword was spotted or was not: The first one is the number of phone-trigrams belonging to the keyword that appears in the table got in the first step. For keywords which contained less than three phonemes, all trigrams that begin or end with the phonemes in the word are considered. The second parameter is the likelihood of such phone-trigrams stored in that table. Keywords remaining after this first step (with a number of phone-trigrams and a likelihood value that remain above two preset thresholds) serve as input for the second and final step, the *"detailed matched at the putative hits"*, composed itself of two different stages: In the first one, the Viterbi algorithm is used to find the best path through a network composed of all of the alternative pronunciations for each keyword and a filler model (composed by a loop of phonemes) for the regions classified as putative keywords by the *"coarse acoustic match for putative hits"* step. This algorithm also hypothesizes the beginning and end times for each putative keyword. The second stage calculates a score for the filler model and for each of the different pronunciations of the keyword between the start and end times computed in the first stage. Finally all the putative hits are ranked based on the best normalized likelihood ratio among the different pronunciations of the keyword. It is computed from the substraction of the log-likelihood ratios of each pronunciation and the filler model calculated by the Viterbi algorithm. Experiments, performed over the 1996 English Broadcast News speech corpus, selecting 36 OOV words, showed that the addition of the *"detailed matched at the putative hits"* step, increased the system performance in a factor of two compared with the *"coarse acoustic match for putative hits"*. The main drawback of this system is that it suffers from a large index size by indexing individual keyword locations.

To solve the problem of such big index [59], Yu and Seide [60], proposed a system based on two stages as well. The first stage calculates an index in a fast step, from which the second stage can hypothesize the final list of keywords according to each utterance. This first stage is based on keeping segments in the audio signal, represented by means of phoneme M-grams (extracted from a lattice), as index, whose expected term frequency (ETF) remains above a given threshold. This ETF was approximated for each segment by using M-gram LMs. The second stage is based on the DP algorithm proposed in [61]. It finds all the paths contained in the lattice represented by the phoneme M-grams which perform an exact match with the actual transcription of the keyword. Experiments, run over the LDC Voicemail corpus [47], with 6058 keywords, 2295 of which were OOV, reported that this two-stage method presented only about 4% FOM relative less than the one based on a full linear search over the whole lattice in which all of the segments in the utterance are computed, being 25 times faster.

Scott et al. [62] presented a system to index conversational speech based on a heuristic score computed from the expected posterior counts of phone n-grams within the recognition lattices. In the first step, a phoneme lattice was generated using the Viterbi algorithm in the HTK tool [63]. To build the final index from such lattice, all of the expected phone n-grams for $n <= 5$ are computed and those whose expected value is less than a preset threshold are rejected in the final index. The expected value is computed from the number of times that the phone n-gram appears in the lattice and the widely used posterior probability computed from the forward-backward algorithm during the lattice generation for such n-gram. The second step builds the keywords from the previous index. It is based on a slide-window over the sequence of phonemes corresponding to the query. Each element of this window is searched in the lattice composed of phone n-grams and if it is found, a score for the query is computed from the expected value of such n-grams. Experiments were done over English, Spanish, Mandarin, Levantine and Persian, using the CTS corpora provided for each by the LDC. In comparing the system with a search over the 1-Best sequence of phonemes during a phoneme decoding and a DP-based MED procedure that used a confusion matrix to deal with insertion, substitution and deletion errors of such sequence, they showed better performance for all of the languages in terms of Mean Average Precision (MAP).

Thambiratnam and Sridharan [64] presented a fast algorithm to index speech based on Dynamic Match Lattice Spotting (DMLS). It combined the fast performance of lattice spotting with DP-based matching algorithms to obtain a desired system performance. In the first step, a phoneme lattice is generated by using a bigram

LM. Next, the resulting lattices were expanded using a four-gram LM and were pruned removing all paths in these lattices with a total likelihood outside a certain threshold of the best path (*lattice pruning*). Finally, the Viterbi algorithm was used to generate the top ten sequence of phones of length 11 at each node in the lattice (*Viterbi pass*). The second step employs a DMLS algorithm. It takes each node in the lattice and searchs within the lattice for the paths that contain the sequence of phonemes whose similarity with the sequence of phonemes of each keyword remains below a threshold to hypothesize such keyword. This similarity is measured by the MED between both sequence of phonemes by using a confusion matrix. Several aspects such as the depth of the lattice, the pruning beamwidth in the *lattice pruning* stage, the number of tokens in the *Viterbi pass* and the MED threshold were investigated in the work. The MED threshold was found to be substantial to achieve the desired tradeoff between misses and false alarms. Two methods applied in the search algorithm also increased the speed of the system. In this address, the estimation of the MED in similar sequence of phones (prefix sequences) is not necessary each time the same prefix sequence appears in the lattice. A second method relied on the comparison of two sequences of phones that are very different. Each element in the MED matrix stores the cost necessary to transform a phoneme in the lattice into the phoneme of the keyword. In this way, this method limits the portion in the MED matrix to be computed, by estimating a lower bound in the column of that matrix which exceeds the MED threshold. Experiments were run over the Switchboard and TIMIT databases, selecting 200 words of six phonemes each, appearing 480 times in the TIMIT database and 360 words of six phonemes each, with a total of 808 times in the Switchboard corpus. The DMLS algorithm got excellent detection performance itself and with the two speed optimizations increased the speed of the algorithm in a factor of five.

3.3.3.3 A phone-state based matrix for STD

Gao et al. [65] proposed an approach based on a phone-state matrix computed during a phone-based HMM decoding process and a posterior algorithm to hypothesize the keywords from such matrix. This matrix has the time in one dimension and phone-state in the another dimension and each element is the phone-state score, obtained during the HMM decoding. This matrix stores for each time t, the maximum score (i.e., the maximum likelihood) for each state of a phone, denoted as Lph, from the triphones which contain the phone. The decoding process also stored the maximum score (likelihood) of any state of any phone for the same time t, denoted as $Lmax$, and the score for each time t by adding all the scores for all of the states Q_t of all of the phones according to the observation X_t. These last scores are represented by $P(X_t|Q_t)$, and the addition of them is denoted as $Lall$. The second step builds the keywords from the phone-state matrix and the elements stored during the decoding process. No LM was used in both steps. Keywords are proposed by a Viterbi-based search over the matrix stored in the first step. This search hypothesizes a putative keyword beginning in every time t initializating each new path by adding the filler information $Lmax$ from the beginning of the utterance ($t = 0$) to the current time t. When the optimal path reaches a keyword end, the confidence of such path is computed based on the Bayesian rule according to the values stored in the phone-state matrix Lph and $Lall$. Finally the keyword is proposed if such confidence remains above a preset threshold. From the next time t, this process is repeated until the time ends. Experiments were performed over Mandarin CTS data, selecting 100 words with 405 occurrences in total. They showed that their approach based on the novel phone-state matrix plus the subsequent search algorithm did improve the approaches based on a syllable and a phone lattice search (from a linear search in these sub-word units of the list of keywords) although got worse performance than the one based on filler models, applied over Keyword Spotting but impossible to be used over STD without making any use of the audio in the search stage.

3.3.3.4 Probabilistic pronunciation model for STD

Pinto et al. [66] proposed a novel probabilistic pronunciation model for each keyword which compensated the errors (insertions, deletions and substitutions) that appear during the 1-Best phoneme-based decoding. In this way, it differs from other approaches because HMMs to integrate contextual error information are used. First, the 1-Best sequence of phonemes was extracted from the audio signal by means of the Viterbi algorithm using a bigram as LM. A confusion matrix to deal with insertion, deletion and substitution errors is trained from such sequence of phonemes. Each keyword model is represented by an HMM in which the phonemes of the keyword are used as hidden states of the model. An additional insertion state (denoted as $*$) before each phoneme for each keyword model to deal with insertion errors was also modeled. It is accessed with a probability P_i, which represents the unconditional probability of insertion and is skipped

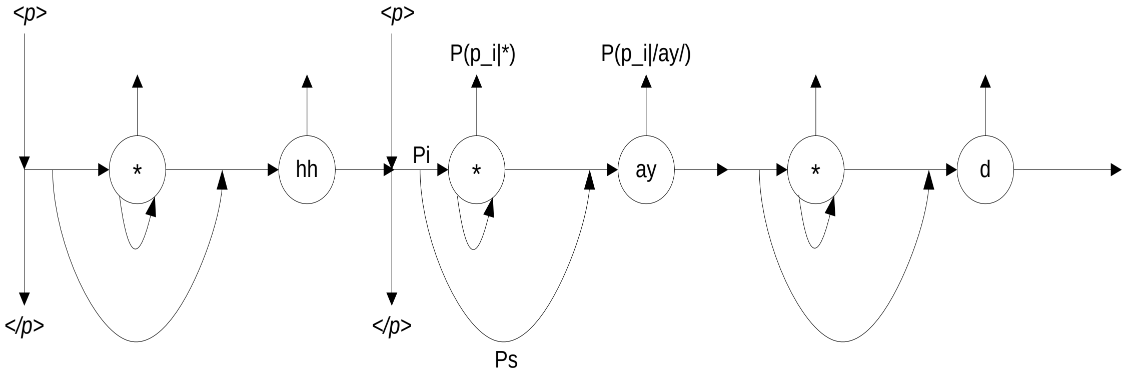

Figure 3.1: A probabilistic pronunciation model for the keyword *hide* with a transcription of /hh/ /ay/ /d/. The symbol '*' denotes the hidden state for insertion. $<p>$ and $</p>$ denote the entry to the model and the exit from the model respectively. P(p_i|*) refers the insertion probability of the phoneme p_i.

with a probability of P_s, which represents the unconditional probability of substitution. Both values are stored in the confusion matrix. They considered the deletion error as a special case of substitution error. The insertion state emission probability is given by the phoneme insertion probability stored in the confusion matrix as well. The emission probability in each phoneme state p_i is given by the conditional probability $P(p_j|p_i)$ stored in the confusion matrix. It represents the probability of the phoneme p_i to be substituted by the phoneme p_j. The deletion of the phoneme p_i is represented by p_j='*'. The HMM is relaxed by allowing a begin and an end in each phoneme of the keyword model. An example of an HMM as probabilistic pronunciation model is depicted in Figure 3.1. In it, it is shown that the contextual information for each phoneme of each keyword is used. The search step was applied over the 1-Best sequence of phonemes and consisted of a Viterbi-based search on a slide window of N phonemes with a shift of 1 phoneme. A garbage model, built from a phoneme-Ngram LM is added to compute a reference score with which compare the keyword model score. The Viterbi algorithm takes the sequence of phonemes hypothesized by the decoding process over each slide window, the HMMs of the search terms and the HMM which represents the garbage model and hypothesizes the list of terms whose score is higher than the score of the garbage model matching. Experiments were run over the CTS data in the NIST STD evaluation 2006, selecting 243 search terms. Comparing this system with a phoneme-based lattice system, whose score is computed from the standard forward-backward algorithm and whose search terms are given from an exact match with the actual transcription of the term over the phoneme lattice by using a recursive match algorithm, it presents a worse STD performance (especially for keywords with less than 10 phonemes). However, it presents a better search speed (about 14 times faster) and a smaller index size (about 1500 times less memory).

3.4 Summary

First, we have reported the main categories related to the OOV modelling in continuous speech recognition and have discussed if they can be applied on Keyword Spotting and STD to solve the OOV problem. Later, we have described the most relevant techniques applied so far on Keyword Spotting and STD. We have also described the common filler models used for Keyword Spotting.

Chapter 4

Contributions to Keyword Spotting

4.1 Introduction

Contrary to LVCSR systems, where each word contributes equally to their performance (measured in terms of WER), in Keyword Spotting, just a few words are important. In this way, model such words in an identical fashion as those in an LVCSR system and absorb the rest of the words of the speech by using some other models (called filler or garbage models or fillers) is the key point for HMM-based keyword spotting systems. This chapter presents the filler models used to absorb those non-keywords of the speech data and introduces several confidence measures in a second level to improve the rates achieved by an HMM-based keyword spotting process in the first level. Therefore, the contributions in this chapter rely on the confidence measures proposed after a state-of-the-art HMM-based keyword spotting process. This chapter is divided into five different parts. The first part presents the HMM-based keyword spotting process along with the filler models used in the first level. The second part introduces the confidence measures used in the second level, along with the additional modules necessary for them. The third part describes the experimental setup used to evaluate the keyword spotting approaches, the fourth one presents the results using the Albayzin database and finally the fifth part describes the main conclusions. The framework of the keyword spotting system is presented in Figure 4.1.

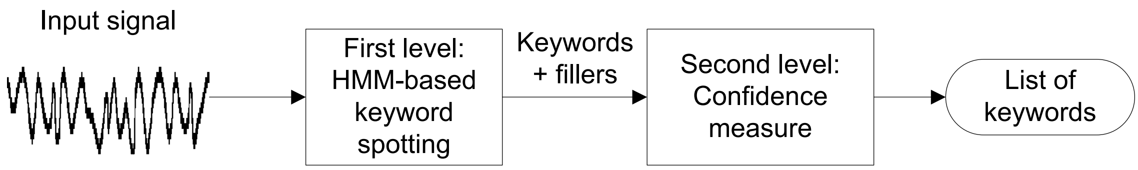
Input signal
First level: HMM-based keyword spotting
Keywords + fillers
Second level: Confidence measure
List of keywords

Figure 4.1: The framework of the keyword spotting system.

4.2 First level: HMM-based keyword spotting process

This process is used as first level for all of the approaches presented for Keyword Spotting in this chapter. It is based on the standard procedure of an LVCSR system, with these main differences: the lexicon of the system is composed by the set of relevant keywords to hypothesize from the speech data plus the filler models, and the LM is also built from those keywords and the filler models. The Viterbi algorithm is used to present the output of this process as a sequence of keywords and filler models (referred as *Keywords+fillers* in Figure 4.1).

The filler models used in this first level are configured as follows:

- Allophone Models (AM): It is composed of a set of 47 phones in Spanish language [21].
- Phoneme Models (PM): It is composed of the standard set of 24 phonemes in Spanish language [22].
- Broad class Models (BM): It clusters the standard set of phones in Spanish language in eight different classes as follows: nasals, closed vowels, opened vowels, median closed vowels, deaf plosives, deaf fricatives, sound plosives and liquids. Appendix C shows the phones that are contained in each class.
- Average Phoneme Model (APM): It considers all of the phones as a single filler, so this filler model is composed of a single model.

A beginning and end silence for each sentence in the Albayzin database were added to each filler model.

To train a LM from text data, which is composed of keywords and non-keywords, causes that such LM should be estimated again when the list of keywords (vocabulary) changes. And, as stated before, the decoding process must be also rerun. In this thesis, we have tried to minimize the components of the system TO modify when a change in the vocabulary occurs. For the same reason, the acoustic models used to build the keywords were trained on the whole training set, (i.e., not only using the keywords), and the filler models were trained from the same whole set (i.e., not only using the non-keywords). On the other hand, it is well-known that when the keywords are modeled by the same acoustic units (phones in our case) that the filler models, the Viterbi-based decoding tends to hypothesize the sequence of phones instead of the keyword identifying it. In this way, in the spirit of Kim et al. [36], the LM used in this first level is defined by a uni-gram where the frequencies of appearances in such uni-gram for both keywords and filler models may differ (pseudo N-gram). Therefore, the LM used follows this equation:

$$prob(kw) = N * prob(filler) \tag{4.1}$$

where kw denotes a keyword, $filler$ denotes a filler model and N denotes a penalty for the filler models against the keywords.

As an example of how the LM is built, in case $N = 3$, we will have the probabilities for each keyword and for each filler model as follows:

$$prob(kw_i) = \frac{3}{3L + M} \tag{4.2}$$

$$prob(filler_i) = \frac{1}{3L + M} \tag{4.3}$$

where L denotes the number of keywords and M denotes the number of filler models. It must be noted that the addition of the two probabilities for the L keywords and the M filler models equals to 1.

Figure 4.2 presents the recognition network used in this first level. Any transition between keywords and fillers is allowed as well as self transitions for both keywords and fillers. This configuration allows multiple keywords to appear in a single utterance and multiple instances of the same keyword in the same utterance. The keyword HMMs are constructed as concatenations of phone HMMs, so no additional training is required.

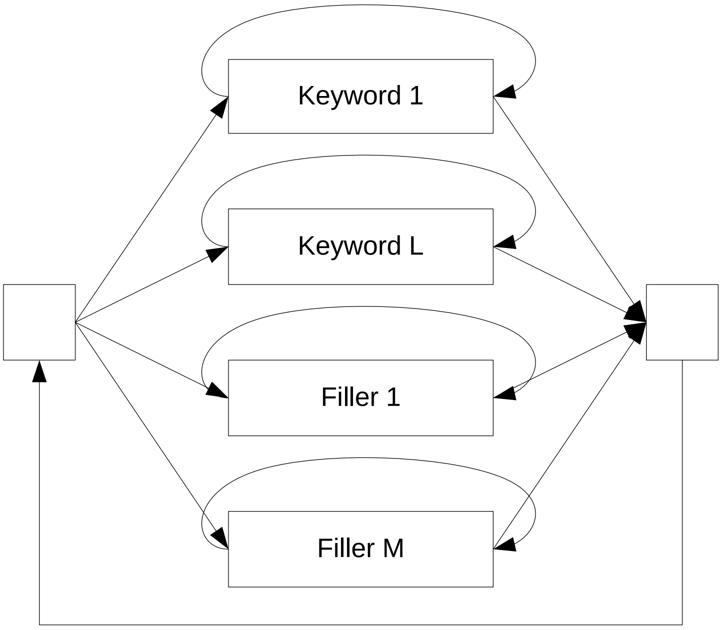

FIGURE 4.2: The recognition network in the first level: HMM-based keyword spotting process.

The confidence score is computed from the Viterbi algorithm during the decoding process. Therefore, for each keyword, it is the sum of the acoustic log likelihood, the word insertion penalty and the LM log likelihood weighted by the language scale factor.

4.3 Second level: Confidence Measures

Errors in Keyword Spotting come from two different scenarios. The first one is produced when a keyword that appears in the speech data is not hypothesized by the system. The second refers to a keyword hypothesized by the system which actually does not occur in the speech data. The former is referred as *miss* and the latter is referred as *false alarm*. On the other hand, a *hit* occurs when the keyword hypothesized by the system is presented in the speech data. So, the relationship between *hit* and *miss* is expressed as follows:

$$prob(hit) = 1 - prob(miss) \tag{4.4}$$

It is obvious to conclude that a system should minimize the number of misses (i.e., maximize the number of hits) and the number of FAs. Nevertheless, what actually happens in keyword spotting systems is that decreasing the number of misses leads to an increment in the number of FAs and that decreasing the number of FAs leads to a dramatical increment in the number of misses. Therefore, a tradeoff between both errors should be found. To conclude, confidence measures are used in Keyword Spotting with the purpose of decreasing the number of FAs as much as possible while maintaning the number of misses as low as possible.

Typically, confidence measures extract a set of relevant parameters in a first stage and take a decision based on such parameters about accepting or rejecting the keyword in a second stage. Therefore, this is the strategy followed to implement the confidence measure for each keyword spotting system. For all of the confidence measures, the same confidence score output by the first level remains in the final output of the system. Next, the confidence measures developed in the thesis are described.

4.3.1 Exact Match

4.3.1.1 Motivation

In using the Viterbi algorithm over continuous speech recognition to calculate the best path composed by a sequence of keywords and filler models, those keywords are influenced by the filler models, as the optimal path contains both keywords and filler models. Producing an N-best list from an HMM-based keyword spotting process has a very high computational cost and when it is applied over continuous speech, it is very likely that the two few candidates in the N-best list only differ in filler models and not in the keyword(s) proposed. And extracting a lattice from the input signal composed of keywords and filler models produces similar effects. Therefore, a second level which makes use of the acoustic information of the keywords, rejecting the filler models, and computes a confidence measure from the words in the lexicon is presented to improve the performance achieved by the first level in isolation.

4.3.1.2 System architecture

The whole keyword spotting system with this confidence measure is presented in Figure 4.3. It is composed of the *HMM-based keyword spotting* process in the first level which hypothesizes a set of putative keywords plus filler models, which are automatically rejected. These keywords are proposed to further verification in the second level, which is composed of two different processes. The first one is based on an *Isolated speech recognition*. It computes the keyword which best matches with each region of speech over which the first level has hypothesized each keyword (i.e., it computes the keyword with the highest likelihood). No LM is used in the *Isolated speech recognition*. The second one acts as a decision stage where keywords proposed by the first level are accepted or rejected. The *Decision stage* accepts the keyword kw if $kw = kw'$; otherwise, kw is rejected, where kw is the keyword proposed by the first level and kw' is the keyword proposed in the *Isolated speech recognition* process. This work has been published in [67].

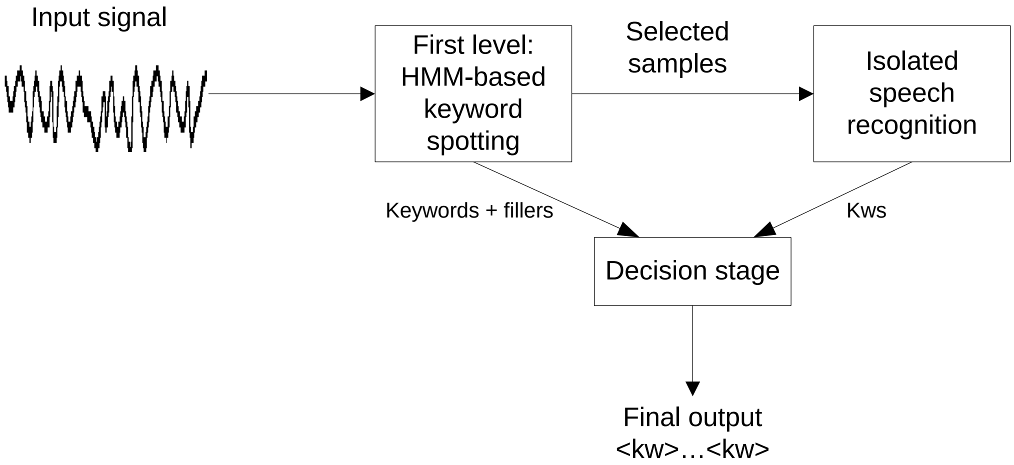

FIGURE 4.3: The system architecture for the *Exact Match* confidence measure in the keyword spotting system. <kw> denotes each keyword in the final output of the system.

4.3.2 Likelihood

4.3.2.1 Motivation

The *Exact match* confidence measure only considers the matching between the keyword hypothesized in the first level over the continuous speech recognition and the keyword hypothesized by the isolated speech recognition. Nevertheless, the widely used posterior probability and likelihood in speech recognition tasks [37, 51, 52, 53, 68, 69, 70] means that a simple modification in the final decision stage may produce a significant improvement in the system performance. Following such methods, Dolfing and Wendemuth [69] presented a confidence measure for an isolated word speech recognition, based on the differences between the likelihood of the N-best candidates. Here, we have followed the same approach to present this confidence measure on Keyword Spotting.

4.3.2.2 System architecture

The system architecture presented for this confidence measure is very similar to that of the *Exact Match* confidence measure. It is depicted in Figure 4.4 and only differs in the output of the *Isolated speech recognition* process. Apart from computing the keyword which best matches with the regions of speech according to each keyword hypothesized by the first level (i.e., the keyword with the highest likelihood in the lexicon), it also outputs the likelihood of each keyword in the lexicon for those regions of speech. The likelihood is computed from the acoustic log likelihood. Let kw be the keyword proposed by the first level and let kw' be the corresponding keyword with highest likelihood (i.e., the keyword which best matches with the regions of speech of the keyword kw). Defining X as the difference between the second highest log likelihood and the highest one and Y as the difference between the third highest log likelihood and the highest one, the *Decision stage* accepts the keyword kw if the Equation 4.5 is satisfied; otherwise kw is rejected.

$$kw = kw' \text{ and } X < X_{beam} \text{ and } Y < Y_{beam} \tag{4.5}$$

Therefore, the difference between this confidence measure and the *Exact Match* relies on the use of the thresholds X_{beam} and Y_{beam}.

It was found during prior research on isolated word speech recognition (out of the scope in this thesis) that using the three best candidates in the confidence measure is enough to achieve a reasonable system performance.

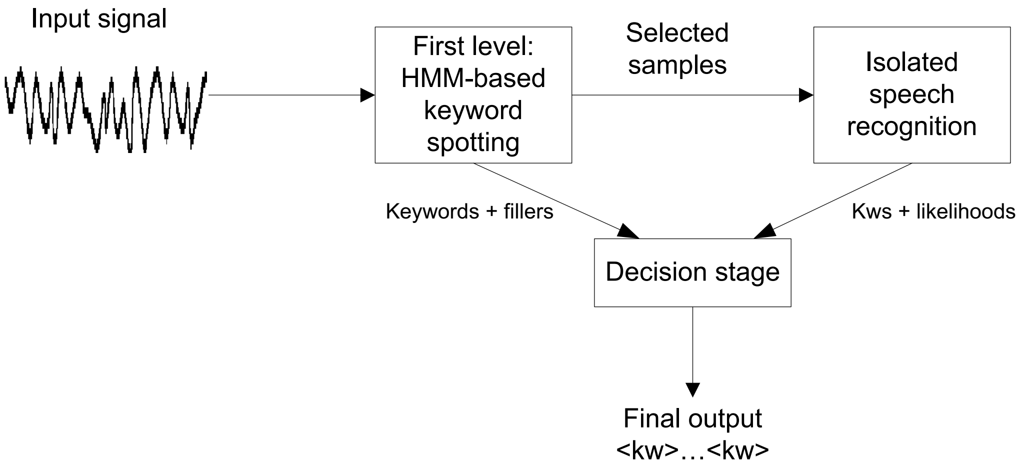

FIGURE 4.4: The system architecture for the *Likelihood* confidence measure in the keyword spotting system. <kw> denotes each keyword in the final output of the system.

This work has been published in [67].

4.3.3 Heuristic Rules

4.3.3.1 Motivation

Hybrid approaches for Keyword Spotting, combining word- and phone-based speech recognition, have been applied in the literature [46]. Such approaches try to benefit from the advantages of both methods. Whereas Yu and Seide [46] merged both methods to present the final output of the system, we have used the phone-based speech recognition in the framework of the confidence measure with the objective of rejecting the FAs presented in the *HMM-based keyword spotting* process.

Therefore, in running these two speech recognition processes at the same time, as more information is added to the system, it is expected that the system performance gets improved compared with the *HMM-based keyword spotting* process in isolation.

4.3.3.2 System architecture

The system architecture for this confidence measure is presented in Figure 4.5. Along with the *HMM-based keyword spotting* process, a *Sub-word unit decoder* is run. It outputs the most likely sequence of phones according to the input speech signal. A bigram LM is used in this *Sub-word unit decoder*. The three additional blocks of the system architecture complete the confidence measure to present the final output of the whole system as follows:

The *Substring selection* module takes the sequence of phones hypothesized by the *Sub-word unit decoder* within the time intervals of each keyword hypothesized by the first level (referred as *Selected strings* in Figure 4.5). This sequence of phones represents the output of the module and is passed towards the *Sub-word performance estimator* module.

The *Sub-word performance estimator* module calculates the number of phones in the *Selected strings* which are correct and incorrect (referred as *Cs* and *INs* respectively in Figure 4.5) according to each keyword proposed by the first level, along with the number of phones of that keyword (referred as *Ns* in Figure 4.5).

The *Decision stage* module takes those three parameters (*Cs*, *INs* and *Ns*) computed during the *Sub-word performance estimator* stage, rejects the filler models output by the first level and accepts each keyword hypothesized by the first level if Equations 4.6, 4.7 and 4.8 are satisfied; otherwise, the keyword is rejected.

$$C_s > IN_s + F_1 + |N_s - C_s - IN_s| \tag{4.6}$$

$$C_s > \frac{N_s}{2} - F_2 \tag{4.7}$$

$$\text{if } N_s < F_3 \text{ then Exact Match} \tag{4.8}$$

The purpose of the Equation 4.6 relies on the fact that the number of correct phones should be much greater than the number of incorrect phones in the sequence of phones to predict the putative keyword is a *hit*. The purpose of the Equation 4.7 focuses on the comparison between the number of correct phones and the number of phones of the keyword. Along with these two equations, in case the number of phones of the keyword is less than a factor $F3$, the keyword is considered to be correct just in case the sequence of phones does an exact match with the actual transcription of such keyword. This third Equation 4.8 was added to prevent the system with short-length keywords from producing a lot of FAs and to set the experimental factors $F1$ and $F2$ in such a way that the performance of long- and medium-length keywords does not degradate. This work has been published in [71].

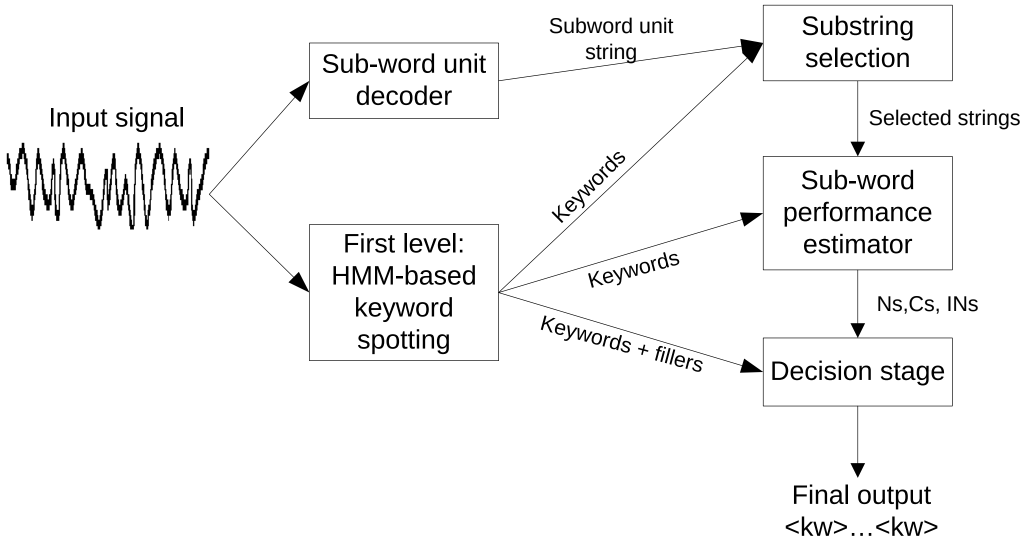

FIGURE 4.5: The system architecture for the *Heuristic Rules* confidence measure in the keyword spotting system. <kw> denotes each keyword in the final output of the system.

4.3.4 Lexical Access

4.3.4.1 Motivation

Fissore et al. [72] proposed a method by using a phoneme-based speech recognition system as a previous step to spot the words within an isolated word speech recognition. It takes the sequence of phonemes extracted during the phoneme-based decoding process and by means of a DP algorithm hypothesizes the word which best matches with such sequence according to the MED criterion.

This confidence measure follows the same approach as the *Heuristic Rules* but it differs in the set of parameters computed and passed towards the decision stage. Instead of using heuristic rules based on the errors produced in the sequence of phones during the phone-based decoding, we have used the DP algorithm proposed by Fissore et al. [72] to hypothesize which putative keywords output by the first level should be rejected.

Apart from the benefits of the *Heuristic Rules* confidence measure itself, it should be noted that the *Heuristic Rules* confidence measure does not make use of any additional training process to benefit from the phone-based speech decoding. Instead, in this *Lexical access* confidence measure, it is necessary to compute the confusion matrix to be used during the MED calculation in a previous training process. In addition to this, this confidence measure does also make use of the *hits* and *FAs* presented in the *HMM-based keyword spotting* process. Therefore, as more information is provided to the system in this confidence measure, it is expected that it improves the *Heuristic Rules* one performance.

4.3.4.2 System architecture

The system architecture with this confidence measure is depicted in Figure 4.6. The *Sub-word unit decoder* and the *Substring selection* modules are the same as those in the *Heuristic Rules* confidence measure. The difference proposed in this confidence measure relies on the *Lexical Access module* and the *Decision stage* as follows:

The *Lexical Access module* is composed of two different stages. The first stage is the training of the alignment costs, which are used in the second stage to compute the set of parameters passed towards the *Decision stage.* The training of the costs is described as follows: Each keyword W is represented as a sequence of R phone sub-word units $W = \{w^1, w^2, \ldots, w^R\}$, and search is performed within S, the output of the *Sub-word unit decoder.* This training is based on the DP algorithm proposed by Fissore et al. [72]. The algorithm computes the cost of matching each keyword W with the decoded output S. The total cost is computed from the costs of four types of alignment error: substitution, insertion, deletion, and continuation. The first three of these are standard in ASR decoding, and 'continuation' [72] is included in order to distinguish an insertion error from, for example, hypothesizing $\{baa\}$ during a time interval in which the correct phone sequence is $\{ba\}$.

Different costs are associated with each type of alignment error and are estimated as follows:

$$\begin{aligned} C_{sub}(h,k) &= -\log \frac{N_{sub}(h,k)}{N_{tot}(h)} && (4.9) \\ C_{ins}(h,k) &= -\log \frac{N_{ins}(h,k)}{N_{tot}(h)} && (4.10) \\ C_{del}(h) &= -\log \frac{N_{del}(h)}{N_{tot}(h)} && (4.11) \\ C_{con}(h,k) &= -\log \frac{N_{con}(h,k)}{N_{tot}(h)} && (4.12) \end{aligned}$$

where we define:

$N_{sub}(h,k)$	total substitutions of test symbol k for reference symbol h
$N_{ins}(h,k)$	total insertions of test symbol k after reference symbol h
$N_{del}(h)$	total deletions of reference symbol h
$N_{con}(h,k)$	total continuations of test symbol k after h

and $N_{tot}(h)$, the total occurrences of reference symbol h, is given by

$$N_{tot}(h) = \sum_k [N_{sub}(h,k) + N_{ins}(h,k) + N_{con}(h,k)] + N_{del}(h) \qquad (4.13)$$

The second stage in the *Lexical access module* determines the cost of matching each lexicon word to the hypothesized sequence of phones. DP is used to calculate the overall cost of matching each keyword W against the hypothesized sequence S. For such purpose, the *lexical access algorithm* is described as follows: Letting r and u be indices for the position within W and S respectively, the local cost function $G(r, u)$ is calculated in recursively as:

$$G(r,u) = \left| \begin{array}{l} G(r-1, u-1) + C_{sub}(w^r, s^u) \\ G(r, u-1) + C_{ins/con}(w^r, s^u) \\ G(r-1, u) + C_{del}(w^r, s^u) \end{array} \right. \tag{4.14}$$

where

$$C_{ins/con}(w^r, s^u) = \left| \begin{array}{ll} C_{ins}(w^r, s^u) & \text{if } s^u \neq s^{u-1} \\ C_{con}(w^r, s^u) & \text{otherwise} \end{array} \right. \tag{4.15}$$

Finally, the local cost function $G(r, u)$ is divided by the length of the decoded output S to normalize the cost computed for different-length keywords.

In this way, confidence measures can be derived from both relative and absolute cost of keywords. For example, if the second-best matching keyword has a cost which is close to that of the lowest cost keyword, then we can assign low confidence to the match. Similarly, if the absolute cost for the best matching keyword is high, then we also have low confidence in this match.

We adapt this idea for detection of FAs as follows: The *lexical access algorithm* is run twice, first using a set of costs estimated against the keywords which were correctly detected by the first level. This identifies a best matching keyword in the lexicon K_{best}, along with its match cost G_{best}. In the second run of the *lexical access algorithm*, a set of costs trained on FAs produced by the first level, is used to return the lowest cost G_{FA}. Both sets of costs associated to the hits and FAs were estimated using another set of data different to the test set.

The *Decision stage* module rejects the filler models output by the first level and accepts the keyword kw proposed by the first level if the Equation 4.16 is satisfied; otherwise it is rejected.

$$kw = K_{best} \text{ and } G_{FA} - G_{best} \geq \alpha \tag{4.16}$$

The first part in Equation 4.16 relies on the fact that a keyword corresponding to a FA will have a sequence of phones so different to the correct transcription of such keyword that the best matching between this sequence and the set of correct costs will produce a different keyword. However, for FAs which contain a very close sequence of phones to the correct transcription, the keyword proposed by the *Lexical access module* will be the same as the one proposed by the first level, and the use of the threshold α is more discriminative to discard them. In the same way, for hits, it is very likely that the keyword proposed from the set of correct costs matches the one proposed by the first level and the costs G_{FA} and G_{best} differ so much that the second part of the Equation 4.16 is also satisfied. This work has been published in [73].

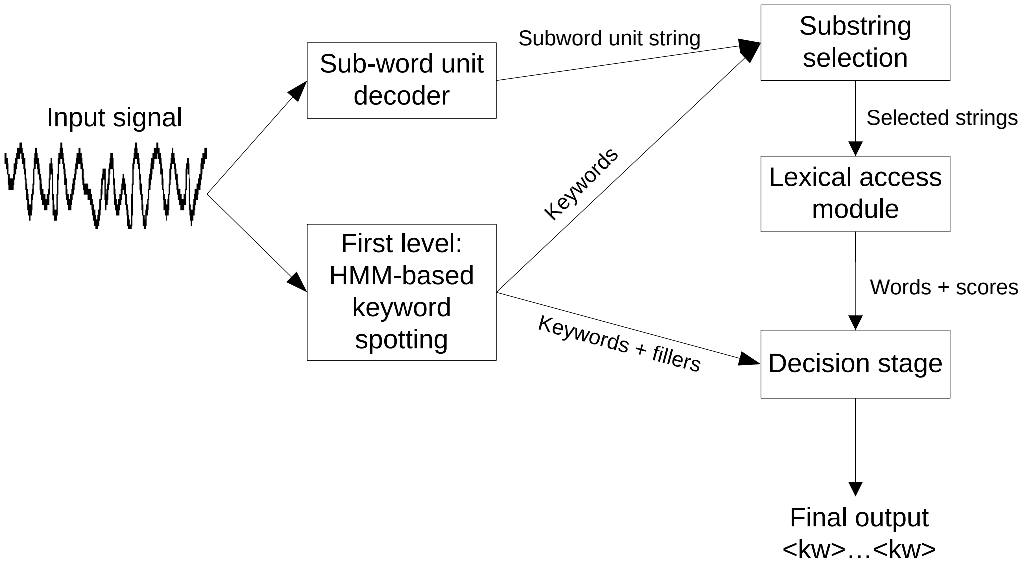

FIGURE 4.6: The system architecture for the *Lexical Access* confidence measure in the keyword spotting system. <kw> denotes each keyword in the final output of the system.

4.4 Experimental setup

4.4.1 Feature extraction

Firstly, the input signal is sampled at 16 Khz with 16 bits per sample and pre-emphasised and transformed into a sequence of frames, using a Hamming window (25 msec window size and 10 msec window shift), then characterised by 12 MFCCs plus energy and their first and second derivatives, giving 39 coefficients in total.

4.4.2 Acoustic modelling

Context-independent (CI) allophones (monophones) were used as keyword acoustic models and CI filler models explained in Section 4.2 as acoustic filler models throughout the CI experiments. All these models along with the beginning and end silence models had a conventional 3-state, left-to-right topology. There was an additional short pause model which had a single emitting state and a skip transition. The output distributions for each of these models consisted of 15-components GMM.

Context-dependent (CD) allophones (triphones) were used as keyword acoustic models and filler models for the CD experiments. They were cross-word and were state-clustered using HTK's standard decision tree method with phonetically-motivated questions, which leads to 5632 shared states. The output distributions for each of these models consisted of 8-components GMM.

4.4.3 Language modelling

Apart from the pseudo N-gram explained in Section 4.2, a bigram was used as LM in the *Sub-word unit decoder* module for the *Heuristic rules* and the *Lexical access* confidence measures.

4.4.4 Lexicon

A set of keywords was extracted from the geographical corpus in the Albayzin database based on their high frequency of occurrence in the development and test sets and suitability as search terms for geographical-domain information retrieval. A complete list of those keywords along with their number of occurrences for both sets of data are presented in Appendix D.

4.4.5 System tuning

The CI and CD acoustic models were trained on the ***phonetic training set***. The number of components GMM for each state for both sets of models was tuned for phone accuracy in the ***phonetic test set***. The bi-gram LM used in the phone-based speech recognition modules was built from the ***phonetic training set***.

Additionally to the HMM- and LM-training procedures, several parameters are substantial to be tuned for each architecture as follows:

In the first level, three parameters were necessary to be estimated: the *word insertion penalty (p)* and *language scale factor (s)* to be used in the decoding algorithm along with the value N in the pseudo N-gram used as LM in the first level. In case the CI experiments, the value N for the N-gram was tuned to achieve the best rate for each of the two metrics used in the evaluation independently on the ***geographic training set***. For the RA and FAR metric, a value of $N = 6$ was used for all of the filler models, which was chosen to get the desired performance in terms of RA and FAR (i.e., to achieve as many hits as possible with an acceptable FAR). For the FOM metric, a different value was achieved for each filler model as follows: $N = 6$ for the AM filler model, $N = 2$ for the PM filler model and $N = 1$ for the BM and APM filler models. The parameter p was set to be 0.0 to avoid deletion errors and the parameter s was tuned to achieve the desired performance in terms of RA and FAR on the ***geographic training set***. For the CD experiments, to take advantage of the HMM-based keyword spotting process and its capability of hypothesizing most of the keywords of the speech signal, all these parameters were tuned to achieve the best keyword detection rate, maintaining an acceptable FAR. In this case, the value N in the pseudo N-gram used in the first level was found to be $N = 12$.

Finally, the parameters necessary for each confidence measure in the second level were tuned as follows:

The *Exact Match* confidence measure does not need any additional parameter to be tuned.

The *Likelihood* confidence measure has two parameters, the thresholds X_{beam} and Y_{beam} below which the keywords proposed by the first level are finally accepted. Both were tuned on the ***geographic training set***.

The *Heuristic Rules* confidence measure needs the parameters p and s for the phone decoding, which were tuned for phone accuracy on the ***phonetic test set***. The parameters F_1, F_2 and F_3 in the *Decision stage* were tuned on the ***geographic training set***.

The *Lexical Access* confidence measure employs the same p and s values for the phone decoding module that the *Heuristic Rules* one. To train the confusion matrix to be used in the DP algorithm we have used the ***geographic training set***, used to estimate the threshold α as well.

The final evaluation for each confidence measure used the ***geographic test set***.

4.5 Results and discussion

We have divided the results presented for Keyword Spotting into two groups. In the first group, results achieved for all of the confidence measures for all of the filler models in the first level using the monophones as acoustic models for the keywords and CI filler models are presented. Later, we have chosen the best CI filler model in the first level (the AM model), we have trained its CD filler model and have evaluated the confidence measures using the triphone-based acoustic models for the keywords and these same models as filler models in the first level. In this way, it is expected that these last results improve the system performance presented by the CI experiments.

4.5.1 CI results

The *Recognition Accuracy* versus *False Acceptance Rate* and the FOM metric, defined in Appendix E, have been used to evaluate the CI results. Significance tests, by using paired t-tests, were run to show if the differences in the FOM value were significant across the keywords. The parameters necessary for each confidence measure were tuned for both metrics independently on the ***geographic training set***.

Table 4.1 presents the results in terms of *Recognition Accuracy* (RA) / *False Acceptance Rate* (FAR) for the ***geographic training set***. And Table 4.2 presents the results for such metric for the ***geographic test set***.

Table 4.3 and Table 4.4 present the results for the ***geographic training set*** and ***geographic test set*** respectively under the FOM metric.

	No CM	Exact Match	Likelihood	Heuristic Rules	Lexical Access
AM	77.7/29.9	76.2/29.8	76.2/24.1	71.6/16.1	76.0/11.8
PM	87.6/43.4	85.7/42.3	76.3/23.8	78.9/22.4	82.6/11.2
BM	97.1/67.6	94.3/67.4	75.7/38.0	84.3/38.6	82.0/13.1
APM	97.0/88.1	93.7/87.7	56.7/45.3	81.5/57.4	74.4/14.6

Table 4.1: Results in terms of RA/FAR for CI acoustic models for the confidence measures in the development set, i.e., the ***geographic training set***.

	No CM	Exact Match	Likelihood	Heuristic Rules	Lexical Access
AM	74.2/29.5	74.0/29.1	73.0/23.4	69.3/15.6	72.2/13.7
PM	84.9/42.8	84.5/41.7	75.5/23.1	77.5/21.6	77.2/13.2
BM	96.3/66.4	96.2/65.8	76.7/36.3	82.8/38.4	78.2/15.0
APM	95.4/87.4	95.3/86.8	56.9/43.3	80.7/56.6	71.0/15.3

Table 4.2: Results in terms of RA/FAR for CI acoustic models for the confidence measures in the test set, i.e., the ***geographic test set***.

Evaluation in terms of RA/FAR From Table 4.2 it is shown that for the first level in isolation (No CM), when the filler model is composed by the same acoustic models as the keywords less keywords are retrieved and therefore less hits and less FAs are output in using the AM filler model. For all of the confidence measures, both the number of hits and the number of FAs are decreased compared with the first level in isolation for all of the filler models. It is interesting to compare the rates achieved in using the BM and APM filler models. Such table shows that the BM filler model outperforms the number of keywords detected correctly while it minimizes the number of FAs compared with the APM filler model for all of the cases. It means that a single filler model to represent all of the phones is not discriminative at all in presenting keywords or that filler, due to almost all of the keywords hypothesized correspond to a FA. In comparing the confidence measures themselves, it is shown the better performance of the *Lexical Access* one when it is compared with the rest for all of the filler models, especially in the AM filler model, where the *RA* is almost the same for all of them, with a lot of reduction in the *FAR*, and even better in comparing with the *Heuristic Rule* one, with a smaller number of FAs. For the PM, BM and APM filler models, the *Lexical access* confidence measure presents a small *FAR* maintaining an acceptable *RA*. And when it is compared with the *Likelihood* one, both rates are improved for the PM, BM and APM filler models. For the AM filler model, the same effect can be deduced as a dramatical decrease in the *FAR* caused a minimum decrease in the *RA*. The best performance of the *Lexical Access* confidence measure is caused by the amount of information given to the whole system. In comparing every confidence measure across the filler models, we showed the following: For the *Exact Match* confidence measure, the BM filler model outperforms the rates got by the APM one. For the *Likelihood* and *Lexical access* confidence measures all of the filler models outperform the APM and the PM one outperforms the rates got by the AM. It is caused by a better keyword detection rate in the first level, which contributes to more hits can be retained after such confidence measure is applied. For the *Heuristic Rules*, the BM filler model outperforms the APM one. For the rest of the comparisons, an increase in the *RA* also caused an increase in the *FAR*. In the *Exact Match* confidence measure, due to most of the occurrences hypothesized by the first level are also hypothesized by the word-based decoding process run in such confidence measure, similar rates that of the first level are achieved for every filler model.

	No CM	Exact Match	Likelihood	Heuristic Rules	Lexical Access
AM	65.5	65.6	66.3	65.5	69.5
PM	60.9	62.3	64.9	63.2	72.6
BM	64.3	64.4	65.6	64.4	69.2
APM	15.7	16.6	42.9	35.4	65.0

Table 4.3: Results in terms of FOM for CI acoustic models for the confidence measures in the development set, i.e., the ***geographic training set***. Higher values indicate better performance.

	No CM	Exact Match	Likelihood	Heuristic Rules	Lexical Access
AM	64.2	64.2	65.5	64.2	67.5
PM	59.8	60.4	65.1	64.1	69.2
BM	61.5	61.6	63.8	61.6	67.1
APM	18.0	18.6	43.7	37.3	62.9

Table 4.4: Results in terms of FOM for CI acoustic models for the confidence measures in the test set, i.e., the ***geographic test set***. Higher values indicate better performance.

Evaluation in terms of FOM The FOM metric gives a single value for all of the filler models according to the confidence measures. In Table 4.4 it is shown that the *Lexical Access* confidence measure outperforms the rest for every filler model. Again, it is caused by the amount of information given to such system. It must be also noted that, although the *Likelihood* confidence measure is worse in terms of *RA* and *FAR* than the *Heuristic Rule* one, the FOM value is better for the *Likelihood* one for the PM filler model. It means that both metrics follow different patterns in evaluating a same system. Actually, in the FOM computation, the putative keywords with a very low score are not taken into account in the final value. Contrary, in the *RA/FAR* values, all of the putative keywords contribute equally to the final values. All of the confidence measures outperform the final rate for all of the filler models, except the *Exact Match* and *Heuristic Rules* ones for the AM filler model. It is a consequence of two factors: First, the better performance of the AM filler model against the rest of the filler models, which leads to a lower margin of improvement. Second, although the FAs are reduced from the first level, some hits are also removed in presenting the final output of the system, which causes this same FOM value. In addition to this, the low score of some FAs, which are missed in the final metric computation, contributes to such effect. The AM filler model presents the best final performance for all of the confidence measures except for the *Lexical Access* one, where the PM filler model performs the best. This is caused by the amount of information provided to such confidence measure along with the worse system performance of the first level in isolation, which causes that the margin of improvement is greater for such filler model. The higher keyword detection rate of the PM filler model in the first level also contributed to such effect. Since the PM filler model presents a worse value than the AM and BM filler models for the first level (*No CM*), when the *Likelihood*, *Heuristic Rules* and *Lexical access* confidence measures are run, it produces a better value than the BM filler model. Such effect is not observed in the comparison between the AM and BM filler models, due to the difference between the BM and AM performance in the first level is greater than the one between the PM and BM filler models. However, the *Exact Match* confidence measure does not follow this pattern, due to the weakness of it. For the APM filler model, and due to its much worse system performance in the first level, even after applying any confidence measure, such filler model does not outperform any other. Paired t-tests were used to evaluate if the differences presented in the FOM value for every filler model across each confidence measure are significant or are not. For the AM filler model, there is only significant difference between the *Lexical access* confidence measure and the *Exact Match* one and between the *Likelihood* and the *Exact Match* with $p < 0.05$. For the PM filler model, the difference was statistically significant between the *Heuristic Rules* and the *Exact Match* with $p < 0.02$, between *Lexical access* and *Exact Match* with $p < 0.05$ and between *Likelihood* and *Exact Match* with $p < 0.04$. For the BM filler model, the difference was significant between *Lexical access* and *Exact Match* with $p < 0.03$ and between *Lexical access* and *Likelihood* with $p < 0.001$. For the APM filler model, all of the differences between the confidence measures were found to be significant with $p < 0.001$ except the one between *Heuristic Rules* and *Likelihood*. Paired t-tests were also used to compare if the improvements in using the confidence measures were significant against the *No CM* and showed the following results: There is no difference in the confidence measures for the AM filler model compared with the *No CM*. For the PM filler model, the difference was found to be significant for the *Heuristic Rules* confidence measure with $p < 0.04$. For the BM filler model, no difference was found to be significant. Finally, for the APM filler model, all of the differences in the confidence measures were found to be statistically significant with $p < 0.001$.

In comparing every confidence measure across each filler model, the paired t-tests showed the following: When *No CM* is applied, there is significant difference between the AM filler model and PM filler model with $p < 0.04$, between AM and APM with $p < 0.001$, between PM and APM with $p < 0.001$ and between BM and APM with $p < 0.001$. For the *Exact Match*, *Likelihood* and *Heuristic Rules* confidence measures, the difference was found to be significant between AM and APM, between PM and APM and between BM and APM with $p < 0.001$. Finally, for the *Lexical access* confidence measure, the difference was found to be significant for AM and BM, AM and APM, PM and BM and BM and APM with $p < 0.001$.

4.5.2 CD results

The CD allophones were used for both the keywords and the filler models due to the best system performance of the CI allophones in terms of FOM for the first level in isolation. Here, we have tried to evaluate the best confidence measure over the best acoustic model configuration. In this case we have used the same two metrics as for the CI results. The aim of this experiment is to achieve a very high detection rate (retrieve almost all of the hits in the first level) and evaluate the FOM value for such *RA* and the resulting *FAR*. Significance tests, as in the CI experiments, showed if the differences in using each confidence measure are significant.

	FOM	RA/FAR	Reduction in FAR
No CM	70.4	84.3/29.0	–
Exact Match	70.4	84.2/28.3	3.3%
Likelihood	70.9	83.4/24.0	23.3%
Heuristic Rules	70.4	83.8/26.8	10.5%
Lexical Access	72.5	83.4/17.4	48.9%

Table 4.5: Results in terms of FOM, RA/FAR and the reduction of FAR compared with No CM for CD acoustic models for the confidence measures in the development set, i.e., the ***geographic training set***.

	FOM	RA/FAR	Reduction in FAR
No CM	68.3	80.0/25.7	–
Exact Match	68.4	79.9/25.1	2.2%
Likelihood	68.6	79.6/21.1	22.9%
Heuristic Rules	68.2	79.7/24.3	7.8%
Lexical Access	69.4	79.1/16.6	43.2%

Table 4.6: Results in terms of FOM, RA/FAR and the reduction of FAR compared with No CM for CD acoustic models for the confidence measures in the test set, i.e., the ***geographic test set***.

Evaluation in terms of FOM From Table 4.6 it it shown that, according to the FOM value, all of the confidence measures outperform the first level in isolation except the *Heuristic Rules* one. It is due to the rules defined for such confidence measure. Although they rejected some FAs of the first level in isolation, some of the hits are also rejected in tuning such confidence measure for *RA/FAR*, giving a slight worse FOM value. Such reduction is confirmed in the *RA* and *FAR* values. However, for the rest of the confidence measures, the number of FAs is reduced in such a way that the loss of some hits is not important in presenting the FOM value. It is confirmed by the relationship between *RA* and *FAR*. The *Likelihood* and the *Lexical Access* confidence measures get a high FAs reduction, compared with the number of hits that are missed, causing the two best FOM values. The *Exact Match* only increases slightly the FOM value due to both the number of hits and FAs are reduced in more or less the same percentage. Paired t-tests were also used to evaluate the improvement between these confidence measures in terms of FOM. In comparing the confidence measures, it was shown that there is significant improvement between the *Likelihood* and the *Heuristic Rules*, between the *Lexical access* and the *Heuristic Rules*, between the *Lexical access* and the *Exact Match* and between the *Likelihood* and *Exact Match* with $p < 0.05$. In comparing with *No CM* the difference was found to be significant for the *Likelihood* confidence measure with $p < 0.03$ and for the *Lexical access* one with $p < 0.04$.

Evaluation in terms of RA/FAR The strongest results are discussed when we compare the percentage of FAs that is reduced for each confidence measure. It is shown that, for a very small decrease in the number of hits (about 1% relative as much in case the *Lexical access* confidence measure), the number of FAs decreases considerably. In this last case, the number of FAs decreases up to 43% relative. It is also interesting the case of the *Likelihood* confidence measure. There, in decreasing the RA in a 0.5% relative, the number of FAs is decreased at about 23% relative. And comparing the *Likelihood* and the *Lexical access* confidence measures, in reducing the number of hits in a 0.6% relative in the latter, the number of FAs is reduced in a 26.3% relative.

4.5.3 Keyword-length based analysis

It is accepted that the final performance in keyword spotting systems depends a lot on the length of the keywords chosen. In this way, with short keywords, more FAs will be generated in the system and with long keywords less FAs will be. It relies on the fact that short keywords may correspond to a subpart of a long word or even can be formed as the join of two different words of the speech data. Contrary, these effects are less likely to occur for long keywords. To represent how the length of the words affects the performance of the first level and each of the confidence measures, the keywords are divided into the following groups: The *Short-lengh Keywords Group (SKG)* contains the keywords with 4, 5 and 6 phones. The *Medium-length Keywords Group (MKG)* contains the keywords between 7 and 9 phones. Finally, the *Long-length Keywords Group (LKG)* groups the keywords with more than 9 phones. Table 4.7 shows the FOM value and Table 4.8 shows the RA and FAR for each of these three groups. These data correspond to the CD experiments.

	SKG	MKG	LKG
No CM	44.6	62.1	84.7
Exact Match	44.8	62.3	84.6
Likelihood	45.4	62.7	84.6
Heuristic Rules	44.9	61.9	84.4
Lexical Access	46.8	63.1	85.4

Table 4.7: Results in terms of FOM for the *SKG*, *MKG* and *LKG* in the test set, i.e., the ***geographic test set***. Higher values indicate better performance.

	SKG	MKG	LKG
No CM	51.3/62.2	80.5/25.2	92.0/3.8
Exact Match	51.3/61.4	80.5/24.7	91.9/3.8
Likelihood	51.0/52.6	79.8/22.9	91.9/3.8
Heuristic Rules	51.3/59.2	79.9/24.7	91.7/3.8
Lexical Access	50.7/36.7	78.9/22.4	91.6/3.1

Table 4.8: Results in terms of *RA/FAR* for the SKG, MKG and LKG in the test set, i.e., the ***geographic test set***

Evaluation in terms of FOM From the individual FOM value of each group it is shown that, as expected, such value improves from short-length keywords to long-length keywords. It is also shown that the use of the confidence measures gives more benefit to short-length keywords due to their worse performance in the *No CM* system, which allows to a greater margin of improvement. The *Heuristic Rules* confidence measure achieves a worse FOM value than the *No CM* for medium- and long-length keywords, which follows the same pattern as the final FOM value explained in the CD experiments. Instead, for short-length keywords, the FOM value is better, which suggests that such confidence measure can be successfully applied on keywords with less than 7 phones.

Evaluation in terms of RA/FAR A similar pattern as that of the FOM metric was found for *RA* and *FAR* values across each group. It is shown that short-length keywords tend to produce a worse RA and FAR than medium- and long-length keywords. These last keywords present the best performance as well. Again, the *Lexical access* confidence measure gives the greatest benefit for all of the groups independently. With it, it is shown that for a very small decrease in the RA value, the FAR is decreased dramatically for each group. Such decrease is higher for the short-length keywords and lower for the long-length keywords due to these have a better performance in the *No CM* case.

4.6 Conclusions

From the CI results, we have shown that all of the confidence measures outperform the final FOM value proposed by the first level in isolation for all of the filler models, except for the Allophone Models (AM). Due to the AM system performance is the best in the first level in isolation, the *Likelihood* and *Lexical access* confidence measures are the only that outperform the final FOM value. For such reason, we have selected the AM filler model as the best CI filler model. As more information is provided to the system, better results are achieved. Therefore, the *Likelihood* confidence measure, which makes use of more information that the *Exact Match* one, outperforms its rates achieved for every filler model. Due to the *Lexical access* confidence measure is the one with the greatest amount of information, the FOM value is the best for every filler model. The *Likelihood* confidence measure outperforms the *Heuristic Rules* one under the FOM metric because a word-based speech recognition instead of a phone-based decoding is run. In addition to this, the likelihood computed during the word-based decoding is more discriminative than the heuristic rules defined in the latter confidence measure in terms of FOM. However, such last improvement was not found to be significant. Contrary, the phone-based decoding in the *Lexical access* confidence measure with the confusion matrix and its decision computation which makes use of the hits and FAs involved in a previous training stage gives greater benefit than the likelihood computed from the word-based decoding, although it was just found to be significant for the Broad class Models (BM) and Average Phoneme Model (APM) filler models. In the same way, the missing of the low-score occurrences, along with the great dependency of the score in the final FOM value caused that for some confidence measures and filler models, the comparison of the confidence measures and the first level in isolation was not found to be significant. In terms of *RA* and *FAR*, it can be also concluded that the *Lexical access* confidence measure performs the best and the *Exact Match* one performs the worst. However, in comparing the *Likelihood* and the *Heuristic Rules* ones, a similar tradeoff between *RA* and *FAR* is observed for the AM, BM and APM filler models, and, contrary to the FOM value, better performance for the *Heuristic Rules* one for the Phoneme Models (PM) filler model. The different computation used by the two metrics caused this last effect.

From the CD results, the *Lexical access* confidence measure presents again the best performance of the confidence measures presented for Keyword Spotting. It is caused by the amount of information given to such system. In comparing it with the approach based on the standard likelihood computation (i.e., the *Likelihood* confidence measure), the final FOM rate is better, although the improvement was not found to be significant. Such conclusion is stronger in comparing the number of hits and FAs that are presented in each confidence measure: In reducing the number of hits in a very small percentage (about 1% relative as much) from the *No CM*, all of the confidence measures decrease the number of FAs in a greater percentage. Powerful results are achieved with the *Likelihood* and *Lexical access* confidence measures, where the *FAR* is decreased up to 43% relative. This last confidence measure achieved a 26.3% relative reduction in the *FAR* with a very slight reduction in the *RA* when it is compared with the *Likelihood* one, which makes use of the standard likelihood got from the word-based speech recognition for each keyword.

Based on the number of phones of the keywords, we can conclude that short keywords are more likely to produce more FAs, due to their inherent more probability to be confused with parts of other long words or even with some segments of two adjacent words.

4.7 Summary

We have presented in this chapter the approaches developed for Keyword Spotting. All of the approaches presented consisted of a same framework: In a first level, a state-of-the-art HMM-based keyword spotting process is used to retrieve a set of keywords according to the input signal. The contributions are presented in the way of four different confidence measures in the second level to improve the rates achieved by the first level. Two of them presented an isolated word speech recognition process which computed the parameters used in the decision stage to accept or reject the putative keywords. The two others made use of a phone-based decoding process along with additional modules to compute the parameters used in the decision stage. This chapter also described the experiments on the ALBAYZIN database. The goal was to compare the four confidence measures presented to form a keyword spotting system. We have shown that the *Lexical access* confidence measure, making use of a phone-based decoding and a DP algorithm which computes a cost for each keyword from a confusion matrix built from the hits and FAs got in the training stage, presents the best rates for the keyword spotting system. It is due to the great amount of information given to it, along with the knowledge and use of the errors appearing in the first level. When it is compared with the standard confidence measure based on the likelihood computed during the word-based speech decoding, the final FOM value is better (although it was not found to be significant) and the *FAR* is reduced in a 26.3% relative maintaining a similar *RA*. We have also reported an analysis of the performance of such confidence measures depending on the number of phones of each keyword. Short-length keywords tend to produce more errors than medium- and long-length ones and the use of the confidence measures gives more benefit to them.

Chapter 5

Contributions to Spoken Term Detection

5.1 Introduction

STD, as Keyword Spotting does, deals with the search of a set of keywords within the audio content. However, contrary to Keyword Spotting, and following the NIST recommendations in terms of speed and accuracy for STD systems, the list of keywords is unknown during the decoding process. It causes that the decoding process must be performed by means of sub-word units (phones, graphemes, syllables...) to build an efficient index. A second step hypothesizes the final list of terms from that index.

This chapter is divided into three different parts. The first part presents a state-of-the-art STD approach used in the other two parts. The second one presents a new technique to estimate the confidence score of the final list of terms in the STD approach along with the results achieved for the ALBAYZIN database. It also compares this technique with the standard confidence scoring techniques proposed in the literature and reports the main conclusions derived from such technique. The third part reports preliminary experiments in information retrieval over an English meetings domain and presents a confidence measure based on decision trees to outperform the performance of the STD system. It also reports the results achieved over the meetings domain and the main conclusions got from such confidence measure. Therefore, the contributions in this chapter rely on the new confidence measurement-based technique presented in the second part and the decision tree-based confidence measure in the third part.

5.2 STD approach

The STD approach used in our work is an implementation provided by the Brno University of Technology [39]. Although it is not the main objective of this chapter, and this approach is completely defined in the next chapter, it consists of a decoding process which produces a lattice of phones in the first step and a method based on a recursive match algorithm to find all the fragments in the lattice that exactly match the actual phone transcription of any search term in a second step (lattice search tool).

5.3 Confidence scoring

5.3.1 Motivation and prior research

Confidence scoring plays a very important role in STD systems. As more accurate the confidence score is, a subsequent threshold-based technique will produce a better STD performance in rejecting FAs.

Most of the confidence measures presented in the literature make use of the posterior probability (posteriors) estimated by the Viterbi algorithm during the decoding process not only for LVCSR and keyword spotting systems [37, 52, 68, 74], but also for STD systems [39, 66]. In these cases, a final normalization of such posteriors is necessary in order to present a final score for the hypothesized list of terms. Such posteriors are estimated directly from the HMMs (trained using the maximum likelihood criterion) used during the decoding step. These HMMs are based on a Bayesian approach that presents two main drawbacks: (1) the likelihood is estimated from a generative model (HMM with the maximum likelihood criterion), which assumes a framewise and component-wise independence of the acoustic features, and a finite number of GMM; (2) in the STD approach presented before, the confidence score calculation is expensive and needs the whole lattice to be computed.

On the other hand, it is well known that an MLP can be used to compute the class posterior probabilities for a classification task. This MLP is built from a standard 3-layer network with softmax output activation. In addition to this, MLPs have been also widely used for speech recognition, by computing the posterior probabilities for phone classes from the acoustic vectors [10]. Silaghi and Bourlard [43] computed the confidence score for each keyword in their keyword spotting system without filler models by using the local posteriors computed in the training of an MLP.

Here, we propose such method to compute the confidence score on the STD approach explained before and compare it with equivalent HMM-based techniques. In this way, this method is not used directly during the Viterbi-based decoding as Silaghi and Bourlard did. Contrary, it is used in a following step after the Viterbi-based decoding when the phone lattices have been already computed. In addition to this, the LM component stored in the lattice was used in the final confidence score computation. Instead, Silaghi and Bourlard did not make use of any LM in their work. This method does not make any assumption of the acoustic features and does not need the whole lattice to be computed. It has been implemented by collaborators in the CSTR group in the University of Edinburgh and in this thesis, we have evaluated it on Spanish language. This work has been published in [75].

5.3.2 HMM-based confidence scoring

In STD systems, the posterior probability $p(K_{t_1}^{t_2}|O^T)$ is the confidence with which the term K appears between the frames t_1 and t_2. According to the Bayesian formulation, it can be expressed as follows:

$$\begin{aligned} p(K_{t_1}^{t_2}|O^T) &= \sum_{\alpha,\beta} p(K_\alpha, K_{t_1}^{t_2}, K_\beta|O^T) && (5.1) \\ &= \sum_{\alpha,\beta} \frac{p(K_\alpha, K_{t_1}^{t_2}, K_\beta, O^T)}{p(O^T)} && (5.2) \\ &= \sum_{C_K} \frac{p(O^T|C_K, K_{t_1}^{t_2})p(C_K, K_{t_1}^{t_2})}{p(O^T)} && (5.3) \end{aligned}$$

where $K_{t_1}^{t_2}$ is the search term from frame t_1 to frame t_2. K_α and K_β are any possible phone strings before and after $K_{t_1}^{t_2}$, with K_α starting at frame 1 and K_β ending at frame T. C_k groups K_α and K_β in Equation 5.3 and represents the left and right context of $K_{t_1}^{t_2}$ respectively.

In Equation 5.3, the conditional probability $p(O^T|C_k, K_{t_1}^{t_2})$ is the acoustic likelihood, and the joint probability $p(C_K, K_{t_1}^{t_2})$ is given by the LM. The denominator $p(O^T)$ is considered to be a constant. Since the Baum-Welch algorithm usually computes $p(O^T|C_k, K_{t_1}^{t_2})$, we denote this confidence measurement as *Baum-Welch* confidence. For such confidence, three partial paths are computed: The first one, denoted as $L_{begin}(K)$ computes all the subpaths that reach the node start of the keyword from the beginning of the lattice to the node start of the keyword. The second one, denoted as $L_{end}(K)$ computes all the subpaths that leave the node end of the keyword from the node end of the keyword to the end of the lattice. Such subpaths are computed using standard forward-backward processes [9] respectively. The third path, denoted as $L(K)$ is the path that contains the keyword. Finally, all the subpaths (those that finish in a terminal node) contained in the lattice, i.e., the whole evidence of the lattice, denoted as L_{all} are also computed. In this last case, only the forward step is required. The confidence score for the partial paths $L_{begin}(K)$ and $L_{end}(K)$ and for the whole path L_{all} were computed as a sum of each subpath score. We denote them as $Sbw_{begin}(K)$, $Sbw_{end}(K)$ and $Sbw_{all}(K)$ respectively in Equation 5.4. Each subpath score was computed as the sum of the acoustic log likelihood plus the LM log likelihood (weighted by the language scale factor) plus the word insertion penalty of its sequence of phones. The confidence score for the path $L(K)$, denoted as $Sbw_{kw}(K)$ in Equation 5.4, was computed in the same way. Therefore, the final confidence score for the keyword K is computed as follows:

$$Sbw(K) = Sbw_{begin}(K) + Sbw_{end}(K) + Sbw_{kw}(K) - Sbw_{all}(K) \tag{5.4}$$

A significant reduction in computational cost can be achieved by replacing the sum over all C_K in Equation 5.3 with the 1-best path, as in Equation 5.5:

$$p(K_{t_1}^{t_2}|O^T) \approx \frac{max_{C_K}\, p(O^T|C_K, K_{t_1}^{t_2})p(C_K, K_{t_1}^{t_2})}{max_{K_1^T} p(O^T|K_1^T)p(K_1^T)} \tag{5.5}$$

Such approximate confidence value is also computed from standard forward-backward processes [9], but in this case, just the best path is considered, and not all the subpaths as in the *Baum-Welch* confidence. Therefore, we denote it as *Viterbi* confidence. The final confidence score is then computed as in the *Baum-Welch* confidence. Equation 5.6 represents the final confidence score for each keyword K for the *Viterbi* confidence.

$$SViterbi(K) = SViterbi_a(K) + SViterbi_b(K) + SViterbi_{kw}(K) - SViterbi_{best}(K) \tag{5.6}$$

where $SViterbi_a(K)$ denotes the score of the best path from the beginning of the lattice to the node start of the keyword, $SViterbi_b(K)$ denotes the score of the best path from the node end of the keyword to the end of the lattice, $SViterbi_{best}(K)$ denotes the score of the best path of the whole lattice and $SViterbi_{kw}(K)$ denotes the score of the path that contains the keyword. Such subpath scores are computed in the same way as in the *Baum-Welch* confidence.

In addition to these two standard HMM-based methods to compute the confidence score, in running the lattice search tool in the second step of the STD approach, it may occur that two occurrences of the same keyword overlap in time (i.e occ_1 may appear between T_1 and T_3 and occ_2 may appear between T_2 and T_4, with $T_1 < T_2 < T_3 < T_4$). Therefore, we have analyzed two different methods to compute the final score, which were implemented in the lattice search tool by the Brno University of Technology [39]. The first one, denoted as *Best Time Best Score (BTBS)* method, selects the occurrence of the keyword which has the best score of all those overlapped. In this case, the confidence score is the one computed for such occurrence. The second one, denoted as *Group Time Group Score (GTGS)* method, selects the occurrence of the keyword with the less starting point. It computes the confidence score as the sum of all of the confidence scores of those occurrences overlapped. As it was found no difference in the system performance in using any kind of these two methods for the *Viterbi* confidence, we have chosen the *GTGS* method for the *Baum-Welch* confidence and for the MLP-based approaches.

5.3.3 MLP-based confidence scoring

The posterior probability for each frame t, denoted as $p(Q_t|O)$, is computed by an MLP. Q_t represents the phone class of the search term at frame t and O represents the sequence of the vectors in the whole observation. Q_t is computed from the lattice produced by the phone-based decoding process. The MLP configuration is depicted in Figure 5.1.

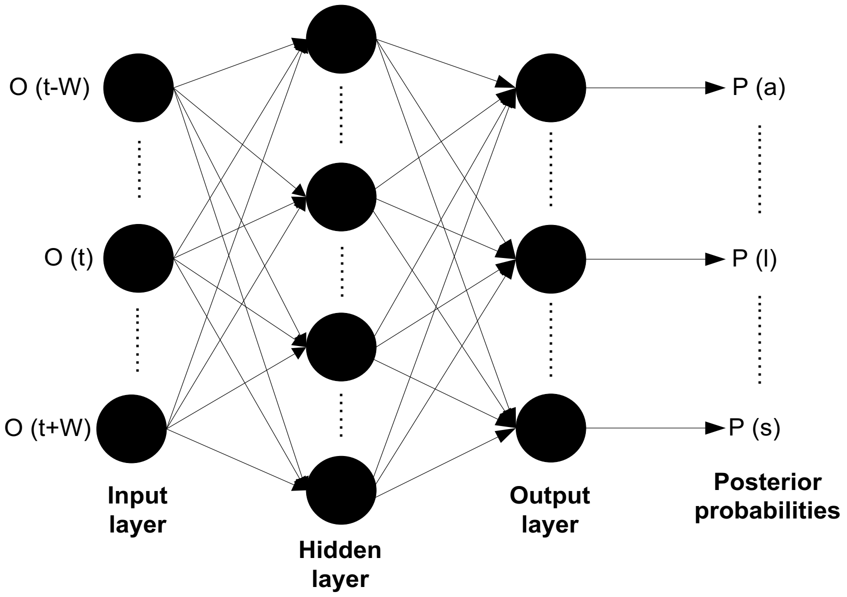

FIGURE 5.1: The MLP network for the MLP-based confidence scoring.

The confidence score for each search term is computed by adding the frame confidence (i.e., the frame-level posterior probability $p(Q_t|O)$ estimated by the MLP), by taking logarithms in Equations 5.7 and 5.8. Therefore this confidence score is independent of the context C_k and it only takes into account acoustic properties. The MLP input layer consists of a window of 2W + 1 frames of acoustic features and we have chosen $W = 4$ in our experiments to form a 9-frame input window. The output layer consists of 47 phones plus a short silence and a beginning and end silence (i.e., 50 units). The hidden layer contains 1100 units and the input layer contains 351 units.

$$
\begin{aligned}
p(K_{t_1}^{t_2}|O^T) &= \prod_{t=t_1}^{t_2} p(Q_t|O^T) && (5.7) \\
&= \prod_{t=t_1}^{t_2} p(Q_t|o_{t-W}, ..., o_t, ..., o_{t+W}) && (5.8)
\end{aligned}
$$

This phone-independency assumption means that linguistic information stored in the lattice is not used. To solve it, dependency between phones has been added as illustrated in Figure 5.2. Two different implementations have been used to model such dependency: *Direct LM integration* and *Baum-Welch LM integration*. In both cases, the phone dependency is modeled by a bigram LM.

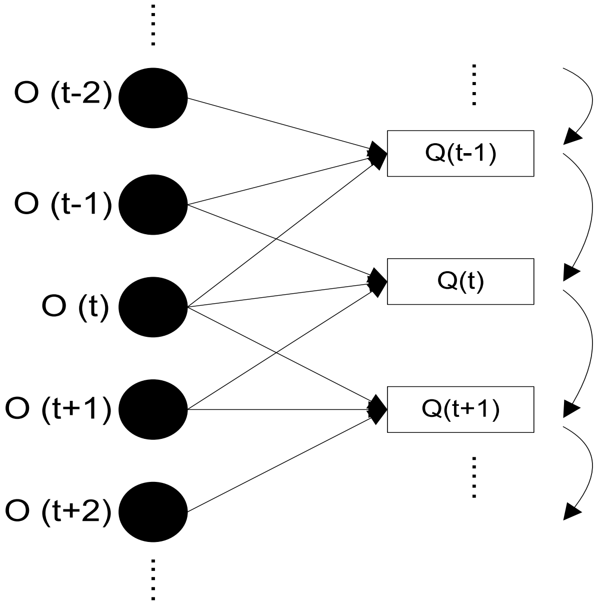

FIGURE 5.2: The phone-dependency representation for the posterior confidence computation. $Q(t)$ is the phone at frame t and $O(t)$ is the acoustic observation at time t.

5.3.3.1 Direct LM integration

In adding the linguistic component to the confidence score computation, we define the variable K^l as the search term K in the word layer. Thus, K^l represents the linguistic layer of the word K. Therefore, the confidence score for a word K is composed of two different units, one is the *acoustic* unit and the another is the *linguistic* unit. The former assigns probabilities from speech features, while the latter gets probabilities from the LM. We assume that $p(K^l, K) = p(K)$, i.e., K^l represents the K's phonetic form of the term K. We also consider that K^l is independent of the acoustic observation O, i.e, $p(K_l|K, O) = p(K_l|K)$. Therefore we can compute the best context C'_{K^l} which has the most of the probability of the accumulated linguistic score, i.e., $\sum_{C_{K^l}} p(K^l|C_{K^l})p(C_{K^l}) \approx p(K^l|C'_{K^l})$. In this way, the posterior probability of the word detected is computed as the product of the acoustic and LM scores as illustrated in Equations 5.9-5.12.

$$
\begin{aligned}
p(K, K^l|O) &= \sum_{C_{K^l}} p(K, K^l, C^l_K|O) && (5.9)\\
&= p(K|O) \sum_{C_{K^l}} p(C_{K^l}, K^l|O, K) && (5.10)\\
&= p(K|O) \frac{\sum_{C_{K^l}} p(K|C_{K^l})p(C_{K^l})}{p(K)} && (5.11)\\
&\approx p(K|O) \frac{p(K^l|C'_{K^l})}{p(K)} && (5.12)
\end{aligned}
$$

In this case, the LM score $p(K^l|C'_{K^l})$ is stored in the lattice and the denominator was considered to be a constant. $p(K|O)$ is given by the acoustic likelihood. Therefore the confidence score computation process is the same as in the phone-independent case. We refer to this confidence score computation as *Posterior with Direct LM integration.* To compute such confidence for each keyword, we simply add the acoustic log likelihood plus the LM log likelihood weighted by the language scale factor plus the word insertion penalty of its sequence of phones.

5.3.3.2 Baum-Welch LM integration

Contrary to the *Direct LM integration*, where just the path containing the term is considered for the final confidence score computation, this implementation regards the whole phone lattice to compute the score. Therefore, the confidence score computation is also considered as a two-step process. In the first step, only *acoustic* confidence is considered. In the second step, the *linguistic* confidence is considered by assuming all the phone alternatives that are stored in the lattice. Therefore, the final confidence is computed as the product of the acoustic and linguistic posterior, as illustrated in Equations 5.13-5.15, where L is defined as the whole phone lattice.

$$\begin{aligned} p(K, K^l|O) &= p(K|O)p(K^l|L) && (5.13) \\ &= p(K|O)\frac{p(K^l, L)}{p(L)} && (5.14) \\ &= p(K|O)\frac{\sum_{C_{K^l}} p(K^l, C_{K^l})}{p(L)} && (5.15) \end{aligned}$$

In this implementation, $p(K^l|L)$ only takes into account linguistic constraints. It must be also noted that this confidence is represented by a *global* score and therefore a forward-backward computation is required. That is why we denote $p(K^l|L)$ as a Baum-Welch LM confidence. We refer to this confidence score computation as *Posterior with Baum-Welch LM integration.* To compute the final confidence score for each keyword, we compute all the subpaths that reach the node start of the keyword in the partial path from the beginning of the lattice to the node start of the keyword, denoted as $Lbwlm_{begin}(K)$, all the subpaths that leave the node end of the keyword in the partial path from the node end of the keyword to the end of the lattice, denoted as $Lbwlm_{end}(K)$ and all the subpaths (those that reach a terminal node) contained in the lattice, i.e, the whole evidence of the lattice, denoted as $Lbwlm_{all}(K)$. Again, the subpaths of the partial paths $Lbwlm_{begin}(K)$ and $Lbwlm_{end}(K)$ were computed from standard forward-backward processes [9] respectively. And the whole path $Lbwlm_{all}(K)$ was computed from the forward step as well. Therefore the confidence score for each partial path $Lbwlm_{begin}(K)$, $Lbwlm_{end}(K)$ and the whole path $Lbwlm_{all}(K)$ is computed as a sum of each subpath score. We denote them as $Sbwlm_{begin}(K)$, $Sbwlm_{end}(K)$ and $Sbwlm_{all}(K)$ respectively in Equation 5.16. However, contrary to the *Baum-Welch* confidence, such score is computed only from the LM component stored in the lattice. Therefore, each subpath score is computed as a sum of the LM log likelihood weighted by the language scale factor plus the word insertion penalty of its sequence of phones. The score for the path that contains the keyword is computed in the same way as the *Direct LM integration.* We denote it as $Sbwlm_{kw}(K)$ in Equation 5.16. Therefore, the final confidence score for a keyword K is computed as follows:

$$Sbwlm(K) = Sbwlm_{begin}(K) + Sbwlm_{end}(K) + Sbwlm_{kw}(K) - Sbwlm_{all}(K) \tag{5.16}$$

5.3.4 Experimental setup

5.3.4.1 Feature extraction

Standard 12 MFCCs plus energy and their first and second derivatives were extracted from the input signal. Standard 12 PLPs plus energy and their first and second derivatives were used as MLP input features. The signal is sampled at 16kHz and stored with 16 bits precision. MFCCs and PLPs were computed at 10ms intervals within 25ms Hamming windows.

5.3.4.2 Acoustic Modelling

The same CD acoustic units as in the keyword spotting experiments in Chapter 4 were used as acoustic models for these experiments.

5.3.4.3 Language Modelling

A bigram trained from the ***phonetic training set*** was used for the lattice-based decoding in the STD approach.

5.3.4.4 System tuning

The lattices used for the HMM- and MLP-based confidence scoring approaches were fixed in such a way that p and s parameters were tuned for FOM metric from the *Viterbi-BTBS*. For the lattice search tool both parameters are tuned again for both HMM- and MLP-based confidence scoring approaches for the metrics used in the evaluation independently. The ***geographic training set*** was used for both tunings.

The ***phonetic training set*** has been used to train the MLP for the MLP-based approaches. It contains 4400 sentences. 4000 of them were used to train the MLP and the rest, i.e., 400, have been used to tune the number of units in the hidden layer, the learning factor and the number of epocs in the MLP training, based on cross-validation accuracy, giving a 70.6% of accuracy. The MLP was trained using the Quicknet software [20].

5.3.5 Results and discussion

The Albayzin database was used for the experiments. The same set of keywords listed in Appendix D and used for the keyword spotting experiments in Chapter 4 was selected from the geographical domain. The FOM and ATWV metrics along with the DET curves were used to evaluate the different confidence scoring approaches. Paired t-tests showed if the differences across the confidence scoring approaches were found to be significant.

	FOM	ATWV
Viterbi-BTBS	50.4	0.25
Viterbi-GTGS	50.4	0.24
Baum-Welch	50.4	0.24
Posterior with Direct LM integration	51.1	0.35
Posterior with Baum-Welch LM integration	50.1	0.26

Table 5.1: Results in terms of FOM and ATWV for the CD phone models for the confidence scoring approaches for the ***geographic training set***. For all measures, higher values indicate better performance.

	FOM	ATWV
Viterbi-BTBS	47.2	0.18
Viterbi-GTGS	47.3	0.18
Baum-Welch	47.3	0.18
Posterior with Direct LM integration	47.5	0.26
Posterior with Baum-Welch LM integration	46.6	0.15

Table 5.2: Results in terms of FOM and ATWV for the CD phone models for the confidence scoring approaches for the ***geographic test set***. For all measures, higher values indicate better performance.

Evaluation in terms of FOM Table 5.2 shows that for evaluation in terms of FOM, as expected, the *Posterior with Direct LM integration* confidence scoring, making use of a discriminative approach in calculating the score, achieves the best rate. However, paired t-tests showed that such value is just statistically significant compared with the *Posterior with Baum-Welch LM integration* confidence scoring. These tests did not show any additional difference between the rest of the confidence scoring approaches. Thus, no significant difference was found between HMM- and MLP-based approaches. On the other hand, the *Posterior with Baum-Welch LM integration* achieves the worst rate, due to the weak LM (bigram) used in the experiments.

Evaluation in terms of ATWV Table 5.2 also shows that for evaluation in terms of ATWV, the *Posterior with Direct LM integration* outperforms any other confidence scoring approach and that the addition of the *Baum-Welch LM integration* in the posterior computation does not contribute to improve the final system performance. It must be noted that although the *Posterior with Baum-Welch LM integration* outperforms slightly the HMM-based confidence scoring approaches for the ***geographic training set***, when the parameters tuned are applied over the ***geographic test set***, such benefit is unreliable. The DET curve in Figure 5.3 shows that although the final absolute ATWV value presented with the *Posterior with Baum-Welch LM integration* is worse than the HMM-based approaches, it oupterforms all of the HMM-based confidence scoring approaches for much of the range. And the *Posterior with Direct LM integration* confidence scoring presents the best DET curve for all the different operating points.

5.3.6 Conclusions

Discriminative approaches, such as NN-based approaches, typically outperform equivalent generative models-based approaches, such as HMMs trained from the maximum likelihood criterion, when both are used for the same task (e.g. ASR task). Here, we have shown that the use of an MLP to compute the posterior probability of each keyword did improve the HMM-based posterior probability for a Spanish STD system. Such improvement is a combination of two different things: (1) the acoustic confidence is a local score, which is very dependent on the current frame and its neighbours; these neighbours are highly correlated. The MLP is able to model this frame-wise dependency, contrary to those HMMs. (2) If enough data are provided to the MLP training, such MLP structure can represent efficiently

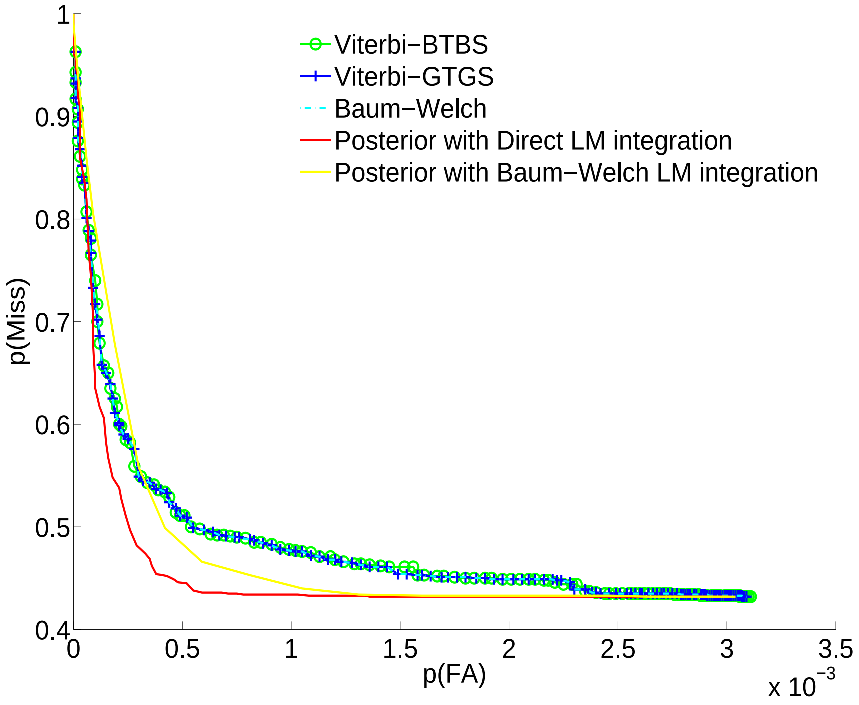

Figure 5.3: The DET curves for the confidence scoring approaches for Spanish STD.

any kind of posterior distribution, whereas systems based on GMM converge to the model, instead of a real distribution.

The worst absolute performance of the discriminative approach compared with the HMM-based approaches in the *Posterior with Baum-Welch LM integration* confidence score computation is caused by the weak bi-gram LM used in these experiments. However, it has been shown that when the system is evaluated from different operating points, such confidence scoring outperforms the HMM-based ones for much of the range.

5.4 Decision tree-based confidence measure

5.4.1 Preliminary work

Recently, meetings domain have been of interest of the community research in finding a selected list of terms on them. In this work we have focused on the STD approach from phone-based lattices as we did in the work for Confidence scoring. Nevertheless, we also present several preliminary results in accessing the information from other kind of units such as words in the output (1-Best and lattice) of an LVCSR system. The data provided by NIST for the RT-04 and RT-05 evaluations have been used in our preliminary experiments for the information retrieval in meetings. Two sets of experiments were run over such data. The LVCSR system used for both word-based recognition and phone-based recognition for both experiments was provided by the CSTR group in the University of Edinburgh. The information of the word-based LVCSR system can be found in [76]. For the phone-based system, the set of phonemes listed in Appendix B was used, along with a bigram LM trained from the text resources explained in Chapter 2. The LM used in the word-based LVCSR was trained from such resources as well. The set of HMMs used for both speech recognition processes has been trained from the data referred as *Meetings 2005 data* in Chapter 2.

The first experiment was conducted on the RT-04 data and was composed of the search of two different sets of terms. The first set contained 85 proper names appearing in the RT-04 evaluation data (denoted as *Proper names* in Table 5.3), of which 76 were INV and 9 were OOV and the second one contains 11500 INV words (denoted as *Common words* in Table 5.3) extracted from the CMU dictionary. In this experiment, INV refers to those proper names whose transcriptions appear in the CMU dictionary and OOV refers to those whose transcriptions were estimated from letter-to-sound rules from a CART module. We present these results in Table 5.3 under FOM and ATWV metrics when using the output of an LVCSR system (in the way of 1-Best and lattice) and when using a phone lattice. The LVCSR system and the phone-based system were tuned on the RT-04 development data for WER and FOM metrics respectively. All the words in both sets appear in the vocabulary of the LVCSR system.

The second experiment was conducted on the RT-05 data and was composed of two different lists of terms as well. The fist one has 64 proper names appearing in the RT-05 evaluation data (denoted as *Proper names* in Table 5.4) and the second one has the same 11500 INV words as the first experiment. Here, it must be noted that there are 16 proper names that do not appear in the vocabulary of the LVCSR system, being unaccesible by the word-based approaches. Their transcriptions were got from a CART module. The transcription of the rest of the names used in the evaluation was found in the CMU dictionary. As in the first experiment, the LVCSR system was tuned on the RT-04 development data for WER and the phone-based system was tuned on the RT-04 development data for FOM. These results are presented in Table 5.4 for the same metrics as in the first experiment.

	FOM		
	Word (1-Best)	Word (lattice)	Phone lattice
Proper names	43.0	63.2	59.1
Common words	30.8	36.5	22.6

	ATWV		
	Word (1-Best)	Word (lattice)	Phone lattice
Proper names	0.42	0.61	0.35
Common words	0.52	0.68	0.25

Table 5.3: Results in terms of FOM and ATWV for the RT-04 evaluation data. For all measures, higher values indicate better performance.

	FOM		
	Word (1-Best)	Word (lattice)	Phone lattice
Proper names	29.7	43.9	39.5
Common words	28.3	33.4	16.5

	ATWV		
	Word (1-Best)	Word (lattice)	Phone lattice
Proper names	0.34	0.48	0.28
Common words	0.59	0.73	0.22

Table 5.4: Results in terms of FOM and ATWV for the RT-05 evaluation data. For all measures, higher values indicate better performance.

From Tables 5.3 and 5.4 similar conclusions can be extracted. The word-based lattice approach outperforms the 1-Best word-based approach, as expected for the two sets of data. However, contrary to what can be expected, the word 1-Best approach achieves a worse FOM value for *Proper names* than the phone-lattice approach. It is due to the fact that the word-based approaches were tuned for WER and not for the specific FOM metric, which causes that some important words (those in the list of search terms) can contribute more to an error than others. The same conclusion can be extracted for the RT-05 evaluation data. However, as it is explained next, the difference between the 1-Best word-based approach and the phone-based one was not found to be significant. For ATWV value, the phone-lattice based approach is worse than the word 1-Best, which exhibits different behaviour that the FOM metric. A further explanation of the different system performance according to the metric used in the evaluation is described in the next chapter. As we increment the number of terms, more dependency exists between the WER and FOM, as more words contribute to the final FOM metric, the phone-lattice approach presents a worse FOM value than the word 1-Best. Paired t-tests were used to evaluate the significance of the FOM values and showed these conclusions: For the RT-04 evaluation data, on the *Proper names* set of data, there is only significant difference with $p < 0.001$ between the Word lattice approach and the Word 1-Best approach. For the set of *Common words*, there is also significant difference between these approaches and between the word lattice and the phone lattice one with $p < 0.001$. For the RT-05 evaluation data, there is no significant difference between the approaches for the *Proper names* set of data. It is caused by the fact that some of the words do not appear in the vocabulary of the LVCSR system, causing a worse performance in the word-based approach. However, in the *Common words* set, the results achieved by the three approaches were found to be statistically significant with $p < 0.001$ between all of them. The no difference for almost all of the approaches within the *Proper names* set is due to the small amount of occurrences of such terms in the corpus (178 occurrences in the RT-04 evaluation and 196 occurrences in the RT-05 evaluation) along with the WER metric used to tune the word-based approaches. However, when the number of occurrences is increased (as it is in the *Common words* set), the differences become significant.

5.4.2 Decision tree-based approach

5.4.2.1 Motivation and prior research

The terms in the list hypothesized by a STD system are classified in two different classes according to the metric evaluation (*hit* and *FA*). In this address, approaches dealing with classification tasks such as the ones based on NNs, SVMs and all able to binary classification can be used for STD. This preliminary work is based on a decision tree- or CART-based approach. It takes several input (prosodic and lexical) features and tries to predict if a keyword is actually a *hit* or a *FA*. Decision tree-based approaches have been already proposed in some other tasks such as sentence boundary detection [77], hot spot in meetings [78] and finding disfluencies in conversational speech [79]. They have been also used to show the correlation between the WER and the prosodic and lexical features according to the audio signal in LVCSR systems [80]. It determines which words are more likely to be misrecognized in the decoding process and therefore to produce a higher WER. Here we propose a new application of the decision trees on STD over the meetings domain. The goal of the decision tree is to classify the keywords proposed by the term searching tool (lattice search tool) as *hit* or FA, with the purpose of rejecting those classified as FA.

5.4.2.2 Decision tree and selected features

Decision tree is a common method for building statistical models from simple feature data. Decision trees are powerful because they can deal with incomplete data, multiple kind of features both in input features and predicted features, and the output trees they produce often contain rules which are easily readable. The following features (attributes in the decision tree) have been selected according to each keyword hypothesized by the STD approach to build the decision tree:

- Duration of the keyword (in hundredth of seconds). It is referred as **keyword duration** in Figure 5.5.
- The number of graphemes (letters) of the keyword. It is referred as **number of graphemes** in Figure 5.5.
- The number of vowel graphemes of the keyword. It is referred as **number of graphemes vowels** in Figure 5.5.
- The number of consonant graphemes of the keyword.
- The number of phones of the keyword.
- The number of vowel phones of the keyword.
- The number of consonant phones of the keyword. It is referred as **numer of phones consonants** in Figures 5.4 and 5.5.
- The position of the keyword (related to the beginning of the utterance, the end of the utterance and the middle of the utterance).
- The posterior probability estimated from the lattice search tool. In this case, the *Viterbi-BTBS* score. It is referred as **likelihood** in Figures 5.4 and 5.5.

- The language model probability of the keyword, computed from the LM component (i.e., the bigram). It is referred as **prob_LM** in Figure 5.5.
- The speaker phone rate, defined as the duration of the keyword (in hundredth of seconds) divided by its number of phones. It is referred as **number of phones duration** in Figure 5.5.
- The speaker vowel rate, defined as the duration of the keyword (in hundredth of seconds) divided by its number of vowels. It is referred as **number of vowels duration** in Figure 5.5.
- The gender of the speaker, i.e., male or female.
- The confusability of the keyword, computed as the number of hits divided by the number of FAs of that keyword. It is referred as **confusability** in Figures 5.4 and 5.5.
- The maximum, minimum and average Levenshtein distance for the keyword when it is matched with the rest. It was computed using the TREP package [81]. The average Levenshtein distance is referred as **average distance** in Figures 5.4 and 5.5.

It must be noted that each audio file is represented by a single speaker. Therefore, the *gender* is defined for each file. The *position of the keyword* is relative to each utterance and the *speaker phone rate* and the *speaker vowel rate* are defined for each speaker who pronounces a keyword.

As in any classification task, it is very likely that the number of occurrences that belong to the different classes varies greatly, even when there are only two classes (*hits* and *FAs* in our case). And the problem caused by such imbalanced data may affect the final performance of the STD system. Several techniques have been proposed to solve this issue [77]. In this initial work, we have chosen to reduce the majority class samples (Random downsampling), due to in our case, the minority class corresponds to *hits* and the ATWV metric used in the evaluation gives more penalty to a miss than a FA and the Random downsampling approach results in poorer performance for the majority class [82, 83] and therefore better performance for the minority class (hits).

The decision tree was built in the traditional way with the stop value criterion, represented by the n stop value. It means that at least n samples are required in a partition before a question split is considered. In addition to this, a same attribute may appear several times in the final decision tree. Two different tuning processes were considered in this work: The first one is depicted in Figure 5.5 and represents the decision tree when the approach is tuned to retrieve as many hits as possible (*biased towards hits*). The second one is depicted in Figure 5.4 and represents the tree when the system is tuned in terms of Classification Error Rate (CER).

As it is shown in both figures, the likelihood or score computed from the lattice search tool is the root of both trees. It means that such attribute has the less entropy (i.e., the maximum gain of information) of all of the attributes described previously. Therefore, it is the attribute that provides the best classification for the keywords. Such value is computed by the lattice search tool and makes use of the information stored in the lattice (i.e., the acoustic likelihood and the lingustic likelihood) during the decoding process. Moreover, it represents the confidence of the keyword, which should be greater when the keyword hypothesized is more likely to be a hit than a FA. Due to the rest of the features do not make use of any additional information to hypothesize that the occurrence is more likely to be a hit or a FA, it caused the likelihood to be the feature that provides with the best discrimination between hits and FAs. Some of the attributes used as input features do not appear in the decision tree. It means that such attributes do not provide the maximum gain of information each time a question split is considered in the training of the decision tree. Thus, such features are irrelevant given the whole set of features in classifying the keywords as *hit* or *FA*.

The different size of both decision trees, depending on the tuning, is caused by the different n stop value got from both tuning processes. It was assigned a value of $n = 35$ in the CER-based tuning and $n = 8$ in the *biased towards hits* tuning. It means that more questions are necessary to split each node or partition (i.e., to form a new level) of the decision tree for the former way of tuning that for the latter one. At the same time, as the decision tree is growing, less questions remain to be used for the next level. It causes that decision trees have less levels when more questions are required to be used before a partition splits.

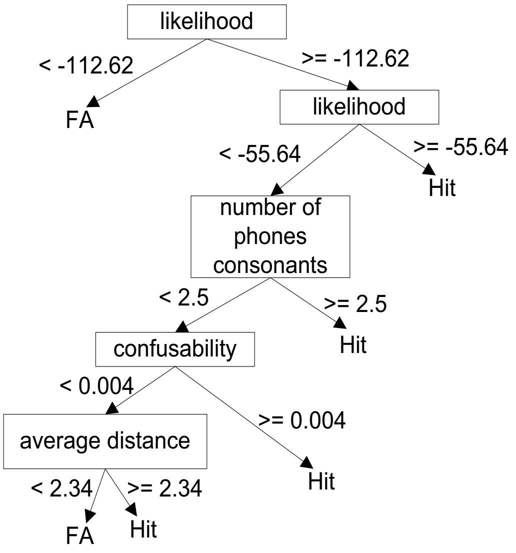

FIGURE 5.4: The decison tree for CER-based tuning.

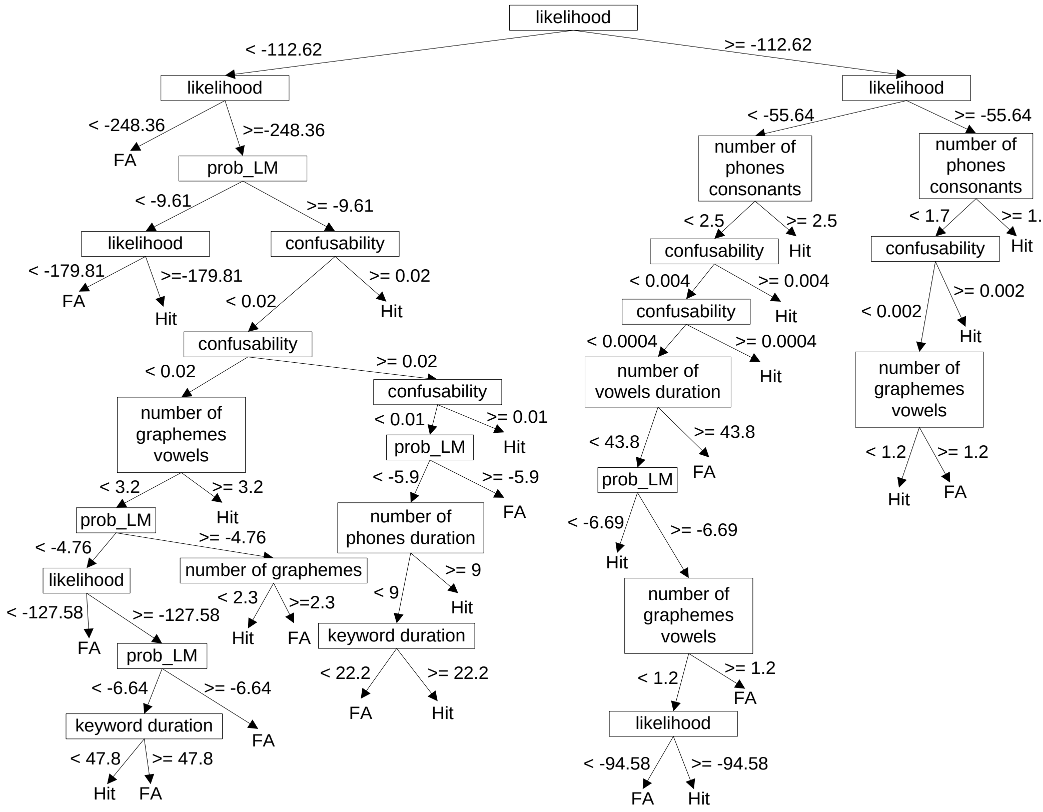

FIGURE 5.5: The decison tree for the *biased towards hits* tuning.

5.4.2.3 Experimental setup

A set of 11500 INV keywords (extracted from the CMU dictionary), and referred as *Common words* in the experiments explained before, was used as search terms. The NIST RT-04 evaluation was used as development set to build the decision tree and the NIST RT-05 evaluation was used as evaluation set. As the CER is the metric widely used to evaluate the performance of CARTs and the ATWV is the metric used to evaluate the STD performance, we have presented two different tuning processes for the whole STD system, composed of the lattice generation, the lattice search tool and the decision tree-based confidence measure. The first and more natural one is to select the parameters in the decision tree which presents the best CER value directly. However, as the ATWV metric gives more penalty to a miss than a FA, we have also tuned the system to achieve the best hit performance. The only parameter necessary to be adjusted is the n stop value, which presents a different value for CER tuning and for *biased towards hits* tuning. The lattices and the final list of terms hypothesized from them for both the RT-04 evaluation and the RT-05 evaluation, used to build and evaluate the decision tree performance respectively, were tuned for FOM in the RT-04 development data.

5.4.2.4 Results and discussion

Table 5.6 shows the ATWV values for the STD system with and without the decision tree-based approach. It shows that the decision tree-based approach outperforms slightly the final value of the STD system compared with the performance without it. Contrary to the ***geographic training set***, where CER-based tuning seems to outperform the *biased towards hits* tuning, in the ***geographic test set***, it is shown that the latter outperforms both the CER-based tuning and the performance without the CART. However, the DET curves in Figures 5.6 and 5.7 for both the ***geographic training set*** and the ***geographic test set*** show similar behaviour for both ways of tuning. For both sets, it is shown that the CER-based tuning outperforms the two other performances for much of the range.

5.4.2.5 Conclusions

Due to the small improvement achieved in terms of absolute ATWV value, the DET curve reports strongest conclusions from a set of different operating points.

	No CM	CART (Hits)	CART (CER)
ATWV	0.33	0.33	0.35

Table 5.5: Results in terms of ATWV without the decision tree-based approach (No CM) and with it when it is tuned for *biased towards hits* (CART (Hits)) and for CER (CART (CER)) for the ***geographic training set***.

	No CM	CART (Hits)	CART (CER)
ATWV	0.19	0.20	0.18

Table 5.6: Results in terms of ATWV without the decision tree-based approach (No CM) and with it when it is tuned for *biased towards hits* (CART (Hits)) and for CER (CART (CER)) for the ***geographic test set***.

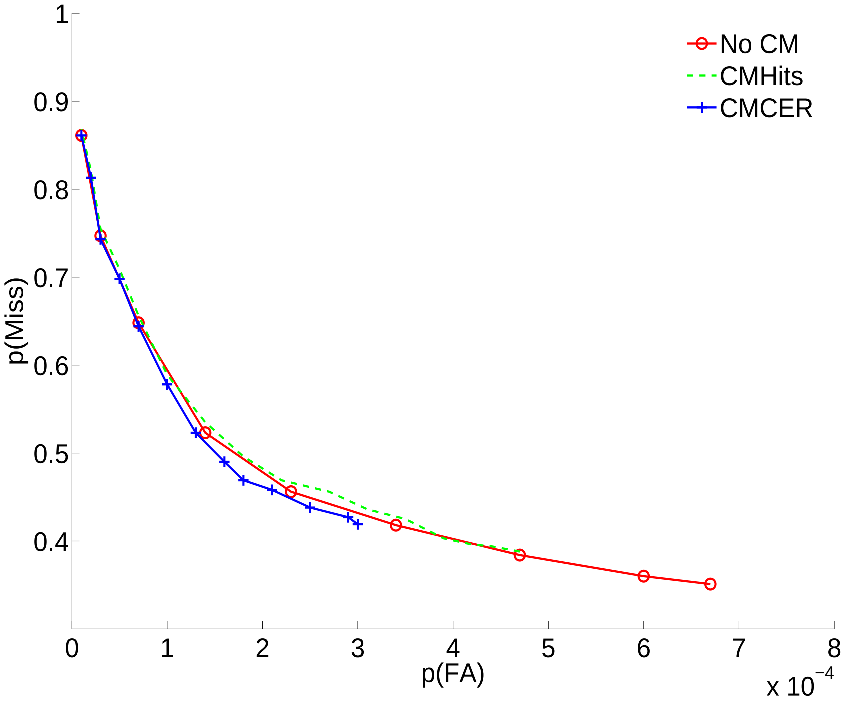

Figure 5.6: The DET curves for the development set without the decision tree-based approach (No CM) and with it when it is tuned for the *biased towards hits* way (CMHits) and for CER (CMCER).

A similar pattern was found in such curve for both the ***geographic training set*** and the ***geographic test set*** (contrary to the single best ATWV value) and shows that the tuning of the CART for CER metric did improve the final STD performance for much of the range. Although the work done in this direction using a decision tree to improve the final STD performance is very preliminary (a minor

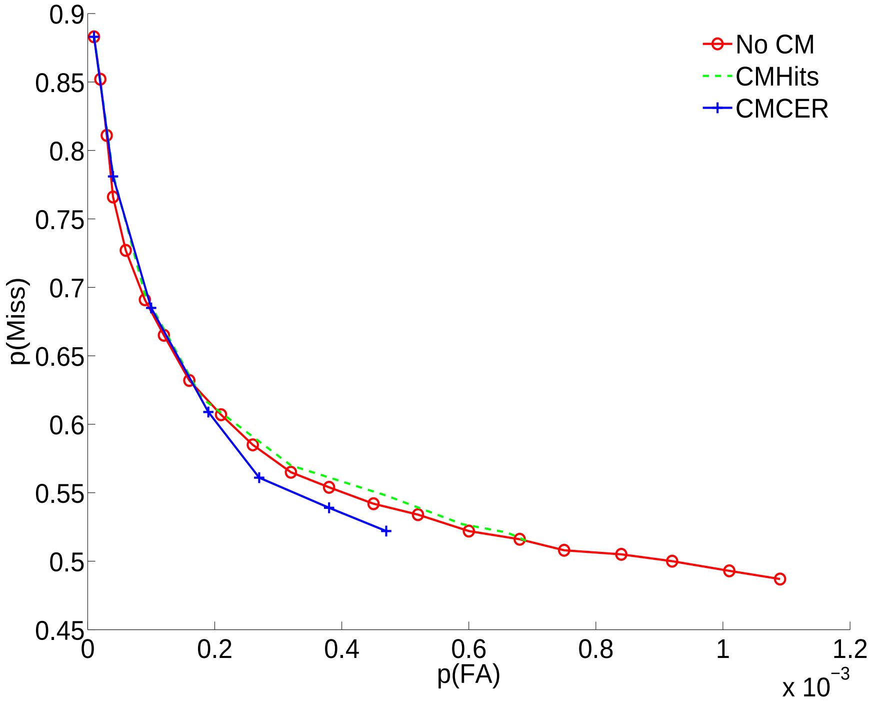

FIGURE 5.7: The DET curves for the test set without the decision tree-based approach (No CM) and with it when it is tuned for the *biased towards hits* way (CMHits) and for CER (CMCER).

improvement is achieved) and more features can be used to build the final decision tree, we have shown that this approach has achieved encouraging results in STD. Based on these results, we can conclude that the final set of features plays a very important role to improve the final STD performance.

5.5 Summary

In this chapter we have presented two approaches applied over STD, which constitute the contributions to STD. The first one is based on a discriminative approach, by means of an MLP training, to compute the final confidence score of the list of keywords hypothesized by the system. The second one is based on a decision tree to reject the keywords proposed by the STD system which it classifies as FA. We have also presented a set of experiments developed over the Spanish ALBAYZIN database for the first approach and over the English meetings domain for the second approach. The final confidence score computation by using the

MLP and the LM component outperformed the rates achieved by the HMM-based techniques due to it makes use of a discriminative approach instead of generative models-based approaches, typically by using HMMs trained with the maximum likelihood criterion, which suffer from several incorrect assumptions such as the framewise and the component-wise independence of the acoustic features. In terms of ATWV, the improvement achieved goes from 0.18 with the best HMM-based confidence score to 0.26 achieved with such MLP-based approach. The use of decision trees as confidence measure needs a more robust feature selection to achieve a better performance than the one showed in this preliminary work since a minor improvement is achieved with the features proposed. Nevertheless, we have shown the potential possibilities of this approach to be applied over STD.

Chapter 6

Phone- versus Grapheme-based systems for Keyword Spotting and Spoken Term Detection in Spanish

6.1 Introduction

The decoding process is a crucial element for both Keyword Spotting and STD systems. Therefore, the choice of their units inventory plays a very important role in the final system performance. Thus, this chapter aims to compare two different acoustic models for Keyword Spotting and STD on the Spanish language. The first part compares both acoustic models using a standard MFCC-based feature extraction. Another important component in Keyword Spotting and STD is the feature extraction from which the input acoustic signal is transformed into a sequence of feature vectors. In this thesis we have studied the influence of the feature extraction process in the choice of the units inventory. The second part of this chapter presents the comparison between the two acoustic models in using two different feature extraction processes on the STD task. Finally, the third part of this chapter presents the combination of both types of acoustic models for both types of feature extraction processes on the STD task from several confidence scoring techniques. Experiments, results and main conclusions are reported for each part separately. The contributions of this chapter rely on the comparison of phone-

and grapheme-based acoustic models for Spanish Keyword Spotting and STD, the influence of the feature extraction process in both types of acoustic models for Spanish STD and the combination of both types of acoustic models for Spanish STD.

6.2 Motivation and prior research

The phoneme-based acoustic models have been widely used for ASR tasks. In doing, the words in the lexicon of the LVCSR systems were modeled by a sequence of phones or phonemes, according to a target language. However, some other related works have been also proposed the use of the graphemes, i.e., letters as units in the acoustic modelling [84, 85, 86, 87, 88].

Killer et al. [84] showed that grapheme-based LVCSR systems for Spanish can achieve performance which is close to that of phone-based systems. In some other languages, notably English, the speech sounds are harder to predict accurately from graphemes, so grapheme-based units typically perform worse than the phone-based units for acoustic modelling [84].

However, Dines and Doss [89], showed that the use of graphemes in English can yield competitive performance for small to medium vocabulary tasks in ASR systems. In experiments on the OGI Numbers95 task [90], a grapheme-based ASR system was found to give similar performance to the phone-based approach. However, on tasks of increased complexity, such as DARPA resource management [91], and CTS [92], the phone-based system gave lower error rates than the grapheme-based system.

Doss et al. [93, 94] also proposed the use of a phone-grapheme based system that jointly model both the phone and grapheme sub-word units during training. During decoding, recognition is performed either using one or both sub-word units. This was investigated in the framework of a hybrid HMM/ANN system. Improvements were obtained over a CI phone-based system using both sub-word units in recognition in two different tasks: isolated word recognition [94] and recognition of numbers [93].

On the other hand, typically in STD task, for OOV search terms, letter-to-sound rules must be used to generate a pronunciation for each search term. This is usually a non-probabilistic issue and a difficult decision, particularly for English language, and errors introduced in this step are difficult to recover from. Instead of enforcing a potentially hard decision on the sequence of phone units, the relationship between graphemes and sounds will be modeled probabilistically by HMMs rather than an external letter-to-sound model. This is expected to work particularly well in languages such as Spanish, where the letter-to-sound mapping is very regular, commonly in the way 1-to-N, contrary to languages such as English where the mapping is N-to-M. It means that the letter-to-sound conversion can be achieved more reliably than for some other languages such as English. By modelling grapheme-based units directly, we have the advantage of replacing a potentially error-prone hard decision with a probabilistic one which naturally accounts for this variation. In case of using a letter-to-sound module with no errors (as it occurs in Spanish), grapheme still presents the advantage of that more powerful LMs can be trained directly from large text corpora, without such module.

Therefore, given the performance of grapheme-based models for Spanish LVCSR and the potential advantages of grapheme over phone-based units for tasks involving OOVs, we propose that grapheme-based acoustic modelling can *outperform* phone-based modelling for certain applications.

In addition to this, given the different information stored in the HMMs in phone- and grapheme-based systems, which may produce different and complementary detections, we also propose that the combination of both acoustic units can *outperform* each system in isolation.

6.3 Comparison of phone- and grapheme-based units on MFCC-based feature extraction

Three different architectures have been used in comparing phone- and grapheme-based acoustic units. The first and second one are able to STD, as it was suggested by NIST, due to the list of terms is unknown during the decoding process, so the audio is indexed in terms of both types of sub-word units and a following step proposes the keywords from such index. The third architecture can be only applied over Keyword Spotting and not over STD following the NIST recommendations (i.e., without using the audio in a step different from the first one), due to it makes use of a prior knowledge of the list of search terms before the decoding process. This work has been published in [95] and [96].

6.3.1 Architecture *1-Best*: 1-Best sub-word unit decoding + lexical access

This architecture is illustrated in Figure 6.1. The first step uses the Viterbi algorithm in the HTK tool [9] to produce the single most likely (1-best) sequence of phones or graphemes, using the HMM sets trained as described in Section 6.3.4.2. We refer to this as the *Sub-word unit decoder* in Figure 6.1, and the output is a sequence of U phone or grapheme sub-word units $S = \{s^1, s^2, \ldots, s^U\}$.

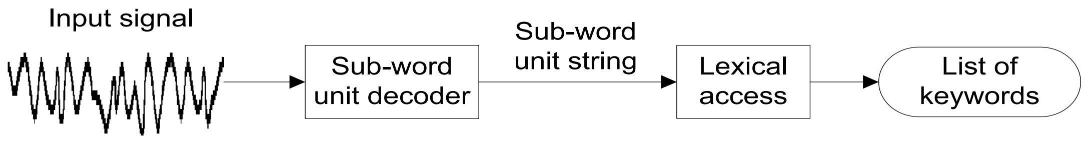

Figure 6.1: The Architecture *1-Best*.

Each keyword W is represented as a sequence of R phone or grapheme sub-word units $W = \{w^1, w^2, \ldots, w^R\}$, and search is performed within S, the output of the *Sub-word unit decoder*. This approach is based on the DP algorithm proposed by Fissore et al. [72], and used in the *Lexical access* confidence measure presented for Keyword Spotting in Chapter 4. The essence of the algorithm is to compute the cost of matching each keyword W with the decoded output S. Therefore, the following algorithm hypothesizes the final list of keywords for both sub-word units. The keyword search over a length L hypothesized sequence of sub-word units progresses as follows:

1. For each keyword K, set the minimum window length to $W_K^{min} = N_K/2 + 1$, where N_K is the number of sub-word units contained in the dictionary entry for keyword K. Set the maximum window length as $W_K^{max} = W_K^{min} + N_K$.

2. Calculate the cost G for each keyword K over each candidate window.

3. Sort keyword hypotheses according to G, removing any for which the cost G is greater than a threshold $\Theta_{G_{max}}$.

4. Remove overlapping keyword hypotheses: make a pass through the sorted keyword hypotheses starting with the highest-ranked keyword, removing all hypotheses with time-overlap greater than $\Theta_{overlap}$%. If a same keyword is hypothesized overlapped, the occurrence with the lowest cost G remains.

5. Return all keyword hypotheses with cost less than $G_{best}+\Theta_{G_{beam}}$, where G_{best} refers to the cost of the highest-ranked keyword and $\Theta_{G_{beam}}$ is beam width.

To estimate the cost G, the same equations as in the *Lexical access* confidence measure in Chapter 4 were used. Such cost G, divided by the length of the decoded output S, is used as confidence score for all of the keywords hypothesized in this architecture.

As an example of the windowing in the grapheme-based approach, searching for the keyword `madrid`, which has a grapheme transcription {`m a d r i d`}, given a grapheme decoder output of {`m a i d r i e d a a n`}, the minimum and maximum windows are $W_K^{min} = 6/2 + 1 = 4$ and $W_K^{max} = 4 + 6 = 10$. The cost G is therefore computed over the following candidate windows:

{`m a i d`}, {`m a i d r`}, {`m a i d r i`}, {`m a i d r i e`}, {`m a i d r i e d`}, {`m a i d r i e d a`}, {`m a i d r i e d a a`}, {`a i d r`}, {`a i d r i`}, ..., {`i e d a`}

6.3.2 Arhitecture *Lattice*: sub-word unit lattice + exact word matching

Lattice search provides a natural extension to the 1-best path architecture above, and again search is based on sub-word (phone or grapheme) units.

The decoding process for the 1-best decoder from Section 6.3.1 was used, except that it was run in N-best mode. The resulting output were lattices generated from the top N tokens in each state. An example of a grapheme lattice is shown in Figure 6.2.

A recursive match algorithm provides an efficient method to find all path fragments in the lattice that exactly match the phone or grapheme string representing search terms. We used an implementation provided by the Brno University of Technology (lattice search tool) [39]. This architecture is depicted in Figure 6.3.

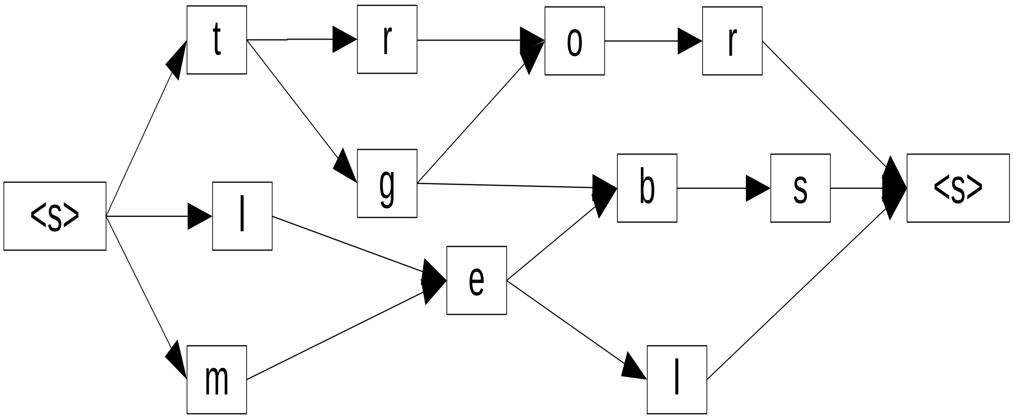

FIGURE 6.2: An example of a lattice containing graphemes. <s> denotes the start and the end of the lattice at the left and the right of the lattice respectively.

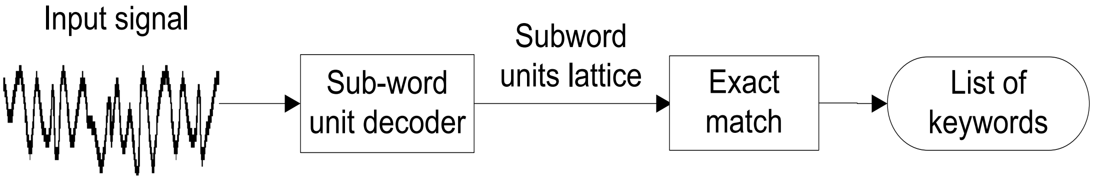

FIGURE 6.3: The Architecture *Lattice*.

For each hypothesized keyword K which the search returns, a confidence score C_K is calculated as follows:

$$C_K = L_a(K) + L(K) + L_b(K) - L_{best} \tag{6.1}$$

where:

- $L_a(K)$ is the log likelihood of the best path from the lattice start to the node of the first phone or grapheme of K.
- $L(K)$ is the log likelihood of keyword K, computed as the sum of the acoustic log likelihood, plus the word insertion penalty plus the total language model log likelihood weighted by the language model scale factor of its constituent phones or graphemes.

- $L_b(K)$ is the log likelihood of the best path from the node of the last phone or grapheme of K to the end of the lattice.
- L_{best} is the log likelihood of the 1-best path over the whole lattice.

$L_a(K)$ and $L_b(K)$ are computed using standard forward-backward processes [9]. This confidence score is the same as the one denoted as *Viterbi* confidence explained in Chapter 5. The *Best Time Best Score* method explained in Chapter 5 was used to remove overlapped occurrences of the same keyword in the search stage.

6.3.3 Architecture ***Hybrid***: hybrid word + sub-word system

This is the same architecture used for Keyword Spotting which makes use of the *Lexical access* confidence measure explained in Chapter 4.

6.3.4 Experimental setup

6.3.4.1 Feature extraction

The input signal is sampled at 16kHz and stored with 16 bit precision. MFCCs were computed at 10ms intervals within 25ms Hamming windows. Energy and first and second order derivatives were appended giving a series of 39-dimensional feature vectors.

6.3.4.2 Acoustic Modelling

The same set of phone-based units used in Chapter 4 for Keyword Spotting as acoustic modelling was used in these experiments.

The grapheme systems were built in an identical fashion to the phone-based systems; the only differences were in the inventory of sub-word units and the questions used for state clustering. The monographeme models used mixtures of Gaussians with 15 components, and the trigrapheme models used 8 components. There are 3575 shared states retained after clustering in the trigrapheme system.

To build state-tied CD grapheme models (trigraphemes) requires a set of questions used to construct the decision tree. There are three ways to generate those questions: using only questions about single graphemes ("singleton questions"), converting from the questions used to state-tie triphones according to a phone-to-grapheme map, or generating questions from data automatically. To simplify the building of the set of questions, we used a singleton question set for state tying in our experiments.

6.3.4.3 Language Modelling

Two different LMs were used for the three architectures. For the architecture *Hybrid*, we have used a bigram for phone- and grapheme-based decoding and a pseudo N-gram as explained in Chapter 4 for the *HMM-based keyword spotting* process. For the architectures *1-Best* and *Lattice* we have used a bigram as LM for both phone- and grapheme-based decoding.

6.3.4.4 System tuning

The ***phonetic test set*** was used to select the number of components GMM in each state of each HMM according to phone and grapheme accuracy. Therefore, the same sets of HMMs were used for the three systems.

The three systems, represented by each of the three architectures used in this chapter were tuned separately as follows:

In the architecture *Hybrid*, the value of N in the pseudo N-gram used as LM was chosen to get a desired tradeoff between precision and recall for all of the acoustic model configurations. The probability for the keyword class was set to be 6 and 12 times that of the filler models in the CI and CD systems respectively. The p and s parameters in the Viterbi decoding, for both the *HMM-based keyword spotting* process and the *Sub-word unit decoder*, were tuned for each system according to the different metrics used in the evaluation. The threshold α in the *Lexical access module* was also tuned for each metric. All of these parameters were tuned on the ***geographic training set***.

In the architecture *1-Best*, the p and s parameters in the phone- and grapheme-based decoding were tuned according to each metric. The thresholds $\Theta_{G_{max}}$,

$\Theta_{overlap}$, and $\Theta_{G_{beam}}$, were set on the ***geographic training set*** according to the different metrics used in the evaluation and the window sizes W_K^{min} and W_K^{max}, were set on the ***geographic training set*** in order to give the desired tradeoff between precision and recall. The set of costs was trained on all of the words in the ***geographic training set***.

In the architecture *Lattice*, the p and s values were tuned according to each metric for both the phone- and grapheme-based decoding and the lattice search tool on the ***geographic training set***. Preliminary experiments [39] found that given a suitably dense lattice, the accuracy improvement from allowing non-exact matches was minimal, and that $N = 5$ gave a suitably dense lattice.

6.3.5 Results and discussion

Apart from the FOM metric already defined for Keyword Spotting, two additional metrics (OCC and ATWV) explained in Appendix D have been used in these experiments. Instead of presenting the ATWV value, we have presented those results graphically in order to show the full range of operating points, using the DET curves. Significance tests in the form of paired t-tests are used to compare systems, in order to determine whether differences are consistent across search terms. The set of keywords used for the keyword spotting experiments in Chapter 4 and listed in Appendix D was used for these experiments.

6.3.5.1 Recognition accuracy

Whilst phone or grapheme recognition is not the main focus of this set of experiments, it is an important factor in STD/Keyword Spotting performance. We present the phone and grapheme accuracy results in Tables 6.1, 6.2 and 6.3. The results presented in Table 6.3 used the p and s parameters tuned on the ***geographic training set***.

For both phone and grapheme systems, performance is improved through the use of the CD acoustic models. The grapheme recognition accuracy is higher, though this is expected as there are fewer graphemes than phones.

	monophone	triphone	monographeme	trigrapheme
Recognition accuracy	63.9%	68.2%	75.2%	79.1%

Table 6.1: Phone and grapheme recognition accuracy for *MFCC* feature vectors for both CI and CD acoustic models. Results are presented on the ***phonetic test set***.

	monophone	triphone	monographeme	trigrapheme
Recognition accuracy	61.9%	65.1%	73.9%	78.6%

Table 6.2: Phone and grapheme recognition accuracy for *MFCC* feature vectors for both CI and CD acoustic models. Results are presented on the ***geographic training set***.

	monophone	triphone	monographeme	trigrapheme
Recognition accuracy	63.0%	66.4%	74.1%	78.4%

Table 6.3: Phone and grapheme recognition accuracy for *MFCC* feature vectors for both CI and CD acoustic models. Results are presented on the ***geographic test set***.

6.3.5.2 Spoken term detection and keyword spotting results

Architecture *Hybrid* uses a standard HMM-based keyword spotting process in combination with a sub-word unit-based confidence measure. In order to examine the gain due to the confidence measure, Table 6.5 presents the results for the *HMM-based keyword spotting* process in isolation.

	HMM-based keyword spotting process			
	monophone	triphone	monographeme	trigrapheme
FOM	65.9	70.4	62.3	69.4
OCC	0.81	0.83	0.80	0.84

Table 6.4: Evaluation of the *HMM-based keyword spotting* process of Architecture *Hybrid* in isolation. Results are given in terms of FOM and OCC for both CI and CD acoustic models, using grapheme and phone units for the ***geographic training set***. For all measures, higher values indicate better performance.

These results show that the performance improvement in moving from CI to CD acoustic models is greater for grapheme-based models than for phones. Paired t-tests show that there is no systematic differences between the results of CD phone- and grapheme-based systems for FOM and OCC values. For the OCC

	HMM-based keyword spotting process			
	monophone	triphone	monographeme	trigrapheme
FOM	65.9	68.3	61.0	67.6
OCC	0.74	0.73	0.66	0.78

Table 6.5: Evaluation of the *HMM-based keyword spotting* process of Architecture *Hybrid* in isolation. Results are given in terms of FOM and OCC for both CI and CD acoustic models, using grapheme and phone units for the ***geographic test set***. For all measures, higher values indicate better performance.

metric, although the triphone-based system outperformed the monophone one in the ***geographic training set***, the contrary occurs for the ***geographic test set***. However, such difference was found to be insignificant.

	FOM			
	monophone	triphone	monographeme	trigrapheme
Architecture *1-Best*	73.4	73.9	66.4	74.8
Architecture *Lattice*	44.6	50.3	59.8	68.1
Architecture *Hybrid*	84.6	84.6	80.6	83.2

	OCC			
	monophone	triphone	monographeme	trigrapheme
Architecture *1-Best*	0.70	0.72	0.68	0.76
Architecture *Lattice*	0.42	0.45	0.53	0.63
Architecture *Hybrid*	0.86	0.86	0.86	0.89

Table 6.6: Results in terms of FOM and OCC for the three architectures for CI and CD phone and grapheme acoustic models for the ***geographic training set***. For all measures, higher values indicate better performance.

Table 6.7 presents the results in terms of FOM and OCC for each of the three architectures described above in Section 6.3.

We first note that comparing the results of the architecture *Hybrid* with those in Table 6.5, the addition of the confidence measure leads to performance improvements for each metric. However, it is only for the monographeme and triphone systems evaluated under the FOM metric that the increases are statistically significant with $p < 0.001$.

Evaluation in terms of FOM Table 6.7 shows that for evaluation in terms of FOM, CD acoustic models give the the best performance for all architectures

	FOM			
	monophone	triphone	monographeme	trigrapheme
Architecture *1-Best*	72.7	73.5	65.9	74.4
Architecture *Lattice*	44.0	47.1	58.1	64.0
Architecture *Hybrid*	80.3	82.3	76.9	79.6

	OCC			
	monophone	triphone	monographeme	trigrapheme
Architecture *1-Best*	0.70	0.72	0.67	0.76
Architecture *Lattice*	0.40	0.42	0.53	0.61
Architecture *Hybrid*	0.85	0.84	0.84	0.85

Table 6.7: Results in terms of FOM and OCC for the three architectures for CI and CD phone and grapheme acoustic models for the ***geographic test set***. For all measures, higher values indicate better performance.

and for both phone- and grapheme-based models. Significance tests show that for the architecture *Lattice*, the grapheme-based systems give consistent increases in performance over the best and the worst phone-based system with $p < 0.001$. Trigraphemes gave the best performance on architecture *1-Best*, though this was not found to be statistically significant. For architecture *Hybrid*, the best results are found using phone-based models, though the difference is not statistically significant.

Evaluation in terms of OCC We find similar patterns where the evaluation is in terms of OCC, though the performance for the phone-based models does not improve by moving from CI to CD acoustic models for architecture *Hybrid*. Graphemes give better performance than phones for all of the systems for architecture *Lattice*, shown to be significant with $p < 0.001$. For architecture *Hybrid*, the results are very similar, and for architecture *1-Best*, the trigrapheme gives the highest performance, though the result is not statistically significant.

Evaluation in terms of ATWV We present the DET curves of the ATWV performance for each of the three architectures. Each plot shows miss against false alarm probability for CI and CD acoustic models, for both phone- and grapheme-based systems, giving an indication of the system performance at a number of operating points.

The DET curves for architecture *1-Best* in Figure 6.4 show that the performances are quite similar for each of the systems, though the trigrapheme acoustic models marginally outperform the others for much of the range.

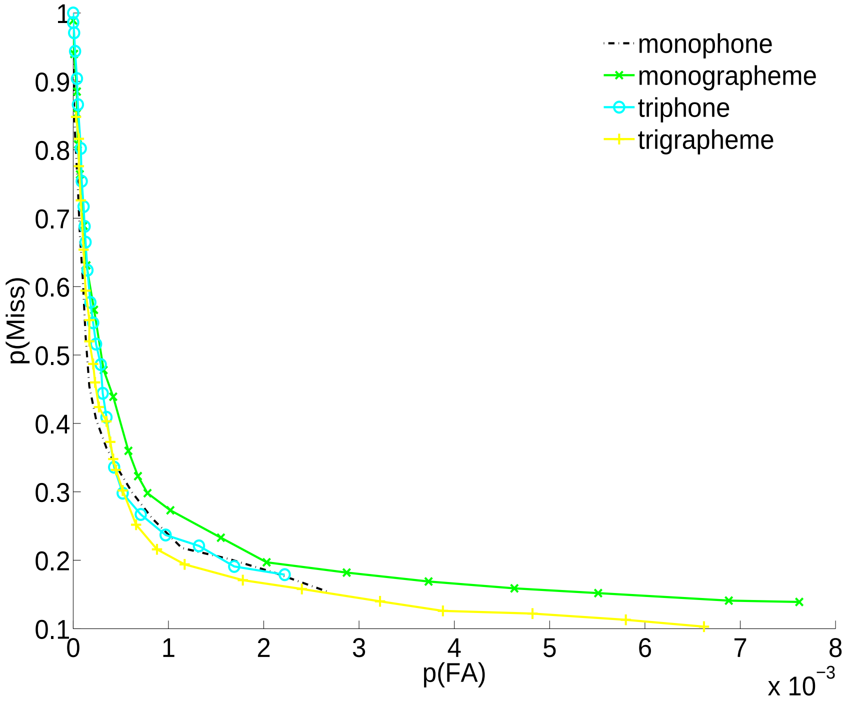

FIGURE 6.4: The DET curves for *MFCC* feature vectors for the CI and CD acoustic models for the Architecture *1-Best*.

Figure 6.5 shows the sizable performance gap between phone- and grapheme-based acoustic models for the architecture *Lattice*, and that for most of the range, the trigrapheme system provides a lower bound. It is also shown that the monographeme system outperforms both monophone and triphone systems.

The DET curves for the architecture *Hybrid* are given in Figure 6.6, and show that the best performance is achieved by the monophone system. It is due to the amount of additional information provided to the system with the *Lexical access module* which helps more the monophone system.

6.3.6 Conclusions

Our results suggest that grapheme-based units perform at least as well as phone-based units for Keyword Spotting and STD, and that the relative performance of

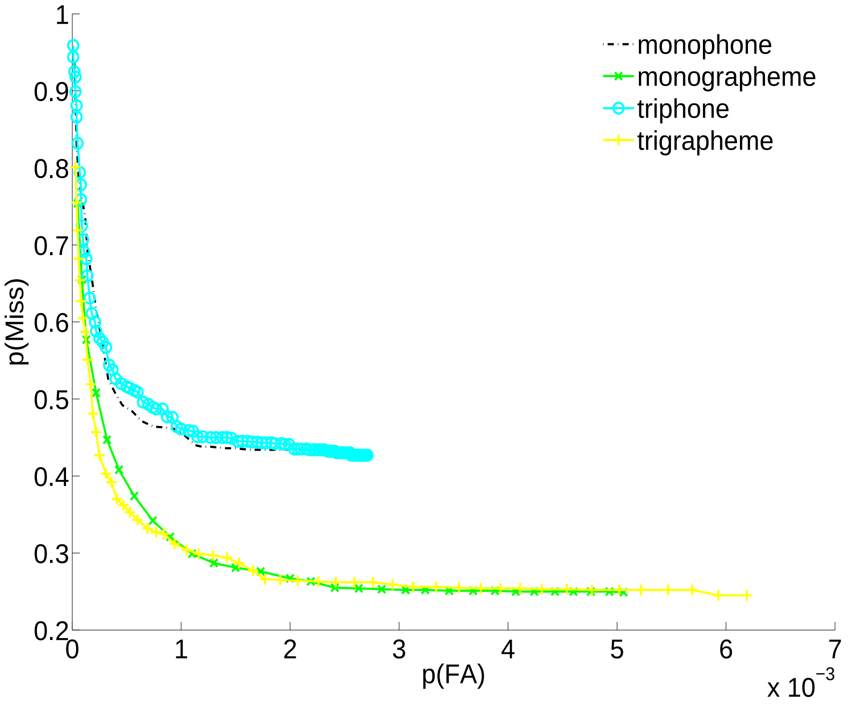

Figure 6.5: The DET curves for *MFCC* feature vectors for the CI and CD acoustic models for the Architecture *Lattice*.

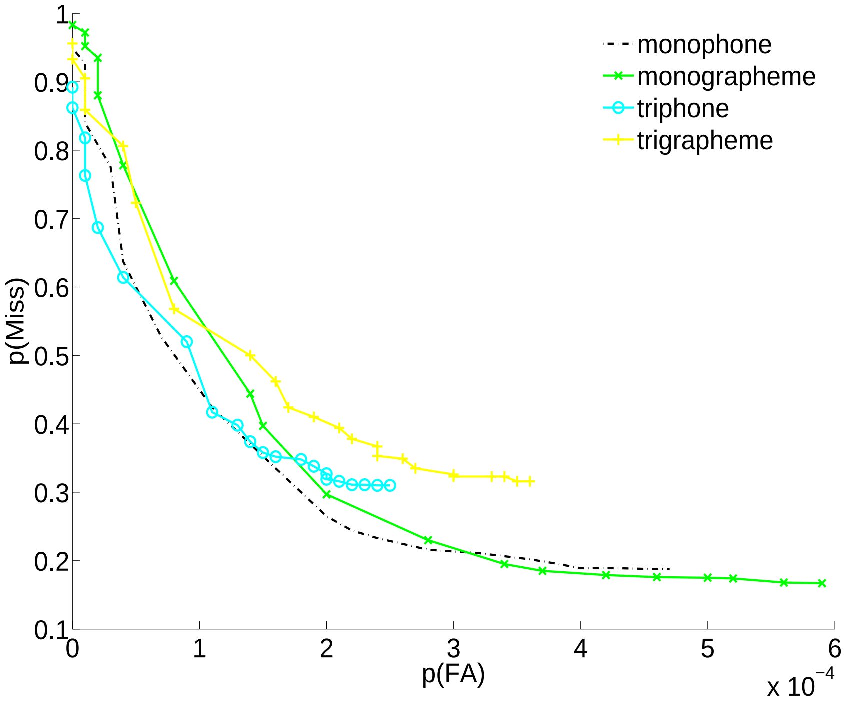

Figure 6.6: The DET curves for *MFCC* feature vectors for the CI and CD acoustic models for the Architecture *Hybrid*.

phone/grapheme models varies according to the architecture. As expected, better results were found for vocabulary-dependant systems.

Hybrid approach Architecture *Hybrid*, which is the most complex and the most vocabulary dependent, gives the overall best performance for each type of sub-word unit, and for each evaluation metric. The DET curves in terms of ATWV metric in Figure 6.6 show that the best performance is achieved by the monophone system. At the same time the difference in FOM and OCC performance across the different acoustic models is not significant. These results are attributed to the addition of other knowledge sources. These include the keyword network in the *HMM-based keyword spotting* process and the empirically-trained costs in the *Lexical access module*, which makes it more robust to weaker acoustic models. However, this architecture cannot perform STD (as recommended by NIST) because it requires knowledge of the keywords when processing the speech data.

1-best approach Architecture *1-Best* is capable of STD. Again, there is not significant variation in performance across the 4 acoustic model types, because of the additional knowledge used in the form of the *Lexical access* module. However the DET curves in terms of ATWV metric in Figure 6.4 shows that the trigrapheme models marginally outperform the others for much of the range.

Lattice-based approach Architecture *Lattice*, with no *Lexical access* module, is the most vocabulary and corpus independent system and conforms with the requirements of recent NIST evaluations. Under this architecture we find more marked performance differences between the different acoustic models. Our experiments give evidence that for the lattice-based approach, grapheme-based systems outperform equivalent phone-based methods.

Comparing the CI and CD systems, we found that the grapheme-based approach benefits more from CD modeling than the phone-based approach. This is expected, as a grapheme may be pronounced quite differently according to context. By comparison, CD allophones belonging to the same central phone are typically subject to a smaller degree of variation.

Grapheme-based modelling We consider that the power of the grapheme-based system on STD tasks, especially in the lattice-based architecture, is attributed to several factors. The first is the probabilistic description of pronunciation variation in the grapheme model, which helps represent all possible pronunciations of a search term in a single form. The second is its capacity to incorporate additional information, including both acoustic and phonological cues, in the lattice, thus improving the decision-making process in the search phase. The regular mapping between phones and graphemes in Spanish language was also found to be substantial to the improvement achieved by the grapheme-based systems along with the fact that there are fewer graphemes than phones, and therefore, less errors are more likely to be presented in the final system. It is confirmed by the ranking presented in the phone and grapheme accuracy-based experiments, where the monographeme system achieved better grapheme accuracy than the phone accuracy got in the triphone system. Although initially such values cannot be compared directly, as they are used for a same final task (STD), as less errors occur in the final output in the grapheme decoding, better performance is achieved by the system when such number of errors plays an important role in the final performance, as in the Architecture *Lattice* occurs.

Grapheme-based systems do not appear advantageous under the 1-best and hybrid approaches of architectures *1-Best* and *Hybrid*, where the single most likely phone or grapheme and keyword sequences are used rather than lattices for keyword search. Given the increased acoustic variation associated with graphemes compared with phones, the advantage arises from postponing hard decisions and keeping multiple decoding paths alive. Furthermore, as stated above, the additional linguistic information from the *Lexical access module* diminishes the relative performance of the different acoustic models.

6.4 The influence of the feature extraction for phone- and grapheme-based units for Spanish STD

The use of *tandem features* in the feature extraction process was shown to improve the system performance in ASR tasks [10, 11, 12, 13, 14, 15, 89] compared with the use of the standard MFCC-, PLP- or LPCC-based features in isolation.

Therefore, in our study of both types of units as acoustic models we have also added to the standard set of 39 MFCCs to represent the audio signal, the tandem features computed as explained in Chapter 2. We have selected the architectures *1-Best* and *Lattice* presented in the Section 6.3 to compare both types of feature vectors for both units for STD on the Spanish language and to extract the conclusions according to the metrics defined previously.

The tuning of these two systems with *tandem features* was made in an identical fashion as those in the Section 6.3. As the CD acoustic units achieved a better performance than the CI ones, we have used the triphone and trigrapheme configurations as the acoustic models over which both types of features are applied. The tandem features-based triphone system had 8-components GMM, with 8876 shared states and the tandem features-based trigrapheme system had 8-components GMM with 4739 shared states retained after clustering.

In building the *tandem features*, several parameters are necessary to compute and to tune prior to extract the final set of features:

1. From the ***phonetic training set***, consisting of 4400 utterances, we have selected 4000 of them to train the MLP and 400 for cross-validation to tune the number of units in the hidden layer of the MLP and the learning factor and the number of epocs in the MLP training. They were tuned based on cross-validation accuracy for both phone- and grapheme-based acoustic units independently.

2. The matrix to be used in the KL transform was computed from the ***phonetic training set***.

3. The number of coefficients that remain after the KL transform was estimated from the ***phonetic training set***. They were 19 for the phone-based system and 16 for the grapheme-based system.

The phone-based system contained an input layer composed of 351 units, a hidden layer with 1100 units and an output layer with 50 units (one for each phone plus the beginning and end silence and the short pause). The grapheme-based system contained an input layer composed of 351 units, 1300 units in the hidden layer and 31 units (one for each grapheme plus the beginning and end silence and the short pause) in the output layer.

A final set of 58 *MFCC+Tandem features* coefficients was used for the phone-based system (39 MFCCs + 19 tandem coefficients after the KL transform) and

55 *MFCC+Tandem features* coefficients were used for the grapheme-based system (39MFCCs + 16 tandem coefficients after the KL transform). In order to compare both acoustic models separately, the phone acoustic models were built from the phone-based tandem features and the grapheme acoustic models were built from the grapheme-based tandem features.

The MLP trained for phone- and grapheme-based acoustic units achieved a 70.6% and 77.0% respectively of cross-validation accuracy. As it is shown for phone and grapheme accuracy in Section 6.4.1.1, the cross-validation accuracy for graphemes is better than for phones, due to there are fewer graphemes than phones.

6.4.1 Results and discussion

The FOM and OCC metrics and DET curves got from the ATWV metric as in Section 6.3 were used to evaluate both types of acoustic models across each kind of feature extraction. Paired t-tests were also used to determine if the differences across the keywords are significant for both the feature extraction processes and the acoustic models.

6.4.1.1 Recognition accuracy

In Table 6.8 we present the results for phone and grapheme accuracy for the ***phonetic test set*** of the Albayzin database.

	Recognition accuracy	
	triphone	trigrapheme
MFCC	68.2%	79.1%
MFCC+Tandem Features	72.1%	81.2%

Table 6.8: Phone and grapheme accuracy for *MFCC* and *MFCC+Tandem Features* configurations for CD phone and grapheme acoustic models on the ***phonetic test set***.

Once it was shown that the *MFCC+Tandem Features* configuration outperformed the use of *MFCC* in isolation, the next step was to use another different corpus to tune the p and s parameters of the decoding process and apply them over the final system evaluation. The tuning of the system was made on the ***geographic training set***, whose results are presented in Table 6.9.

	Recognition accuracy	
	triphone	trigrapheme
MFCC	65.1%	78.6%
MFCC+Tandem Features	68.2%	80.2%

Table 6.9: Phone and grapheme accuracy for *MFCC* and *MFCC+Tandem Features* configurations for CD phone and grapheme acoustic models on the ***geographic training set***.

And Table 6.10 presents the final results from the tuning of the parameters on the ***geographic training set***. It shows an improvement for both acoustic configurations when using tandem features. Paired t-tests showed that all of the improvements for all of the different corpora were significant with $p < 0.001$.

	Recognition accuracy	
	triphone	trigrapheme
MFCC	66.4%	78.4%
MFCC+Tandem Features	69.4%	80.5%

Table 6.10: Phone and grapheme accuracy for *MFCC* and *MFCC+Tandem Features* configurations for CD phone and grapheme acoustic models on the ***geographic test set***.

6.4.1.2 STD results

We present in Table 6.12 the results in terms of FOM and OCC for the Architecture *1-Best* and those for the Architecture *Lattice* in Table 6.14.

	FOM	
	triphone	trigrapheme
MFCC	73.9	74.8
MFCC+Tandem Features	75.6	76.7

	OCC	
	triphone	trigrapheme
MFCC	0.72	0.76
MFCC+Tandem Features	0.75	0.77

Table 6.11: Results in terms of FOM and OCC for *MFCC* and *MFCC+Tandem Features* configurations for the *1-Best* architecture for CD phone and grapheme acoustic models for the ***geographic training set***. For all measures, higher values indicate better performance.

	FOM	
	triphone	trigrapheme
MFCC	73.5	74.4
MFCC+Tandem Features	75.5	76.4

	OCC	
	triphone	trigrapheme
MFCC	0.72	0.76
MFCC+Tandem Features	0.75	0.78

Table 6.12: Results in terms of FOM and OCC for *MFCC* and *MFCC+Tandem Features* configurations for the *1-Best* architecture for CD phone and grapheme acoustic models for the ***geographic test set***. For all measures, higher values indicate better performance.

	FOM	
	triphone	trigrapheme
MFCC	50.3	68.1
MFCC+Tandem Features	54.9	70.7

	OCC	
	triphone	trigrapheme
MFCC	0.45	0.63
MFCC+Tandem Features	0.45	0.65

Table 6.13: Results in terms of FOM and OCC for *MFCC* and *MFCC+Tandem Features* configurations for the *Lattice* architecture for CD phone and grapheme acoustic models for the ***geographic training set***. For all measures, higher values indicate better performance.

Evaluation in terms of FOM Tables 6.12 and 6.14 show that for evaluation in terms of FOM, *MFCC+Tandem features* gives the best performance for both phone- and grapheme-based acoustic models for both architectures and that the trigraphemes perform the best. Significance tests showed that for the architecture *Lattice*, the grapheme-based system gives consistent increases in performance over the phone-based system with $p < 0.001$ for *MFCC* and *MFCC+Tandem Features* configurations. They also showed that the improvement achieved with the grapheme-based system and the *MFCC* feature vectors was significant compared with the phone-based system and the *MFCC+Tandem Features* configuration with $p < 0.001$. However, there is no significant difference between the use of *MFCC* or *MFCC+Tandem Features* in each kind of acoustic models for such architecture. Significance tests over the Architecture *1-Best* show that there is no difference in

	FOM	
	triphone	trigrapheme
MFCC	47.1	64.0
MFCC+Tandem Features	51.0	65.2

	OCC	
	triphone	trigrapheme
MFCC	0.42	0.61
MFCC+Tandem Features	0.45	0.63

Table 6.14: Results in terms of FOM and OCC for *MFCC* and *MFCC+Tandem Features* configurations for the *Lattice* architecture for CD phone and grapheme acoustic models for the ***geographic test set***. For all measures, higher values indicate better performance.

using *MFCC+Tandem Features* or *MFCC* in isolation across each kind of acoustic models and no difference in using phone- or grapheme-based acoustic models for each feature extraction either. It is caused by the lexical information provided to such architecture.

Evaluation in terms of OCC We find similar patterns where the evaluation is done in terms of OCC for the Architecture *1-Best*, where the trigraphemes perform the best. Again, no significant difference exists in using any kind of feature extraction or any kind of acoustic models. It is due to the lexical information provided to such approach. However, for architecture *Lattice*, the difference in using *MFCC+Tandem Features* or *MFCC* was shown to be significant with $p < 0.001$ for both phone- and grapheme-based systems and the use of the grapheme-based system was found to be significant with $p < 0.001$ compared with the phone-based system for both types of features as well. In addition to this, the improvement achieved by the grapheme-based system with the *MFCC* feature vectors was found to be significant compared with the phone-based system with the *MFCC+Tandem Features* configuration with $p < 0.001$.

The better OCC value and worse FOM value of the *MFCC* and trigrapheme than the *MFCC+Tandem Features* and triphone was found to be insignificant in the Architecture *1-Best*.

Evaluation in terms of ATWV We present the DET curves from the ATWV performance for each of the two architectures. Figure 6.7 shows the DET curves for the Architecture *1-Best* comparing both types of feature vectors for both types of acoustic units. It is shown that *MFCC+Tandem Features* outperforms the use of *MFCC* in isolation for both CD phones and graphemes, and that the *MFCC+Tandem Features* with the CD graphemes achieves the best performance for much of the range. However, for low FA rate, the triphone configuration from *MFCC+Tandem Features* achieves the best performance.

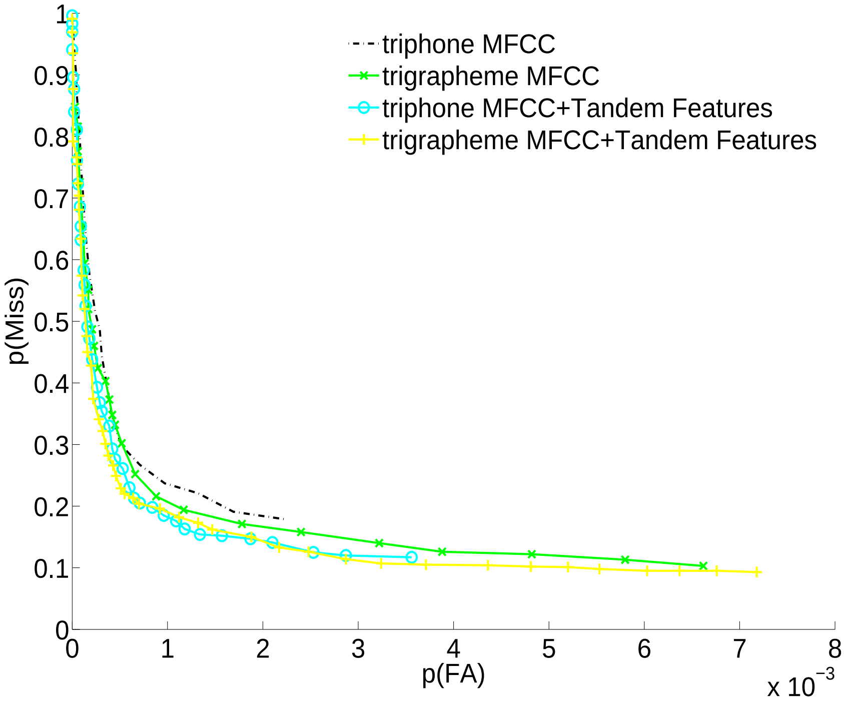

Figure 6.7: The DET curves for the CD acoustic models for *MFCC* and *MFCC+Tandem Features* configurations for the Architecture *1-Best*.

Figure 6.8 shows the DET curves for the Architecture *Lattice* comparing both sets of features for both acoustic units. Again, it is shown that *MFCC+Tandem Features* outperforms *MFCC* configuration and that the trigrapheme acoustic models achieve the best performance for the whole range. In this case, as no additional information is presented, more improvement is achieved in using the *Tandem Features* than in the Architecture *1-Best*, and greater variation is found across the different configurations as well.

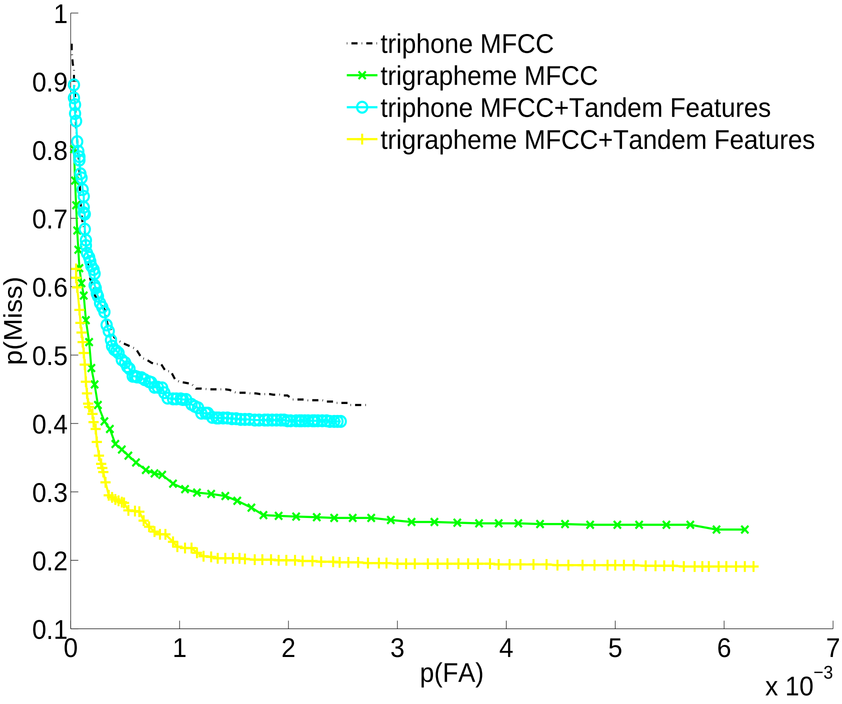

FIGURE 6.8: The DET curves for the CD acoustic models for *MFCC* and *MFCC+Tandem Features* configurations for the Architecture *Lattice*.

6.4.2 Conclusions

We have shown that by augmenting the standard *MFCC* feature vectors with the use of *tandem features* the STD performance for Spanish language for both types of acoustic units is improved for the two architectures presented in this thesis. However, the change in the acoustic models from phones to graphemes is more relevant in the STD system than the change in the feature vectors, as it is shown in the Architecture *Lattice*. The improvement achieved in using the *tandem features*, is greater for the Architecture *Lattice* due to it does not receive any additional information as the Architecture *1-Best* does. Contrary to Architecture *1-Best*, where the *MFCC+Tandem Features* configuration outperforms the *MFCC* for both phone- and grapheme-based models (though such difference was not found to be significant due to the lexical information given to such system), for the Architecture *Lattice*, the different performance of the phone- and grapheme-based units is so great that the use of *tandem features* in the phone units does not still outperform the standard *MFCC* configuration for grapheme-based units. It indicates that the grapheme-based units are still better sited in dealing with the

STD task even after a more robust feature extraction for phone-based units is used. Therefore, we conclude again that the grapheme-based STD system outperforms the phone-based one for Spanish.

6.5 Combination of phone- and grapheme-based units for Spanish STD

To evaluate the final system performance in using phone and grapheme acoustic models, we have chosen a system where the only information given to it is the training of the acoustic models. Although it has been shown that the best system performance is provided by the CD grapheme acoustic models, we propose to make an analysis about how they complement each other. Therefore, the Architecture *Lattice* explained in Section 6.3.2 was used to combine the phone- and grapheme-based sub-word units. We have chosen a simple system combination, called *detection combination* in the spirit of ROVER [97]. In this way, the hypotheses of the two systems that overlap in time and hypothesize the same keyword are merged as a single detection and hypotheses with no overlapping or different keywords overlapped in time are copied into the final system directly. In case of overlapping of the same keyword, the confidence scores of both detections are accumulated to form the new confidence score and in case of non-overlapping or overlapping of different keywords, the individual confidence score of the single detection (i.e., of each putative keyword) does not change.

In Section 6.3 it was shown that the best results for phone- and grapheme-based units were achieved with the CD units. Therefore, we have used them as baseline to show the improvement achieved with the combination of both. In following our previous work with *MFCC* and *MFCC+Tandem Features* as feature vectors, we will report the combination of both acoustic units by using both types of features.

Three different types of confidence scoring computation have been used in the combination. They have been explained in Chapter 5, and here, we refer them as follows: *Viterbi-BTBS*, *Baum-Welch* and *Posterior with Direct LM integration.*

The tuning of the system consisted of the estimation of the parameters p and s during the Viterbi decoding and the lattice search tool. For the *Viterbi-BTBS* confidence scoring, they were tuned directly for each metric and the same parameters and lattices are used for the *Baum-Welch* confidence scoring. For the *Posterior with Direct LM integration* confidence scoring, the lattices tuned for each metric for the *Viterbi-BTBS* confidence scoring were used and new p and s parameters were estimated for the lattice search tool. Viterbi- and Baum-Welch-based approaches make use of standard forward-backward processes [9] to compute the final score, so the values estimated for the Viterbi-based one perform well when are applied over the Baum-Welch approach. However, for the *Posterior with Direct LM integration* approach, where the score is computed by using the posterior probability of each frame for each acoustic unit from the MLP, these parameters need to be retrained to achieve the best performance.

6.5.1 Results and discussion

The FOM (an occurrence-weighted) and ATWV (a term-weighted) metrics have been used in the experiments in combining phone- and grapheme-based units. Paired t-tests were run to show if the differences in the system performance were found to be significant under the FOM metric.

	FOM		
	triphone	trigrapheme	combination
Viterbi-BTBS	50.3	68.1	75.9
Baum-Welch	49.7	68.2	75.9
Posterior with Direct LM integration	51.1	67.9	75.2

	ATWV		
	triphone	trigrapheme	combination
Viterbi-BTBS	0.25	0.36	0.39
Baum-Welch	0.25	0.36	0.39
Posterior with Direct LM integration	0.35	0.36	0.36

Table 6.15: Results in terms of FOM and ATWV for *MFCC* feature vectors with the three confidence scoring computations for CD phone and grapheme acoustic models and their combination for the ***geographic training set***. For all measures, higher values indicate better performance.

Evaluation in terms of FOM Table 6.16 shows that for evaluation in terms of FOM, the results achieved for all of the confidence scoring computations are better when the phone- and grapheme-based units are combined than when they are not.

	FOM		
	triphone	trigrapheme	combination
Viterbi-BTBS	47.1	64.0	72.5
Baum-Welch	46.9	63.6	72.2
Posterior with Direct LM integration	47.5	64.3	72.3

	ATWV		
	triphone	trigrapheme	combination
Viterbi-BTBS	0.19	0.32	0.32
Baum-Welch	0.20	0.31	0.32
Posterior with Direct LM integration	0.26	0.28	0.28

Table 6.16: Results in terms of FOM and ATWV for *MFCC* feature vectors with the three confidence scoring computations for CD phone and grapheme acoustic models and their combination for the ***geographic test set***. For all measures, higher values indicate better performance.

	FOM		
	triphone	trigrapheme	combination
Viterbi-BTBS	54.9	70.7	80.4
Baum-Welch	55.0	70.8	80.5
Posterior with Direct LM integration	55.5	70.4	79.0

	ATWV		
	triphone	trigrapheme	combination
Viterbi-BTBS	0.31	0.45	0.49
Baum-Welch	0.31	0.45	0.49
Posterior with Direct LM integration	0.36	0.31	0.42

Table 6.17: Results in terms of FOM and ATWV for *MFCC+Tandem Features* configuration for the three confidence scoring computations for CD phone and grapheme acoustic models and their combination for the ***geographic training set***. For all measures, higher values indicate better performance.

It is confirmed by the experiments run on the *MFCC+Tandem Features* configuration, which present a better value than the *MFCC* configuration for all of the cases in Table 6.18. Paired t-tests showed that there is significant difference comparing CD phones and graphemes with the combination of both for all of the confidence scoring computations for both *MFCC* and *MFCC+Tandem Features* configurations with $p < 0.001$. There is also significant difference between CD phones and CD graphemes for all of the confidence scoring computations for both *MFCC* and *MFCC+Tandem Features* configurations with $p < 0.001$. However, there is no significant difference when we compare each confidence scoring computation by

	FOM		
	triphone	trigrapheme	combination
Viterbi-BTBS	51.0	65.2	75.4
Baum-Welch	51.2	65.2	75.6
Posterior with Direct LM integration	51.6	65.4	74.7

	ATWV		
	triphone	trigrapheme	combination
Viterbi-BTBS	0.20	0.41	0.42
Baum-Welch	0.20	0.42	0.42
Posterior with Direct LM integration	0.28	0.29	0.34

Table 6.18: Results in terms of FOM and ATWV for *MFCC+Tandem Features* configuration for the three confidence scoring computations for CD phone and grapheme acoustic models and their combination for the ***geographic test set***. For all measures, higher values indicate better performance.

using CD phones, CD graphemes or their combination for both *MFCC+Tandem Features* and *MFCC*. And there is no significant difference when we compare each kind of feature extraction across each acoustic model and confidence score computation. It is also interesting to note that although the *Posterior with Direct LM integration* confidence scoring performs worse than the two others confidence scoring computations in the grapheme-based system for both types of feature vectors on the ***geographic training set***, it performs the best on the ***geographic test set***, although such improvement is not statistically significant. And for the combination of phone- and grapheme-based systems it is seen that the *Posterior with Direct LM integration* confidence scoring is the worst for the *MFCC+Tandem Features* configuration. It is due to the best performance of each system in isolation, which causes the combination to be less powerful. However, as stated before, paired *t*-tests showed that such difference was not found to be significant. The rest of small differences in the FOM value for the ***geographic training set*** and the ***geographic test set***, which caused that for example *Viterbi-BTBS* is worse than *BaumWelch* for trigrapheme on *MFCC* configuration for the ***geographic training set*** but the contrary is observed for the ***geographic test set*** were found to be insignificant as well.

Evaluation in terms of ATWV Tables 6.16 and 6.18 also show that for the evaluation in terms of ATWV, the results achieved with the combination of phone- and grapheme-based models outperform each model or achieve a similar result.

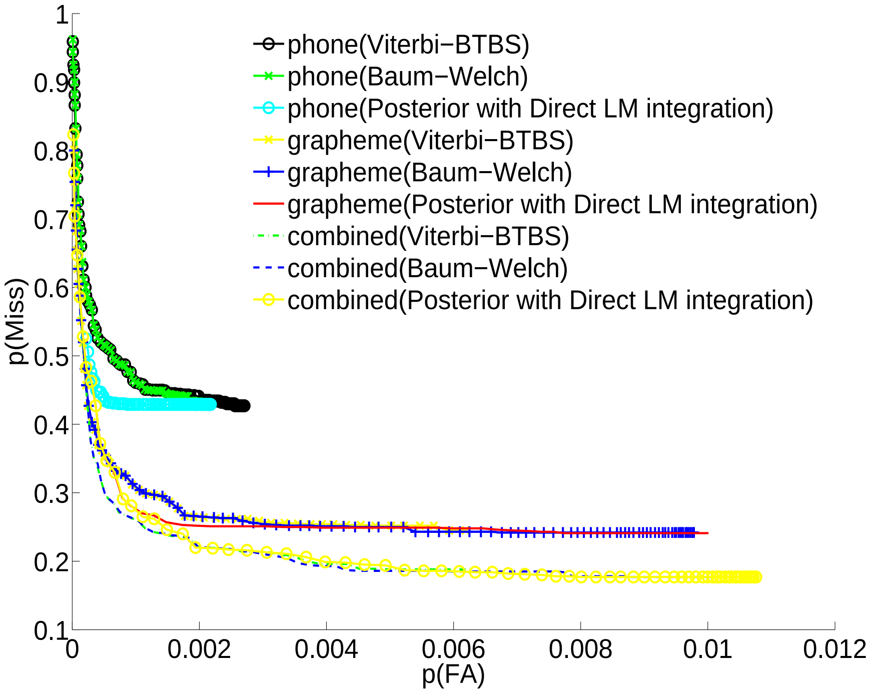

FIGURE 6.9: The DET curves for *MFCC* feature vectors for the three confidence scoring computations for the CD phone and grapheme acoustic models and their combination.

We show that the use of *MFCC+Tandem Features* did improve or did achieve the same result that the standard *MFCC* features for all of the acoustic models and confidence scoring computations. It is also interesting to note that although the *Posterior with Direct LM integration* confidence scoring performs always better for phone-based acoustic models than the two others, the contrary occurs for grapheme-based systems. It results in the worst system performance for the combination with such confidence scoring computation of phone- and grapheme-based systems. Despite this lower ATWV value, the DET curves in Figures 6.9 and 6.10 show that in using *MFCC* feature vectors, the grapheme-based *Posterior with Direct LM integration* outperforms the two others confidence scoring computations when the FA is low and achieves similar performance when the FA is high. In using *MFCC+Tandem Features* the contrary effect is observed. The *Posterior with Direct LM integration* outperforms or achieves similar performance when the Miss is low and worse performance than the two others when the FA is low. Since the *Posterior with Direct LM integration* makes use of a discriminative approach to calculate the final score, it is more reliable that the *MFCC* feature vectors benefit more from such approach than when using the *MFCC+Tandem Features* for the grapheme-based system, where the features got from such approach are added prior to the decoding process. However, it is also shown in both figures that, for much of the range, all of the confidence scoring computations perform the same in the grapheme-based system. Contrary, in the phone-based system, the improvement achieved with the *Posterior with Direct LM integration* confidence scoring is similar for both *MFCC* and *MFCC+Tandem Features* configurations due to the posterior probabilities are computed more reliable for the set of phones than for the set of graphemes. The DET curves also show that the combination in using the *Posterior with Direct LM integration* confidence scoring outperforms the others, despite the final worst ATWV value, especially for *MFCC+Tandem Features* for almost all the range, except when the FA is extremely low, where the two others perform the best and the same. For *MFCC* feature vectors, all of the combinations perform almost the same, except when the FA is extremely low, where the *Posterior with Direct LM integration* confidence scoring is worse than the two others.

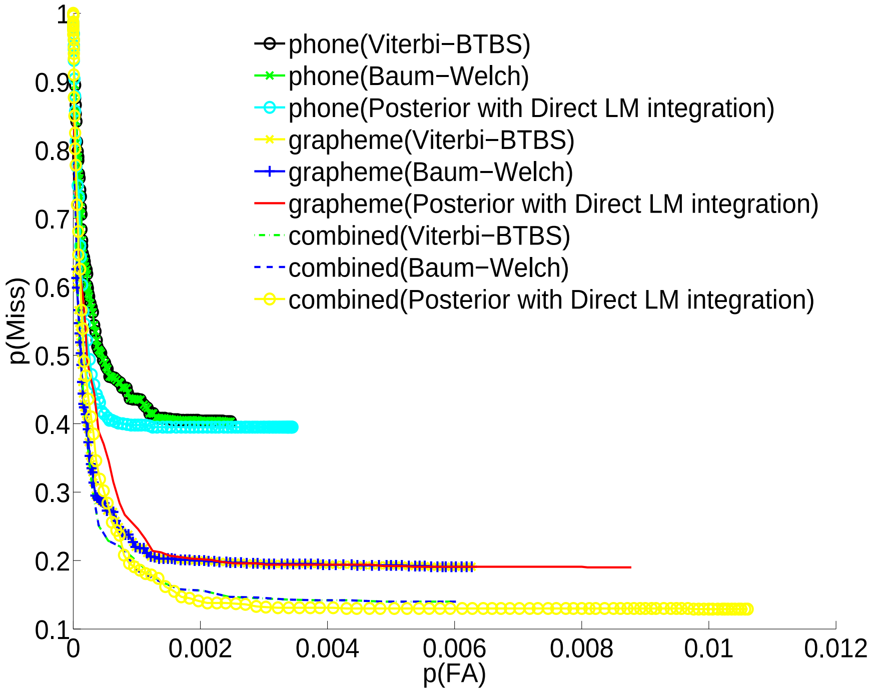

Figure 6.10: The DET curves for *MFCC+Tandem Features* configuration for the three confidence scoring computations for the CD phone and grapheme acoustic models and their combination.

Variation in the performance of a same system according to the metric Two metrics have been used to evaluate the combination of phone- and grapheme-based systems for Spanish STD. As it is shown in Tables 6.16 and 6.18, the choice of the best acoustic model, the best features along with the best confidence scoring computation, depends on the metric used in the evaluation. The *Posterior with Direct LM integration* performs the best in terms of FOM (although such difference was not found to be significant) and the worst in terms of ATWV for the grapheme-based system for both types of feature vectors. And the combination of phone- and grapheme-based systems only provides a powerful result for the *MFCC+Tandem Features* and the *Posterior with Direct LM integration* confidence scoring in terms of ATWV, whereas in terms of FOM the difference is much higher and significant for the combination for all of the cases. It reveals that both metrics follow different patterns in analyzing a same system. The FOM metric is occurrence-weighted, which means that it is computed from all of the terms, so it needs to know which term is evaluated to compute the final FOM value. In addition to this, such metric assigns a different threshold for each term when the final value is computed. Contrary, the ATWV metric is term-weighted, which means that all of the detections are considered to be independent in the formula, and there is no need in knowing which term is. Moreover, a single threshold is provided to such metric for all of the terms. In addition to this, if there is a term that has a very high occurrence, and such term has a very good performance in the system, the FOM metric and in general, all the occurrence-weighted metrics, will be biased towards such term. However, in the ATWV metric, the final value is computed for each term and then is averaged over all of the terms.

6.5.2 Analysis of the complementary behaviour of phone and grapheme acoustic units

The complementary behaviour of both acoustic models has been analyzed by grouping the keywords to form the same groups as for Keyword Spotting. Table 6.19 shows the combination along with the FOM values for both acoustic models for each group from the *MFCC* feature vectors and the *Viterbi-BTBS* confidence scoring.

From Table 6.19, it is shown that for long-length keywords, the phone-based system outperforms the grapheme-based one, while the contrary occurs for short- and

	SKG	MKG	LKG
triphone	36.9	41.4	57.0
trigrapheme	72.6	72.8	51.6
combination	75.0	75.0	68.9

Table 6.19: Results in terms of FOM for the *SKG*, *MKG* and *LKG* on the ***geographic test set*** with the *MFCC* feature vectors and the *Viterbi-BTBS* confidence scoring. Higher values indicate better performance.

medium-length keywords. Therefore, it causes that both systems provide complementary detections, and thus the combination provides an improvement in the final system. The rest of the results from the different confidence scoring computations and the *MFCC+Tandem Features* feature extraction follow the same pattern. A more exhaustive analysis was made from the individual FOM according to each keyword from the results presented in Table 6.19: Figures 6.11, 6.12 and 6.13 show that some keywords have the best FOM value with the phone-based system and others with the grapheme-based one for the three groups. Such effect is greater in short and long keywords. Again, it is shown that both systems provide complementary information, and therefore better rates are achieved in combining them.

6.5.3 Conclusions

The combination of both phone- and grapheme-based acoustic units for the Spanish STD task outperforms each of these units in isolation, even when we augment their performance varying the feature extraction process and we make use of a discriminative approach to compute the confidence score for each of the terms hypothesized. This is more noticeable in the analysis in terms of FOM. Due to some keywords have a better FOM value in the phone-based system and others benefit more from the grapheme-based system, the combination causes that the final system performance is improved. This improvement is caused by two different things: (1) there is theoretical evidence [98] that in the combination of two different systems, although one of them achieves the best performance, the patterns misclassified by both would not necessarily overlap. It causes that both systems provide complementary errors and therefore, complementary detections, making the combination improve the rates achieved by each system. The complementary

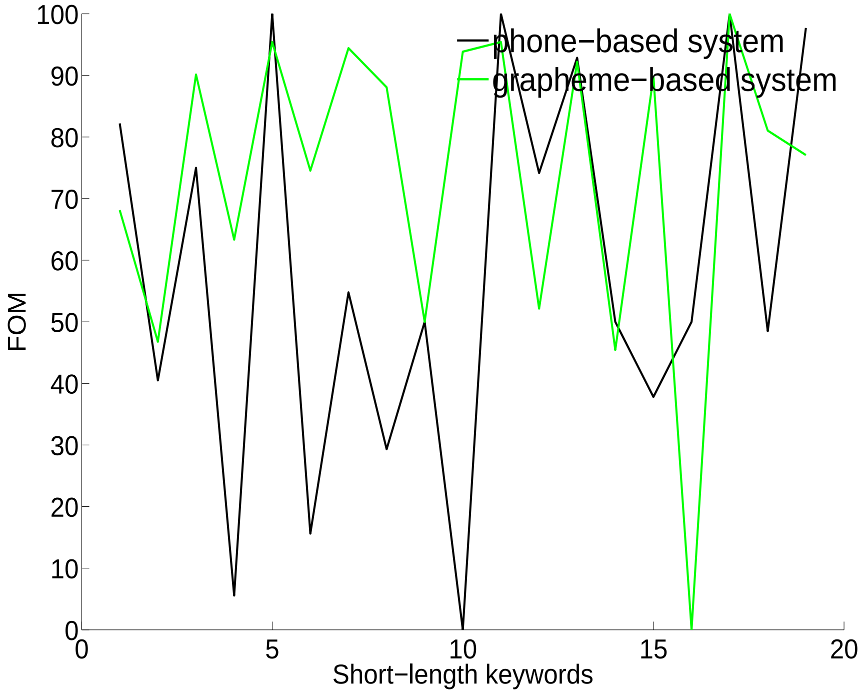

Figure 6.11: The FOM value for the phone- and grapheme-based systems for each keyword in the *SKG* (Short-length keywords group).

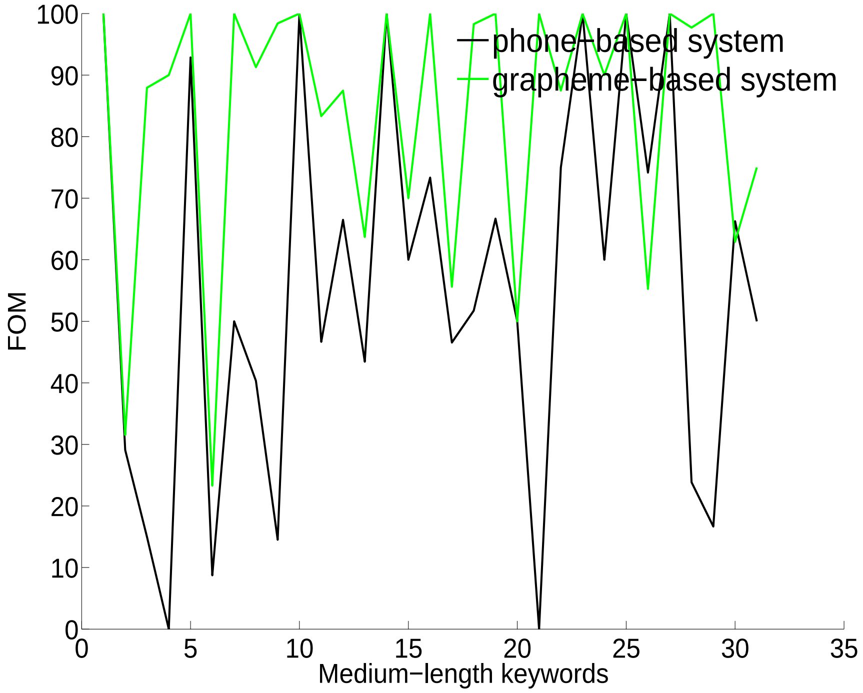

Figure 6.12: The FOM value for the phone- and grapheme-based systems for each keyword in the *MKG* (Medium-length keywords group).

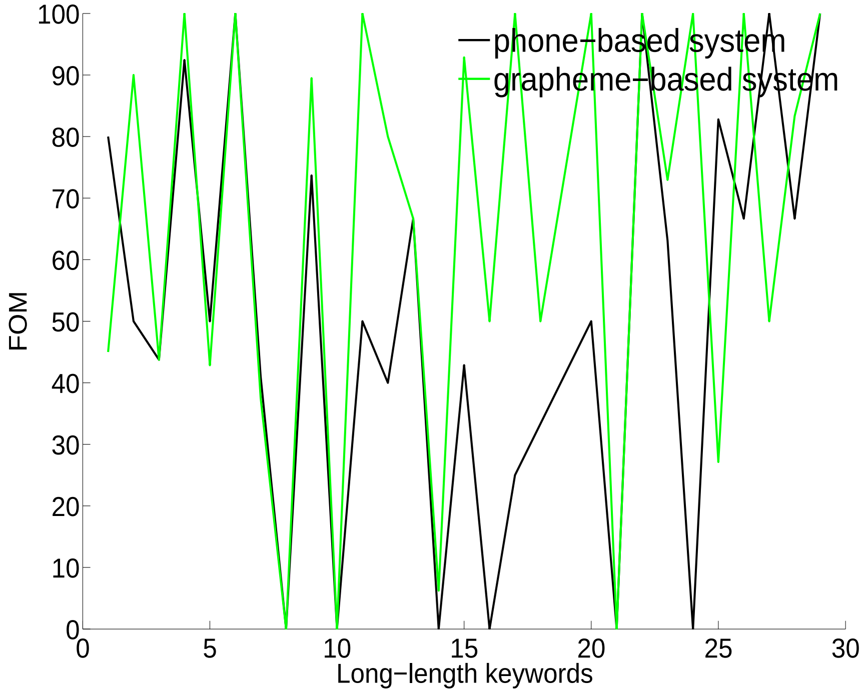

FIGURE 6.13: The FOM value for the phone- and grapheme-based systems for each keyword in the *LKG* (Long-length keywords group).

detections are caused by the different information modeled (phone and grapheme) in both systems, which causes that the search space is represented in a different way by the set of phones and the set of graphemes. (2) the probability of a term not to be proposed by the phone-based system is p_1, the probability of a term not to be proposed by the grapheme-based system is p_2, whereas the probability of a term not to be proposed by either the phone- or the grapheme-based system is always smaller than p_1 and p_2.

6.6 Summary

In this chapter we have presented and compared two different types of acoustic units to be used for Keyword Spotting and STD on the Spanish language. We have shown that grapheme-based systems outperform phone-based systems when the only information that is trained is the acoustic units used during the decoding stage. In terms of FOM, such improvement goes from 47.1 to 64.0 for the CD systems. In terms of OCC, such improvement goes from 0.42 to 0.61 for the same

CD systems. Under both metrics, the differences were found to be significant. It is a consequence of the regular letter-to-sound mapping that exists in the Spanish language and the less number of graphemes (letters) compared with the number of phones defined in the standard set of allophones for such language. However, when some lexical information is added to both systems, the differences were not found to be significant, since more information is provided to the system. It has been also shown that the grapheme-based system still outperforms the phone-based one even when we augment the standard *MFCC*-based feature extraction with the *tandem features* when the only information used is the acoustic units. In this case, in terms of FOM, the improvement goes from 51.0 using the *MFCC+Tandem Features* for the CD phones to 64.0 using the *MFCC* for the CD graphemes and to 65.2 using the *MFCC+Tandem Features* for CD graphemes. In terms of OCC, the same comparison causes that the improvement goes from 0.45 to 0.61 and 0.63 respectively. Again, such differences were significant for the paired *t*-tests. We have also shown the powerful combination of them which leads to a significant improvement in the final STD system performance when the results achieved by each unit in isolation are merged. It is due to the different system performance exhibited for some keywords in running both systems. In terms of FOM, it was shown that such improvement goes from 65.4 to 74.7 for the best system performance in isolation, corresponding to the *MFCC+Tandem Features* with the CD graphemes and the *Posterior with Direct LM integration* confidence scoring computation.

Chapter 7

Summary, contributions and future work

7.1 Summary

This thesis addressed the problem of accessing the OOV words in traditional LVCSR systems to find a set of keywords within the audio content. Keyword Spotting and STD approaches try to solve such problem. While Keyword Spotting achieves better accuracy due to the set of keywords is known prior the decoding process, STD allows to search for such keywords faster without the need of running the decoding process again. In this section we present a summary of the work presented in this thesis for both Keyword Spotting and STD.

7.1.1 Prior research and State-of-the-art

Most of the keyword spotting systems are based on HMMs where the list of keywords to search is represented by their phonetic transcription and filler models are used to absorb the non-keywords in the speech data [33, 35, 39]. Confidence measures have been also proposed to improve the performance of the final system. Some of these are based on the posterior probability or likelihood computed from the Viterbi-based decoding process [35, 37]. SVMs and NNs have been also applied to classify the putative keywords as hit or FA from a defined set of input features [38, 53].

On the other hand, STD as defined and recommended by NIST, differs from Keyword Spotting in that the list of keywords to search is unknown during the decoding process. Therefore, such decoding process must be performed by means of sub-word units (commonly in the way of a lattice) and a subsequent search within them hypothesizes the final list of keywords without making use of the audio in this stage. For this task, hybrid LVCSR-based techniques to deal with the INV words and sub-word based techniques to deal with the OOV words have been used. The main approaches which deal with those OOV words are based on a modified Levenshtein distance computation where a confusion matrix deals with the errors contained in the sub-word unit recognition process [57, 64]. This distance is computed between the sequence of sub-word units and the actual transcription of each keyword.

7.1.2 Keyword Spotting approaches

In Chapter 4, we have proposed several approaches to deal with Keyword Spotting. All of them make use of a two-level architecture design. The first level is based on the same HMM-based keyword spotting process. The difference between the approaches relies on the confidence measures presented for each. The first and second confidence measures present an isolated word speech decoding, which computes the keyword *kw* which best matches with the frames corresponding to each keyword hypothesized by the first level. It also computes a likelihood for all the keywords. With such data, the first confidence measure makes use of the keyword *kw* in isolation and the second one uses both the keyword *kw* and the likelihoods to decide if each keyword proposed by the first level is accepted or rejected. The third confidence measure is based on the output of a phone-based decoding, composed of a sequence of phones. It is matched with the actual transcription of the keyword prior to decide if each keyword proposed by the first level is accepted or rejected based on the number of errors in such sequence. The fourth confidence measure differs from the third one in that the errors are used to train a confusion matrix (computing deletion, insertion and substitution errors) prior to calculate a modified Levenshtein distance between the hypothesized and the actual sequence of phones according to each keyword which decides if each keyword proposed by the first level is accepted or rejected.

In Chapter 4, we have also presented the results achieved by the keyword spotting approaches by grouping the short-length, medium-length and long-length keywords together. We have shown that short keywords contribute more to an error than long ones. It has been also shown that the best confidence measure, which makes use of the confusion matrix, achieved the best improvement for short keywords.

7.1.3 Spoken Term Detection approaches

In Chapter 5, two different approaches have been proposed for STD. The first one is based on a discriminative approach to compute the final confidence score for each keyword proposed by the term search tool from a phone lattice. It is based on the posterior probability computed by an MLP for each phone contained in the keyword along with the language model for that keyword. The second one is based on a decision tree to reject FAs proposed by the term search tool from a phone lattice. The decision tree is built from a set of features (prosodic and lexical features) according to the keywords proposed by that tool and is used to reject those keywords that it classifies as FA.

7.1.4 Phone- and grapheme-based units for Spanish Keyword Spotting and STD

Chapter 6 explores two different sets of units for Keyword Spotting and STD on the Spanish language. Traditionally, phone-based acoustic units, widely used in LVCSR systems, have been used for such tasks. In this work we have compared keyword spotting and STD approaches by using the HMMs corresponding to both sets of units. We have also presented the combination of the output of each system from each set of acoustic units on a STD system prior to present the final output of the whole STD system.

7.1.5 Experimental results

This thesis presents the experiments on the Spanish geographical domain in the Albayzin database [26] and on the English meetings domain [30, 31].

For Keyword Spotting on the Spanish Albayzin database, we have shown that the approach based on the phone-decoding and the confusion matrix outperforms the rest of the confidence measures in terms of FOM and *RA* and *FAR* and that such approach is statistically significant in terms of FOM when it is compared with the other phone-based decoding approach and with the isolated word speech-based confidence measure which does not make use of the likelihood computation. It is also significant when no confidence measure is applied (i.e., the first level in isolation). This best approach achieves a 43% relative improvement in terms of *FAR* when it is compared with the first level in isolation and a 26% relative improvement when it is compared with the second best confidence measure (the one based on the likelihood computed from the isolated speech recognition), maintaining a similar *RA* in both cases.

For STD, we have shown that the MLP-based discriminative approach on the Albayzin database outperforms the final system performance in terms of FOM and ATWV when it is compared with traditional approaches for confidence scoring based on generative models (HMMs trained with the maximum likelihood criterion) (see Section 5.3.2). Although in terms of FOM, such approach was not found to be significant, and the improvement achieved is very low (from 47.2 to 47.5) compared with the best approach based on HMMs, the improvement achieved in terms of ATWV goes from 0.18 to 0.26, which is about 44% relative better. The DET curves also showed that such discriminative approach did improve the final STD performance for much of the range. The decision tree-based approach was found to improve the system on the English meetings domain in about 5% relative (from 0.19 to 0.20). In addition to this, the DET curves showed that such approach did improve slightly the final system performance for much of the range. However, more features are necessary to be used in the decision tree to achieve a better performance.

In comparing phone- and grapheme-based acoustic units for Spanish language, we have shown that for STD, when no more information apart from the acoustic models is presented to the STD system, the grapheme-based system outperforms significantly the phone-based one for all of the metrics presented. In terms of FOM, the improvement achieved is about 36% relative (from 47.1 to 64.0). In terms of OCC, the improvement is about 45% relative (from 0.42 to 0.61). Even when we augment the standard MFCC-based feature extraction with the *Tandem Features*, the MFCC-based grapheme system still outperforms the *MFCC+Tandem*

Features-based phone system. In terms of FOM, the improvement is about 25% relative (from 51.0 to 64.0) with the grapheme-based system. In terms of OCC, the improvement is about 35% relative (from 0.45 to 0.61). The combination of both systems was found to be significant in terms of FOM for the best feature extraction (when *Tandem Features* are merged with the standard MFCCs) and for the best confidence score computation (*Baum-Welch* confidence). In this case, about 16% (from 65.2 to 75.6) of relative improvement was achieved with the combination compared with the grapheme-based system. The difference was also found to be significant for the best grapheme-based system and its combination with the phone-based one in terms of FOM. The improvement in this case is about 14% relative (from 65.4 to 74.7). However, in terms of ATWV the combination was not found to be so strong, achieving similar rates that of the grapheme-based system for the best system, except for the *Posterior with Direct LM integration* confidence score computation and *MFCC+Tandem Features* configuration, where the combination achieved about a 17% relative improvement (from 0.29 to 0.34) compared with the grapheme-based system. Nevertheless, the DET curves showed that the combination of both systems did improve the final system for much of the range for all of the confidence score computations and feature vectors used.

To compare the different approaches presented for Keyword Spotting and STD throughout this thesis work, Figure 7.1 shows the different performance provided by the best confidence measure applied over Keyword Spotting and the 1-best- and lattice-based approaches for STD to search for the set of keywords on the Spanish ALBAYZIN database. As expected, Keyword Spotting achieves better performance than the STD-based approaches paying the vocabulary-dependent price. The second best performance is achieved by the 1-Best approach (referred to as **Lexical Access STD** in Figure 7.1), which makes use of a confusion matrix trained previously to deal with the errors appearing in the sub-word based decoding. And the combination of phone- and grapheme-based STD systems (referred to as **combination STD** in Figure 7.1), which makes use of a search in a lattice for the sequence of sub-word units representing the term, achieved the worst rate, due to it is the less-trained and a vocabulary-independent approach. However, when the Miss is low, both STD approaches perform the same.

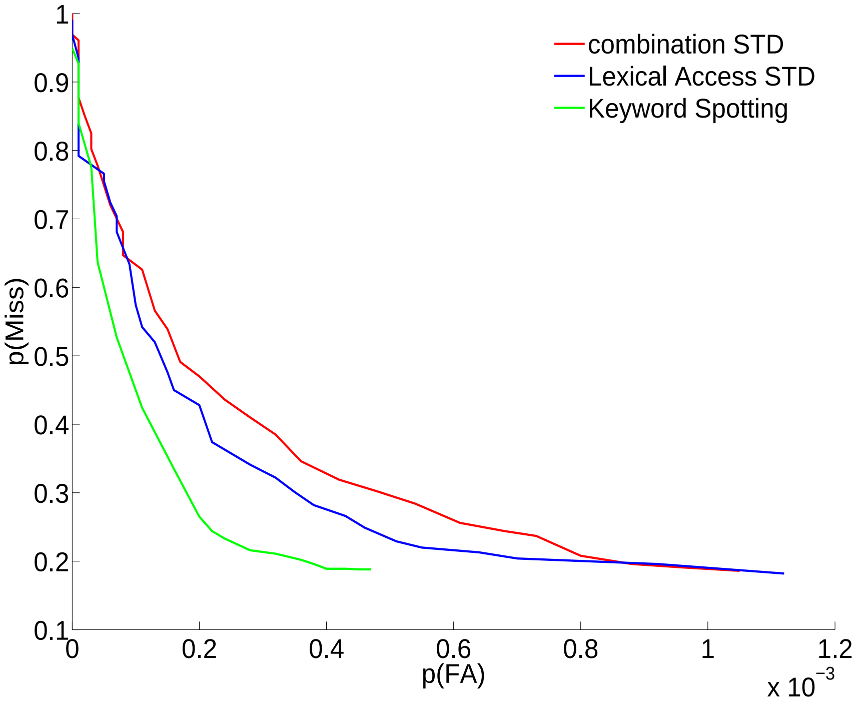

Figure 7.1: The comparison of the Keyword Spotting and STD approaches for OOV words in information retrieval.

7.2 Contributions and publications

Here, we present a highlight of the most important contributions and findings extracted throughout the thesis:

- A confidence measure based on a modified Levenshtein distance criterion and a phone-based speech decoding, which outperforms the widely used confidence measure based on the likelihood that in our case is computed during an isolated word speech decoding for a keyword spotting system (see Chapter 4).
- Application of an MLP-based technique along with the LM to compute the confidence score for the list of terms in an STD approach, which outperforms traditional HMM-based techniques (see Chapter 5).
- The use of decision trees over STD to reject the occurrences classified as FAs, which improves the STD performance (see Chapter 5).

- The comparison of grapheme- and phone-based acoustic units for Spanish STD. It was found that grapheme-based acoustic units outperform phone-based acoustic units for Spanish STD (see Chapter 6).
- The combination of grapheme- and phone-based STD systems, which outperforms each system in isolation for Spanish STD (see Chapter 6).

These contributions resulted in the following publications:

- J. Tejedor, D. Bolaños, J. Garrido and J. Colás, "Búsqueda y extracción de información en Audio Mining", *In Proceedings of IADIS International Conference WWW / INTERNET*, October 6-7, 2006. Murcia, Spain.
- J. Tejedor and J. Colás, "Spanish keyword spotting system based on filler models, pseudo N-gram language model and a confidence measure", *In Proceedings of IV Jornadas en Tecnología del Habla*, November 8-10, 2006. Zaragoza, Spain.
- J. Tejedor, R. García, M. Fernández, F.J. López-Colino, F. Perdrix, J.A. Macías, R.M. Gil, M. Oliva, D. Moya, J. Colás and P. Castells, "Ontology-based retrieval of human speech", *In Proceedings of International Workshop on web semantics (DEXA)*, September 3-7, 2007. Regensburg, Germany.
- D. Wang, J. Frankel, J. Tejedor and S. King. "Comparison of phone and grapheme-based spoken term detection", *In Proceedings of IEEE International Conference on Acoustics, Speech and Audio Processing (ICASSP)*, March-April 30-4, 2008. Las Vegas, USA.
- J. Tejedor, D. Wang, J. Frankel, S. King and J. Colás. "A comparison of grapheme and phoneme-based acoustic units for Spanish spoken term detection", *Speech Communication (SPECOM)*, November, 2008.
- J. Tejedor, S. King, J. Frankel, D. Wang, J. Colás and J. Garrido. "A novel two-level architecture plus confidence measures for a keyword spotting system", *In Proceedings of V Jornadas en Tecnología del Habla*, November 12-14, 2008. Bilbao, Spain.
- D. Wang, J. Tejedor, J. Frankel, S. King and J. Colás. "Posterior-based confidence measures for spoken term detection", *In Proceedings of IEEE International Conference on Acoustics, Speech and Audio Processing (ICASSP)*, April 19-24, 2009. Taipei, Taiwan.

7.3 Future work

Apart from the research addressed in this work, there are still enough lines that must be investigated in the future either to improve the performance of the systems presented here or to use them within a full information retrieval system. Here, we propose some possible future work as follows:

- The choice of the units inventory was found to be substantial in STD systems. Such units are used to index the audio in terms of sub-word units during the decoding process. In this thesis work, two types of acoustic units (graphemes and phones) have been explored for Spanish STD. In the future, new acoustic units will be analyzed to choose the most efficient set of them for Spanish STD (e.g. phonemes, broad classes, graphones and so on).

- In this thesis work, a simple combination of the output of the grapheme- and phone-based STD systems has shown to outperform the final STD performance. New combinations of these acoustic units and the ones referred in the previous item will be explored to improve the STD performance trying to achieve a performance as close as possible to keyword spotting approaches. Based on the results achieved in this thesis work, we propose MLP-based techniques from grapheme-based units to compute the confidence score of each occurrence hypothesized by the phone-based STD system. In the same way, MLP-based techniques from phone-based units will compute the confidence in the grapheme-based STD system.

- Features used in the decision tree building are a key point to achieve accurate and precise classification rate. In this thesis work, a preliminary set of features has been explored. However, more features such as pitch and energy will be used in the decision tree to improve the final performance. OOV words, contrary to the set of INV words used in this work, will be also selected to try the effectiveness of such approach.

- A complete Information Retrieval system will be addressed in the future. Its architecture will be based on an LVCSR system to deal with the INV words and a module to deal with the OOV words. The implementation of this module will be based on the results achieved during this thesis work, for both Keyword Spotting and STD. A description of the system has been published in [99].

Appendix A

Inventory of Spanish phones

The following table presents the inventory of phones used for the experiments on the Spanish language. An example of a word containing such phone, where the phone in the word is emphasized, is also presented.

Phone	*Example*	*Phone*	*Example*
a	cas*a*	d	*d*uero
e	ord*e*nador	D	ver*d*e
i	fol*i*o	f	gol*f*o
o	perr*o*	g	*g*alicia
u	*u*sar	G	a*g*otar
A	c*a*sa	X	rio*j*a
E	p*e*rro	j	astur*i*ana
I	cas*i*no	J/	*y*olanda
O	zarag*o*za	J	castilla*y*león
U	l*u*stro	k	*c*asa
an	alm*a*nzor	l	e*l*che
en	form*e*ntera	L	casti*ll*a
in	m*i*nero	m	*m*ano
on	m*o*ntera	n	*n*ada
un	n*u*merito	Nn	ni*ñ*o
An	m*a*nta	p	*p*ato
En	m*e*nta	r	santande*r*
In	n*i*mio	R	sie*rr*a
On	m*o*nte	s	*s*egre
Un	m*u*ndo	t	*t*rafalgar
b	*b*eso	T	*z*aragoza
B	ne*v*ada	w	pis*u*erga
T/	*ch*ina	gs	e*x*acto
N	hu*n*gría	–	–

Table A.1: Inventory of Spanish phones along with an example of each. *Phone* denotes the name of the phone and *Example* denotes the word example for each phone.

Appendix B

Inventory of English phonemes

The following table presents the inventory of phonemes used for the experiments on the English language. An example of a word containing such phoneme, where the phoneme in the word is emphasized, is also presented.

Phoneme	*Example*	*Phoneme*	*Example*
hh	*h*elp	ow	sm*o*te
ey	liber*a*tor	p	*p*ress
aa	helic*o*pter	iy	oak*y*
l	he*l*ga	jh	mytholo*g*y
ih	heinr*i*ch	m	*m*unch
s	hear*s*t	ao	*o*scars
t	injus*t*ice	eh	optom*e*trist
k	jo*k*er	dh	*th*ough
ay	libr*a*ry	y	*y*ule
ax	*a*bbreviate	uw	t*wo*
n	*n*eeds	aw	*ow*l
w	needle*w*ork	sh	*sh*ape
b	a*b*ate	er	rout*er*
ng	pi*ng*	g	*g*oths
ah	h*u*t	uh	h*oo*ray
f	*f*lush	th	*th*ong
v	*v*ain	ch	*ch*aritable
z	tribulation*s*	oy	t*oy*ing
d	succee*d*	zh	sei*z*ure
ae	*a*b	r	*r*etype

TABLE B.1: Inventory of English phonemes along with an example of each. *Phoneme* denotes the name of the phoneme and *Example* denotes the word example for each phoneme.

Appendix C

Broad class model as filler model

The following table groups the set of allophones (phones) in Spanish defined in the Appendix A into the following eight classes to form the broad class filler model used for Keyword Spotting.

Broad Class	*Phones*
opened vowels	a, an, A, An
closed vowels	i, in, I, In, u, un, U, Un, w, j
median closed vowels	e, en, E, En, o, on, O, On
deaf plosives	p, t, T/, k
sound plosives	b, B, d, D, g, G, X
deaf fricatives	f, T, s, gs
nasals	m, n, N, Nn
liquids	l, L, R, r, J, J/

Table C.1: Eight broad class models to build the broad class filler model. *Broad Class* denotes the name of each class and *Phones* denotes the name of the phones contained in each class.

Appendix D

Keywords from the Albayzin geographical domain

The following table presents the list of keywords used for the Keyword Spotting and STD tasks on the Spanish language along with the number of occurrences of each in the development and test sets and the total number of them in both sets.

Keyword	*Occ. dev*	*Occ. test*	*Keyword*	*Occ. dev*	*Occ. test*
sistema ibérico	70	32	cataluña	94	62
sevilla	27	12	mediterráneo	281	148
castilla y león	71	46	pirineos	84	48
duero	67	36	guadiana	42	24
guadalquivir	43	30	elche	2	2
belcaire	0	2	cabo de gata	16	14
santander	6	14	arosa	19	8
caudaloso	59	32	galicia	140	58
golfo de cádiz	11	4	tomelloso	0	2
antequera	0	2	murcia	22	14
aragón	52	28	gijón	2	2
segre	5	2	río ebro	58	50
puig campana	0	2	jándula	0	2
trafalgar	1	2	navarra	14	6
segura	27	14	la rioja	34	8
archipiélago	95	68	asturias	53	30
kilómetros	202	140	finisterre	30	4
guadalentín	0	6	manzanares	9	6
canarias	33	24	penibético	25	20
sistema central	43	38	madrid	135	72
zaragoza	11	10	sierra nevada	17	14
cantábrico	176	66	país vasco	60	30
formentera	0	2	mulhacén	20	8
ferrol	1	4	pico del moro almanzor	1	2
atlántico	129	72	picos de europa	25	16
macizo galaico	4	6	júcar	41	32
riotinto	0	2	pisuerga	11	10
golfo de san jorge	3	2	veleta	10	12
barcelona	29	12	cabo verde	0	2
asturiana	2	4	montes de toledo	0	6
cabo de masca	0	2	miño	49	32
columbretes	4	2	sierra de aitana	0	2
baleares	56	20	mallorca	17	8
tarifa	3	2	pedraforca	1	2
navía	0	2	océano	55	18
urbión	2	2	pico maroma	0	2
mar menor	4	4	la coruña	12	6
viana del bollo	0	2	sierra morena	25	10
valencia	91	70	andalucía	141	60
TOTAL	2872	1672	-	-	-

TABLE D.1: List of keywords from the ALBAYZIN geographical domain along with the number of occurrences (Occ.) in development (dev) and test sets for each and the total (TOTAL) number of them in both sets.

Appendix E

Evaluation metrics

The following metrics have been used throughout this thesis to evaluate the systems presented:

- The *Recognition Accuracy* and *False Acceptance Rate* are defined as follows:

$$Recognition\ Accuracy(\%) = \frac{Number\ of\ hits}{Number\ of\ true\ occurrences} * 100 \tag{E.1}$$

$$False\ Acceptance\ Rate(\%) = \frac{Number\ of\ FAs}{Number\ of\ FAs + Number\ of\ hits} * 100 \tag{E.2}$$

  where *Number of hits* is the number of correct keywords detected by the system, *Number of true occurrences* is the total number of actual keyword occurrences in the speech data and *Number of FAs* is the number of false alarms output by the system.

- The Figure-of-Merit (FOM) was defined by Rohlicek et al.[100] for the task of keyword spotting. It gives the average detection rate over the range $[1, 10]$ false alarms per keyword per hour. The FOM is computed as follows: all of the occurrences for each keyword are ranked in score order. The number of hits before the *5*'th false alarm is used to compute a single FOM value for each keyword. Next, all of the individual FOM values are averaged over the total number of occurrences. Let T the set of keywords, then for each keyword $k \in T$, let $H_k(f)$ be the number of correct detections of the keyword k allowing f FAs per hour. The FOM value for each keyword k is computed as follows:

$$FOM(k) = \frac{1}{10} \sum_{f=0}^{f=10} H_k(f) \approx H_k(5) \tag{E.3}$$

Finally, the FOM value is computed as follows:

$$FOM = \frac{\sum_{k \in T} N_{true}(k) FOM(k)}{\sum_{k \in T} N_{true}(k)} \tag{E.4}$$

where $N_{true}(k)$ represents the number of actual occurrences of the keyword k.

- The Detection Error Tradeoff (DET) curve shows the system performance from different operating points. From a set of terms (keywords) and speech data, let $N_{correct}(t)$, $N_{FA}(t)$ and $N_{true}(t)$ represent the number of correct, false alarm, and actual occurrences of term t respectively. In addition, we denote the number of non-target terms (which gives the number of possibilities for incorrect detection) as $N_{NT}(t)$. We define miss and false alarm probabilities, $P_{miss}(t)$ and $P_{FA}(t)$ for each term t as:

$$P_{miss}(t) = 1 - \frac{N_{correct}(t)}{N_{true}(t)} \tag{E.5}$$

$$P_{FA}(t) = \frac{N_{FA}(t)}{N_{NT}(t)} \tag{E.6}$$

Finally, the DET curve plots the P_{miss} against the P_{FA} by computing all of the occurrences of the terms.

- The occurrence-weighted value (OCC) was defined by NIST [8] specifically for the spoken term detection task. In this metric, a cost $C_{FA} = 0.1$ for false alarms is defined, along with a value $V = 1.0$ for correct detections. The OCC value, according to the definitions in the DET curve calculation, is computed by adding a value for each correct detection and substracting a cost for the false alarms as follows:

$$OCC = \frac{\sum_{t \in terms} [VN_{correct}(t) - C_{FA}N_{FA}(t)]}{\sum_{t \in terms} VN_{true}(t)} \tag{E.7}$$

- The Actual Term Weighted Value (ATWV) metric, defined also specifically by NIST [8] for the spoken term detection task, averages a weighted accumulation of miss and false alarm probabilities, $P_{miss}(t)$ and $P_{FA}(t)$, over all of the terms, as follows:

$$ATWV = 1 - \frac{\sum_{t \in terms} [P_{miss}(t) + \beta P_{FA}(t)]}{\sum_{t \in terms} 1} \tag{E.8}$$

where $\beta = \frac{C}{V}(P_{prior}(t)^{-1} - 1)$. The NIST evaluation scoring tool sets a uniform prior term probability $P_{prior}(t) = 10^{-4}$, and the ratio $\frac{C}{V}$ to be 0.1 with the effect that there is an emphasis placed on recall compared to precision in the ratio 10:1.

Appendix F

Conclusiones

Las principales contribuciones y conclusiones extraídas a lo largo de la tesis son las siguientes:

- La medida de confianza presentada en el capítulo 4 para el "Reconocimiento de Palabras Clave", basada en un reconocedor de alófonos y el cálculo de una distancia de Levenshtein modificada a partir de la secuencia de alófonos reconocida correspondiente a los intervalos temporales de las palabras clave propuestas por el primer nivel, mejora en un 43% relativo la tasa de falsas aceptaciones con una ligera reducción de un 1% relativo en la tasa de palabras correctas comparado con el primer nivel. Cuando esta medida de confianza se compara con la que hace uso de un reconocedor de palabras aisladas y la puntuación (confianza) obtenida durante dicho proceso, la mejora que se produce es de un 26% relativo, con un mínimo empeoramiento del 0.6% relativo en la tasa de palabras correctas.

- El uso de un perceptrón multi-capa junto con el modelo de lenguaje presentados en el capítulo 5 para estimar la puntuación (confianza) de cada palabra clave propuesta por el sistema de "Detección de Términos Hablados" mejoró en un 44% relativo el uso de las técnicas basadas en Modelos Ocultos de Markov.

- El uso de árboles de decisión para rechazar aquellas palabras clave propuestas por el sistema de "Detección de Términos Hablados" del capítulo 5 que se clasifican como falsas aceptaciones mejoró en un 5% relativo el rendimiento final del sistema.

- En el capítulo 6, las unidades acústicas basadas en grafemas mejoraron a las basadas en alófonos en el sistema de ”Detección de Términos Hablados” que únicamente contiene la información de dichas unidades para español.
- En el capítulo 6, la combinación del sistema de ”Detección de Términos Hablados” basado en grafemas y el basado en alófonos, para presentar la salida final del sistema de ”Detección de Términos Hablados” que únicamente contiene la información de dichas unidades para español, mejoró a cada sistema por separado.

Estas contribuciones han dado lugar a las siguientes publicaciones:

- J. Tejedor, D. Bolaños, J. Garrido y J. Colás, ”Búsqueda y extracción de información en Audio Mining”, *En las Actas de la Conferencia Internacional IADIS WWW / INTERNET*, 6-7 Octubre 2006. Murcia, España.
- J. Tejedor y J. Colás, ”Spanish keyword spotting system based on filler models, pseudo N-gram language model and a confidence measure”, *En las Actas de las IV Jornadas en Tecnología del Habla*, 8-10 Noviembre, 2006. Zaragoza, España.
- J. Tejedor, R. García, M. Fernández, F.J. López-Colino, F. Perdrix, J.A. Macías, R.M. Gil, M. Oliva, D. Moya, J. Colás y P. Castells, ”Ontology-based retrieval of human speech”, *En las Actas de las Jornadas Internacionales de la web semántica (DEXA)*, 3-7 Septiembre, 2007. Regensburg, Alemania.
- D. Wang, J. Frankel, J. Tejedor y S. King. ”Comparison of phone and grapheme-based spoken term detection”, *En las Actas de la Conferencia Internacional IEEE en Procesamiento acústico, de voz y de audio (ICASSP)*, 30-Marzo-4-Abril, 2008. Las Vegas, Estados Unidos.
- J. Tejedor, D. Wang, J. Frankel, S. King y J. Colás. ”A comparison of grapheme and phoneme-based acoustic units for Spanish spoken term detection”, *Speech Communication (SPECOM)*, Noviembre, 2008.
- J. Tejedor, S. King, J. Frankel, D. Wang, J. Colás y J. Garrido. ”A novel two-level architecture plus confidence measures for a keyword spotting system”, *En las Actas de las V Jornadas en Tecnología del Habla*, 12-14 Noviembre, 2008. Bilbao, España.

- D. Wang, J. Tejedor, J. Frankel, S. King y J. Colás. "Posterior-based confidence measures for spoken term detection", *En las Actas de la Conferencia Internacional IEEE en Procesamiento acústico, de voz y de audio (ICASSP)*, 19-24 Abril, 2009. Taipei, Taiwan.

Bibliography

[1] J. Garofolo, G. Auzanne, and E. Voorhees. The trec spoken document retrieval track: A success story. In *Proc. of the Text Retrieval Conference (TREC-8)*, 2000.

[2] A.G. Hauptmann and H.D. Wactlar. Indexing and search of multimodal information. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 195–198, April 1997.

[3] B. Logan, P. Moreno, J.M. Van Thong, and E. Whittaker. An experimental study of an audio indexing system for the web. In *Proc. of International Conference on Spoken Language Processing (ICSLP)*, volume 2, pages 676–679, October 2000.

[4] M. Federico. A system for the retrieval of italian broadcast news. *Speech Communication, SPECOM*, 32(1-2):37–47, September 2000.

[5] K. Koumpis and S. Renals. Content-based access to spoken audio. *IEEE Signal Processing Magazine*, 22:61–69, 2005.

[6] J.M. Huerta, S. Chen, and R.M. Stern. The 1998 carnegie mellon university sphinx-3 spanish broadcast news transcription system. In *Proc. of the DARPA Broadcast News Transcription and Understanding workshop*, March 1999.

[7] H. Meinedo, D. Caseiro, J. Neto, and I. Trancoso. Audimus.media: A broadcast news speech recognition system for the european portuguese language. In *Lecture Notes in Computer Science*, volume 2721, pages 9–17, 2003.

[8] NIST. *The spoken term detection (STD) 2006 evaluation plan.* National Institute of Standards and Technology, v10 edition, September 2006.

[9] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book (for HTK Version 3.2).* Microsoft Corp. and Cambridge University Engineering Department, 2002.

[10] H. Hermansky, D.P.W. Ellis, and S. Sharma. Tandem connectionist feature extraction for conventional hmm systems. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1635–1638, June 2000.

[11] Q. Zhu, B. Chen, N. Morgan, and A. Stolcke. On using mlp features in lvcsr. In *Proc. of International Conference on Spoken Language Processing (ICSLP)*, pages 921–924, October 2004.

[12] Q. Zhu, A. Stolcke, B.Y. Chen, and N. Morgan. Using mlp features in sri's conversational speech recognition system. In *Proc. of European Conference on Speech Communication and Technology (Eurospeech)*, pages 2141–2144, September 2005.

[13] Q. Zhu, B.Y. Chen, F. Grezl, and N. Morgan. Improved mlp structures for data-driven feature extraction for asr. In *Proc. of European Conference on Speech Communication and Technology (Eurospeech)*, pages 2129–2131, September 2005.

[14] N. Morgan, B.Y. Chen, Q. Zhu, and A. Stolcke. Trapping conversational speech: extending trap/tandem approaches to conversational speech recognition. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 537–540, May 2004.

[15] M.C. Benítez, L. Burget, B. Chen, S. Dupont, H. Garudadri, H. Hermansky, P. Jain, S. Kajarekar, N. Morgan, and S. Sivadas. Robust asr front-end using spectral-based and discriminant features: experiments on the aurora tasks. In *Proc. of European Conference on Speech Communication and Technology (Eurospeech)*, pages 429–432, September 2001.

[16] M.Y. Hwang, W. Wang, X. Lei, J. Zheng, O. Cetin, and G. Peng. Advances in mandarin broadcast speech recognition. In *Proc. of European Conference on Speech Communication and Technology (Eurospeech)*, pages 2613–2616, August 2007.

[17] M. Karafiat, F. Grezl, P. Schwarz, L. Burget, and J. Cernocky. Robust heteroscedastic linear discriminant analysis and lcrc posterior features in meeting recognition. In *Proc. of International Workshop on Machine Learning for Multimodal Interaction (MLMI)*, pages 1–4, May 2006.

[18] J. Zheng, O. Cetin, M.Y. Hwang, X. Lei, A. Stolcke, and N. Morgan. Combining discriminative feature, transform and model training for large vocabulary speech recognition. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 633–636, April 2007.

[19] A. Faria and N. Morgan. Corrected tandem features for acoustic model training. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4737–4740, April 2008.

[20] D. Johnson. *http://www.icsi.berkeley.edu/Speech/qn.html.* International Computer Science Institute (ICSI).

[21] A. Quilis. *El comentario fonológico y fonético de textos.* ARCO/LIBROS, S.A., 1998.

[22] E. Alarcos. *Gramática de la lengua española. Real Academia Española.* Colección Lebrija y Bello, Espasa Calpe, 1995.

[23] H. Hasan Ali. *Reconocimiento de 1000 palabras independiente del locutor mediante modelos ocultos de Markov.* PhD thesis, Escuela Técnica Superior de Ingenieros de Telecomunicación. Universidad Politécnica de Madrid, 1990.

[24] J. Ferreiros. *Aportación a los métodos de entrenamiento de modelos de Markov para reconocimiento de habla continua.* PhD thesis, Escuela Técnica Superior de Ingenieros de Telecomunicación. Universidad Politécnica de Madrid, 1996.

[25] G.D. Forney. The viterbi algorithm. *Proc. of IEEE*, 61(3):268–278, March 1973.

[26] 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 *Proc. of European Conference on Speech Communication and Technology (Eurospeech)*, volume 1, pages 653–656, September 1993.

[27] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. The icsi meeting corpus. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 364–367, April 2003.

[28] S. Burger, V. MacLaren, and H. Yu. The isl meeting corpus: the impact of meeting type on speech style. In *Proc. of International Conference on Spoken Language Processing (ICSLP)*, pages 301–304, September 2002.

[29] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiat, M. Lincoln, J. Vepa, and V. Wan. The ami meeting transcription system: Progress and performance. In *Proc. of International Workshop on Machine Learning for Multimodal Interaction (MLMI)*, pages 414–431, May 2006.

[30] NIST. *http://www.nist.gov/speech/tests/rt/2004-spring/index.html. Rich Transcription Spring 2004 Evaluation.* National Institute of Standards and Technology, May 2004.

[31] NIST. *http://www.nist.gov/speech/tests/rt/2005-spring/index.html. The 2005 Spring NIST Rich Transcription (RT-05S) Evaluation database.* National Institute of Standards and Technology, July 2005.

[32] I. Bazzi. *Modelling Out-of-vocabulary words for robust speech recognition.* PhD thesis, Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June 2002.

[33] R.C. Rose and D.B. Paul. A hidden markov model based keyword recognition system. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 129–132, April 1990.

[34] A.S. Manos and V.W. Zue. A segment-based wordspotter using phonetic filler models. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 899–902, April 1997.

[35] H. Cuayahuitl and B. Serridge. Out-of-vocabulary word modeling and rejection for spanish keyword spotting systems. In *Proc. of Mexican International Conference on Artificial Intelligence (MICAI)*, pages 156–165, 2002.

[36] J.G. Kim, H.Y. Jung, and H.Y. Chung. A keyword spotting approach based on pseudo n-gram language model. In *Proc. of the Conference on Speech and Computer (SPECOM)*, pages 156–159, September 2004.

[37] L. Xin and B. Wang. Utterance verification for spontaneous mandarin speech keyword spotting. In *Proc. of International Conference on Info-tec and Info-net (ICII)*, volume 3, pages 397–401, November 2001.

[38] J. Ou, C. Chen, and Z. Li. Hybrid neural-network/hmm approach for out-of-vocabulary words rejection in mandarin place name recognition. In *Proc. of International Conference On Neural Information Processing (ICONIP)*, November 2001.

[39] I. Szoke, P. Schwarz, P. Matejka, L. Burget, M. Karafiat, M. Fapso, and J. Cernocky. Comparison of keyword spotting approaches for informal continuous speech. In *Proc. of International Conference on Spoken Language Processing (ICSLP)*, pages 633–636, September 2005.

[40] T.J. Hazen and I. Bazzi. A comparison and combination of methods for oov word detection and word confidence scoring. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 397–400, May 2001.

[41] Q. Guo, Y.H. Yan, Z.W. Lin, B.S. Yuan, Q.W. Zhao, and J. Liu. Keyword spotting in auto-attendant system. In *Proc. of International Conference on Spoken and Language Processing (ICSLP)*, volume 2, pages 1050–1052, October 2000.

[42] C. Yining, L. Jing, Z. Lin, L. Jia, and L. Runsheng. Keyword spotting based on mixed grammar model. In *Proc. of International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 425–428, May 2001.

[43] M-C. Silaghi and H. Bourlard. Iterative posterior-based keyword spotting without filler models. In *Proc. of IEEE workshop Automatic Speech Recognition and Understanding (ASRU)*, December 1999.

[44] H. Bourlard and N. Morgan. *Connectionist speech recognition.* Kluwer Academic Publishers, 1994.

[45] L.F. Lamel, J-L. Gauvain, and M. Eskenazi. Bref, a large vocabulary spoken corpus for french. In *Proc. of European Conference on Speech Communication and Technology (Eurospeech)*, pages 505–508, September 1991.

[46] P. Yu and F. Seide. A hybrid word/phoneme-based approach for improved vocabulary-independent search in spontaneous speech. In *Proc. of International Conference on Spoken Language Processing (ICSLP)*, pages 635–643, 2004.

[47] M. Padmanabhan, G. Ramaswamy, B. Ramabhadran, P.S. Gopalakrishnan, and C. Dunn. Voicemail corpus part i (ldc98s77) and part ii (ldc2002s35). In *http://www.ldc.upenn.edu*, 1998.

[48] P. Schwarz, P. Matejka, and J. Cernocky. Towards lower error rates in phoneme recognition. In *Proc. of International Conference on Text, Speech and Dialogue (TSD)*, pages 465–472, September 2004.

[49] D.A. Dahl, M. Bates, M. Brown, W. Fisher, K.H. Smith, D. Pallet, C. Pao, A. Rudnicky, and E. Shriberg. Expanding the scope of the atis task: The atis-3 corpus. In *Proc. of DARPA Human Language Technology Workshop*, pages 43–48, March 1994.

[50] R. El Méliani and D. O'Shaughnessy. Accurate keyword spotting using strictly lexical fillers. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages 907–910, April 1997.

[51] T. Schaaf and T. Kemp. Confidence measures for spontaneous speech recognition. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages 875–878, April 1997.

[52] S. Cox and R. Rose. Confidence measures for the switchboard database. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 511–514, May 1996.

[53] Y. Ben Ayed, D. Fohr, J. P. Haton, and G. Chollet. Keyword spotting using support vector machines. In *Proc. of International Conference on Text, Speech and Dialogue (TSD)*, pages 285–292, November 2002.

[54] Y. Ben Ayed, D. Fohr, J.P. Haton, and G. Chollet. Confidence measures for keyword spotting using support vector machines. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 588–591, April 2003.

[55] L. Ferrer and C. Estienne. Improving performance of a keyword spotting system by using a new confidence measure. In *Proc. of European Conference on Speech Communication and Technology (Eurospeech)*, pages 2561–2564, September 2001.

[56] J. Junkawitsch, L. Neubauer, H. Hge, and G. Ruske. A new keyword spotting algorithm with pre-calculated optimal thresholds. In *Proc. of International Conference on Spoken Language Processing (ICSLP)*, pages 2067–2070, October 1996.

[57] A. Amir, A. Efrat, and S. Srinivassan. Advances in phonetic word spotting. In *Proc. of International Conference on Information and Knowledge Management (CIKM)*, pages 580–582, November 2001.

[58] 1998 hub-4 broadcast news evaluation english test material. In *http://www.ldc.upenn.edu*, 1998.

[59] S. Dharanipragada and S. Roukos. A multistage algorithm for spotting new words in speech. *IEEE Transactions on Speech and Audio Processing*, 10(8): 542–550, November 2002.

[60] P. Yu and F. Seide. Fast two-stage vocabulary independent search in spontaneous speech. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 481–484, March 2005.

[61] F. Seide, P. Yu, C. Ma, and E. Chang. Vocabulary-independent search in spontaneous speech. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 253–256, May 2004.

[62] J. Scott, J. Wintrode, and M. Lee. Fast unconstrained audio search in numerous human languages. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 77–80, April 2007.

[63] S.J. Young, N.H. Russel, and J.H.S. Thornton. *Token passing: A simple connectionist model for connected speech recognition systems.* Cambridge University Engineering Department, 1989.

[64] K. Thambiratnam and S. Sridharan. Rapid yet accurate speech indexing using dynamic match lattice spotting. *IEEE Transactions on Audio and Speech Processing*, 15(1):346–357, January 2007.

[65] P. Gao, J. Liang, P. Ding, and B. Xu. A novel phone-state matrix based vocabulary-independent keyword spotting method for spontaneous speech. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 425–428, April 2007.

[66] J. Pinto, I. Szoke, S.R.M. Prassana, and H. Hermansky. Fast approximate spoken term detection from sequence of phonemes. In *Proc. of Speech search workshop at SIGIR*, July 2008.

[67] J. Tejedor, S. King, J. Frankel, D. Wang, J. Colás, and J. Garrido. A novel two-level architecture plus confidence measures for a keyword spotting system. In *Proc. of V Jornadas en Tecnología del Habla*, November 2008.

[68] Z. Rivlin, M. Cohen, V. Abrash, and T. Chung. A phone-dependent confidence measure for utterance rejection. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 515–517, May 1996.

[69] J.G.A Dolfing and A. Wendemuth. Combination of confidence measures in isolated word recognition. In *Proc. of International Conference on Spoken Language Processing (ICSLP)*, pages 5–8, December 1998.

[70] N. Moreau and D. Jouvet. Use of a confidence measure based on frame level likelihood ratios for the rejection of incorrect data. In *Proc. of European Conference on Speech Communication and Technology (Eurospeech)*, pages 291–294, September 1999.

[71] J. Tejedor and J. Colás. Spanish keyword spotting system based on filler models, pseudo n-gram language model and a confidence measure. In *Proc. of IV Jornadas en Tecnología del Habla*, pages 255–260, November 2006.

[72] L. Fissore, P. Laface, G. Micca, and R. Pieraccini. Lexical access to large vocabularies for speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(8):1197–1213, August 1989.

[73] J. Tejedor, R. García, M. Fernández, F.J. López-Colino, F. Perdrix, J.A. Macías, R.M. Gil, M. Oliva, D. Moya, J. Colás, and P. Castells. Ontology-based retrieval of human speech. In *Proc. of International workshop on web semantics (DEXA)*, pages 485–489, September 2007.

[74] F. Wessel, R. Schluter, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9(3):288–298, March 2001.

[75] D. Wang, J. Tejedor, J. Frankel, S. King, and J. Colás. Posterior-based confidence measures for spoken term detection. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2009.

[76] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiat, M. Lincoln, I. McCowan, D. Moore, V. Wan, R. Ordelman, and S. Renals. The 2005 ami system for the transcription of speech in meetings. In *Proc. of International Workshop on Machine Learning for Multimodal Interaction (MLMI)*, pages 450–462, May 2006.

[77] Y. Liu, N. Chawla, M. Harper, E. Shriberg, and A. Stolcke. A study in machine learning from imbalanced data for sentence boundary detection in speech. *Computer Speech and Language*, 20(4):468–494, October 2006.

[78] B. Wrede and E. Shriberg. Spotting hotspots in meetings: Human judgements and prosodic cues. In *Proc. of European Conference on Speech Communication and Technology (Eurospeech)*, pages 2805–2808, September 2003.

[79] Y. Liu, E. Shriberg, and A. Stolcke. Automatic disfluencies identification in conversational speech using multiple knowledge sources. In *Proc. of European Conference on Speech Communication and Technology (Eurospeech)*, pages 957–960, September 2003.

[80] S. Goldwater, D. Jurafsky, and C. Manning. Which words are hard to recognize? prosodic, lexical and disfluency factors that increase asr error rates. In *Proc. of Anual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL:HLT)*, pages 380–388, June 2008.

[81] V. Laurikari. *http://laurikari.net/tre*, December 2006.

[82] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer. Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, February 2002.

[83] N. Japkowic and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis Journal*, 6(5):429–450, October 2002.

[84] M. Killer, S. Stuker, and T. Schultz. Grapheme based speech recognition. In *Proc. of European Conference on Speech Communication and Technology (Eurospeech)*, pages 3141–3144, December 2003.

[85] S. Kanthak and H. Ney. Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 845–848, May 2002.

[86] S. Kanthak and H. Ney. Multilingual acoustic modeling using graphemes. In *Proc. of European Conference on Speech Communication and Technology (Eurospeech)*, pages 1145–1148, May 2003.

[87] B. Mimer, S. Stuker, and T. Schultz. Flexible decision trees for grapheme-based speech recognition. In *Proc. of the Conference Elektronische Sprachsignalverarbeitung (ESSV)*, 2004.

[88] P. Charoenpornsawat, S. Hewavitharana, and T. Schultz. Thai grapheme-based speech recognition. In *Proc. of Human Language Technologies NAACL*, pages 17–20, June 2006.

[89] J. Dines and M. Magiami-Doss. A study of phoneme and grapheme based context-dependent ASR systems. In *Proc. of Machine Learning and Multimodal Interaction (MLMI)*, pages 215–226, June 2007.

[90] R.A. Cole, M. Fanty, M. Noel, and T. Lander. Telephone speech corpus development at CSLU. In *Proc. of International Conference on Spoken Language Processing (ICSLP)*, pages 1815–1818, September 1994.

[91] P.J. Price, W. Fisher, and J. Bernstein. A database for continuous speech recognition in a 1000 word domain. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 651–654, May 1998.

[92] B. Chen, O. Cetin, G. Doddinton, N. Morgan, M. Ostendorf, T. Shinozaki, and Q. Zhu. A CTS task for meaningful fast-turnaround experiments. In *Proc. of Rich Transcription Fall Workshop*, November 2004.

[93] M. Magiami-Doss, S. Bengio, and H. Bourlard. Joint decoding for phoneme-grapheme continuous speech recognition. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 177–180, May 2004.

[94] M. Magiami-Doss, T. A. Stephenson, H. Bourlard, and S. Bengio. Phoneme-grapheme based automatic speech recognition system. In *Proc. of IEEE workshop on Automatic Speech Secognition and Understanding (ASRU)*, pages 94–98, December 2003.

[95] J. Tejedor, D. Wang, J. Frankel, S. King, and J. Colás. A comparison of grapheme and phoneme-based acoustic units for spanish spoken term detection. *Speech Communication, SPECOM*, 50(11-12):980–991, November 2008.

[96] D. Wang, J. Frankel, J. Tejedor, and S. King. Comparison of phone and grapheme-based spoken term detection. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4969–4972, March–April 2008.

[97] J.G. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover). In *Proc. IEEE workshop Automatic Speech Recognition and Understanding (ASRU)*, pages 347–354, December 1997.

[98] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on pattern analysis and machine intelligence*, 20(3):226–239, March 1998.

[99] J. Tejedor, D. Bolaños, J. Garrido, and J. Colás. Búsqueda y extracción de información en audio mining. In *Proc. of IADIS International Conference WWW / INTERNET*, October 2006.

[100] J.R. Rohlicek, W. Russell, S. Roukos, and H. Gish. Continuous hidden Markov modeling for speaker-independent word spotting. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 627–630, 1989.