UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Integration of Speech Biometrics in a Phone Payment System: Text-Independent Speaker Verification

A Degree Thesis
Submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by
Anna Barón Garcia

In partial fulfilment of the requirements for the degree in
AUDIOVISUAL SYSTEMS ENGINEERING

Advisor: Javier Hernando Pericas

Barcelona, September 2016

Abstract

Nowadays, the integration of biometrics in security systems is a prominent research and application field. Also, it is clear that speech is the most common form of communication, which makes a swell candidate. While using speech as a biometric, one could say there are two types of systems that should be analyzed: those systems which do know what the speaker is going to say upon verification and those that do not. This degree thesis offers an overview of both systems, focusing on those that do not know what the speaker is going to say beforehand, also known as text-independent systems. To be able to determine which would be the best approach to integrate speech biometrics into a security system, both types of systems are compared; and two methodologies are also analyzed for the text-independent system. To conclude, one of those methodologies is implemented in a software library which allows the creation a text-independent speaker verification system.

Resum

En l'actualitat, la integració de biometries en els sistemes de seguretat és una branca d'investigació i aplicacions prominent. A més a més, la veu és un dels mitjans més comuns de comunicació, cosa que fa que sigui una bona candidata per a aquests sistemes. Si prenem la parla com a biometria, es pot dir que hi ha dos tipus de sistemes bastant diferenciats a analitzar: aquells sistemes els quals saben el que dirà la persona que s'intenta verificar i aquells que no saben el que dirà. Aquest treball ofereix una visió amplia dels dos tipus de sistemes, centrant-se en els sistemes on no es sap el que es dirà, també coneguts com sistemes de text independent. Per decidir quin seria la millor manera d'integrar la parla com a biometria en un sistema de seguretat, es comparen ambdós sistemes i, en el cas del sistema de text independent, es comparen també dues metodologies diferents. Per acabar, s'implementa una d'aquestes metodologies a unes llibreries de software per dur a terme un sistema de verificació de locutor amb text independent.

Resumen

En la actualidad, la integración de biometrías en los sistemas de seguridad es una rama de investigación y de aplicaciones prominente. Además, está claro que la voz es el medio más común de comunicación y es por eso que es una buena candidata. Usando el habla como biometría, se podría decir que hay dos tipos de sistemas diferentes a analizar: aquellos sistemas que saben de antemano aquello que va a decir el locutor que intenta verificarse y aquellos que no lo saben. Este trabajo ofrece una visión amplia de los dos tipos de sistemas, centrándose en los sistemas donde aquello que se va a decir no se sabe, también conocidos como sistemas de texto independiente. Para decir cuál sería la mejor manera de integrar el habla como biometría en un sistema de seguridad se comparan ambos sistemas y, en el caso del sistema de texto independiente, se comparan también dos metodologías diferentes. Para finalizar, se implementa una de estas últimas en unas librerías de software para poder llevar a cabo un sistema de verificación de locutor de texto independiente.

Acknowledgements

First of all, I would like to express my sincere gratitude to my tutor Javier Hernando for always providing insightful answers and expertise, and for pushing me to do better during these past 8 months. I would also like to praise Miquel, whose help was essential to conclude this thesis.

I would like to thank my family and friends for encouraging me all these years, and dedicate this to Tivy and Marc, to whom I wanted to prove you can go through anything and come out victorious in the end. And to Irina, whose courage and determination are rightfully contagious.

And last but not least, I would like to wholeheartedly thank Noe for her unconditional support and love, and for being by my side every step of the way.

Revision history and approval record

Revision	Date	Purpose
0	18/07/2016	Document creation
1	19/07/2016	Document revision
2	14/09/2016	Document revision
3	23/09/2016	Final version

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Anna Barón Garcia	anna.baron.garcia@alu-etsetb.upc.edu
Javier Hernando	javier.hernando@upc.edu
Miquel Angel India	miquelindia90@gmail.com

Written by:		Reviewed and approved by:	
Date	23/09/2016	Date	24/09/2016
Name	Anna Barón	Name	Javier Hernando
Position	Project Author	Position	Project Supervisor

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Project background

There has been a lot of proposals to which biometric measurements should be used for recognition and verification systems –e.g. fingerprints, face recognition, voice- and all of them have their pros and cons according to their accuracy and implementation costs. It is believed that speech biometrics have stood out as a compelling biometric due to two factors. Firstly, speech is a natural signal to produce and users do not consider providing a speech sample for authentication as an intrusive step, in contrast to providing fingerprint recognition where studies have shown that users feel slightly threatened by. Also, telephone systems provide a network of sensors for obtaining and delivering speech signal, so there is no need for special signal transducers or to implement a new network for the system access points.

Specifically, verification systems based on speech have made a greater impact in the commercial sector rather than identification systems. The main difference between the two is that a verification system determines whether or not an unknown voice is from a particular enrolled speaker –a claimed identity is confirmed- and an identification system associates an unknown voice with one of the enrolled speakers –there is no claim nor confirmation, just association-.

As a general overview, speech verification systems can be text-constrained –i.e. the user says a password the system is programmed specifically for- or text-independent with any spoken utterances. Regarding security issues, text-dependent verification systems offer better results, but are less socially secure. For example, someone could be repeating a 'PIN number' (the text constrained) through the phone and be overheard, which could lead to someone else recording said number. Even though both of systems are fit for some applications or others, this degree thesis will focus on the study of both systems' performance and the implementation of a text-independent speaker verification system.

1.2 Goals

The purpose of this thesis is to study and implement a proper system that treats speech biometrics in a way that can be used for a security application. It presents a general approach to the methodology used in both the text-constrained and text-independent analysis of speech verification systems, and explains the integration of said biometrics into an application. There are two clear goals: study the methodology that is used for applying speech biometrics in security applications, and the implementation of methods that allow such applications.

In detail, the main goals are:

- To study the historical perspectives and current research in the field of speaker verification systems: text-dependent and text-independent.
- To analyze the structure of a text-dependent speaker verification system software, the SpkIdAPI.
- To analyze the structure of a text-independent speaker verification system software: the ALIZE / LIA_RAL software.
- To integrate text-independent methods for speaker verification into the SpkIdAPI software.
- To analyze the results on different methodology and software and determine the best implementation for a text-independent speaker verification system.

1.3 Overview

This degree thesis is organized as follows:

- Chapter 1: offers a brief introduction on the subject at hand.
- Chapter 2: is an overview of the state-of-the-art methodology for the speaker verification process: the general system, the GMMs, the i-vector and scoring.
- Chapter 3: describes the methodology and principal steps of the speaker verification process, along with the software used to apply said methodology.
- Chapter 4: is a description of the experiments that have been carried out and the results obtained.
- Chapter 5: is the estimated budget for the project.
- Chapter 6: concludes the bachelor's thesis and presents some future lines of work.

Chapter 2

State of the art

Speech biometrics and speaker verification systems have been a prominent field of study in the past few decades. With the ever-growing computation capability of computerized machines, the complexity of the methodology proposed in research has been also increasing and expanding its limits. This chapter presents a humble review of the research currents and the definition of speaker verification systems.

2.1 Speech Biometrics

As mentioned in the introduction, voice is the most natural way of communication and, consequently, it has a high user and social acceptance. As stated, speaker recognition can use different channels such as the telephone or the microphone, making speech biometrics very attractive from the point of view of security systems, given the fact that telephones and microphones are very accessible technologies.

In commercial applications, speaker verification is generally used in combination with voice identification. Even though combining both biometric systems is a powerful approach for a security application, it should be noted that speaker identification normally requires more training than other biometrics and can suffer from reverberation, illnesses or background noises.

2.2 Speaker Verification Systems

A speaker verification system is composed of two distinct phases: the training phase and the testing phase. In Fig. 2.1 there is a modular representation of the training phase: extract the parameters from the signal and then use them for statistical modeling to obtain a representation of the speaker (speaker model). In 2.2 there is a modular scheme of the test phase, in which a signal with a claimed identity is introduced, it has its speech parameters extracted and they are compared with the aforementioned speaker model of the claimed identity. The system obtains scores from that comparison and decides to accept or reject the claim.

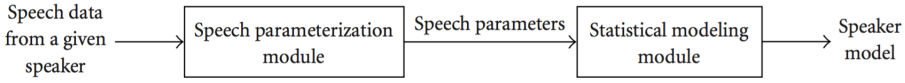

Figure 2.1: Training phase and modeling. [1]

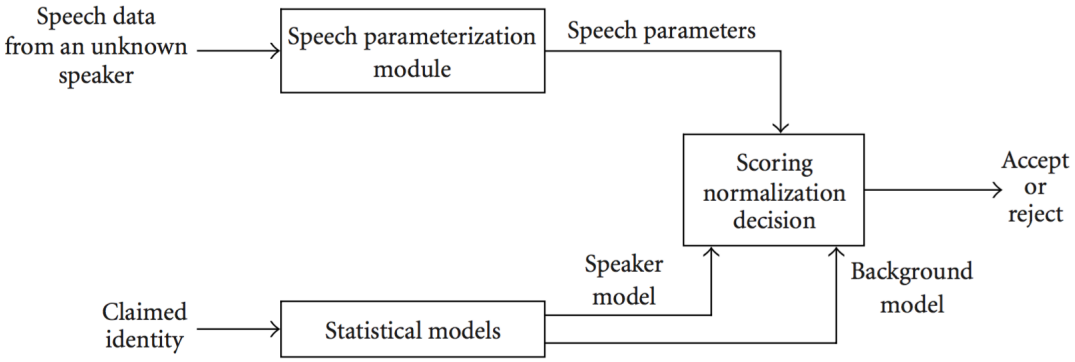

Figure 2.2: Testing phase and scoring. [1]

2.3 Text-Independent Speaker Verification

A text-independent speaker verification system would be a one where the text is not constrained –i.e. the text that the speaker is about to say is not known beforehand-. During the last decade, several approaches to text-independent speaker verification have been studied and developed, due to its attractive nature, both from a methodology perspective and a market perspective.

As seen in Fig. 2.1, after the speech parametrization is done, several techniques are used to estimate speaker models. Most common ones are: Gaussian Mixture Models (GMM), Support Vectors Machines (SVM) and i-vector Estimation.

2.3.1 Gaussian Mixture Models

Gaussian Mixture Model (GMM) are models where a sum of Gaussian probability distributions are used to model each speaker. GMM can be viewed as a representation of the various acoustic classes that make up the sounds of the speaker. Each class represents possibly one speech sound or a set of speech sounds [2].

Therefore, GMMs are a linear combination of Gaussian probability distribution functions (pdfs). They have the capability to form an approximation to an arbitrary pdf for a large number of mixture components. It is proven that a finite number of Gaussians is sufficient to form a smooth approximation to the pdf and each speech cluster is represented by a Gaussian. To estimate GMM parameters the maximum likelihood estimation (MLE) can be used, and for a large set of training feature vectors, it is also proven that the model estimated converges.

To solve this, the EM algorithm is performed, which iteratively refines the GMM parameters to increase the likelihood of the estimated model for the feature vectors.

The GMM-based speaker verification has a particular problem: session variability. The term session variability refers to all the phenomena which cause two recording of a given speaker to sound different from each other, and thus affecting the verification.

2.3.2 Joint Factor Analysis

It is due to the problem of session variability in GMMs that several approaches to solve the issue were presented. A model referred to as Joint Factor Analysis (JFA) of speaker and channel variability was introduced [3]. It treats channel effects as continuous rather than discrete and it exploits correlations between Gaussians in modeling the speaker variability.

It also has a main drawback: it is mathematically and computationally demanding.

2.3.3 Support Vector Machines

Support vector machines (SVM) are supervised binary classifiers [4], based on the idea of finding, from a set of supervised learning examples, the best linear separator for distinguishing between the positive examples and negative examples.

In the past few years of research, the application of SVM in the GMM supervector space has yield interesting results. The combination between the JFA and the SVMs for speaker verification has also been a common approach. It consists in directly using the speaker factors estimated with JFA as input of the Support Vector Machines.

2.3.4 Total Variability Space

In recent experiments [4] it has been proved that channel factors estimated using JFA, which are supposed to model only channel effects, also contain information about speakers. Based on this, there is a proposed a speaker verification system based on factor analysis as a feature extractor. The factor analysis is used to define a new low-dimensional space named Total Variability Space. In this new space, a given speech utterance is represented by a new vector named total factor or identity vector.

In this approach, the identity vectors or 'i-vectors' are super-vector representations on a different plane (the Total Variability plane) that are unique for each speaker-utterance.

2.3.5 Scoring

- Cosine Distance [4]: It is a scoring technique which directly uses the value of the Cosine Kernel between the target speaker i-vector and the test i-vector as a decision score.

  One of the advantages of this scoring technique is that no target speaker enrollment is required, unlike for Support Vector Machines, where the target speaker-dependent supervector needs to be estimated in an enrollment step.

  The use of the Cosine Kernel as a decision score for speaker verification makes the process faster.

- Probabilistic Linear Discriminant Analysis [5]: The current PLDA methodology for speaker verification is a two-step process: Firstly, given a development database of labelled data, make an ML point estimate of the PLDA model. After that, given the unlabelled data of a detection trial, plug in the above point estimate to compute the posterior for the target vs non-target trial.

  The PLDA model assumes statistical independence among enrollment i-vectors, which may be difficult to achieve in practice. It should be noted that enrollment i-vectors from a given target speaker might share common attributes like acoustic content, transmission channel etc., thus invalidating the independence assumption. Multiple i-vectors can be integrated directly into the PLDA model.

2.4 Text-Dependent Speaker Verification

A text-dependent speaker verification system is the one in which the text said upon verification is known beforehand. The Hidden Markov Model is the oldest approach in research and models each speaker as a Markov Model [6].

The Hidden Markov Model (HMM) is a popular stochastic model for modeling both the stationary and transient properties of a signal.

The Hidden Markov Model approach is often used on the phoneme level, where one HMM is used to model a single phoneme with a fixed set of states, since HMMs capture well the short periods of rapid change in pronouncing sounds.

The structure of a HMM is composed by a set of states with transitions between each state. For each transition from a state, a probability of taking that transition is assigned. These probabilities sum one.

They are essentially stochastic finite state machines which output a symbol each time they depart from a state. The symbol is probabilistically determined; each state contains a probability distribution of the possible output states.

The sequence of states is not directly observable, therefore they are called hidden. Generally speaking, the HMM is a state machine using the input audio frames to determine the next state.

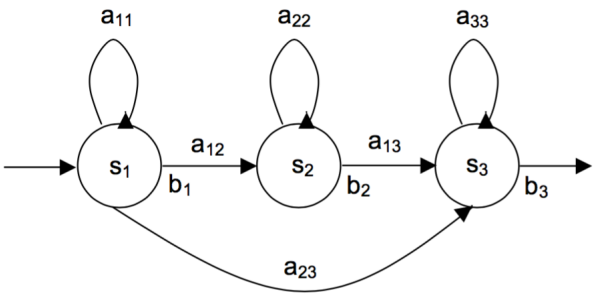

Figure 2.3: HMMs (3 states) [6]

Chapter 3

Methodology

Having established the research context, this chapter contains a description of the methods studied and used in the experiments of this thesis.

3.1 Feature Extraction

The first part of the system consists in transforming the speech signal waveform into a representation that is less redundant and more compact: a vector of acoustic features. For this, cepstral features are ones most commonly used in speaker recognition systems. In Fig. 3.1 a scheme of how to obtain the Mel-Frequency Cepstral Coefficients is shown.

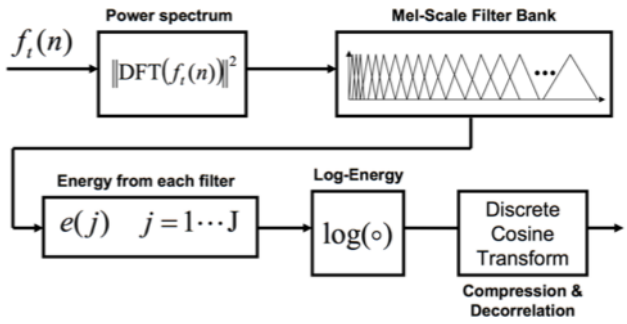

Figure 3.1: Mel-Frequency Cepstral Coefficients extraction system

At first, the speech signal analysis is done locally by applying a window whose duration is shorter than the signal, and it is moved along to signal until the end is reached. This is done as so the speech signal (in its nature, rapidly changing) can be assumed stationary if divided in segments. The length and type of the window may vary in different experiments, but 20 to 30 ms and Hamming or Hanning windows are often used. For every windowed signal, the modulus of FFT is applied is extracted, and a power spectrum is obtained.

Even though it is the spectrum that is computed, the interest lays in the envelope of the spectrum. This is why it is multiplied by a filterbank; a series of band pass filters that are multiplied one by one with the spectrum to get an average value in a particular frequency band. Mel-Scale Filters are used in speech recognition,

due to their accuracy to mimic the perceptive pitch, modeling the actual non-linear perception of frequencies in the human auditory system. The conversion between the Mel scale and the Frequency scale is given by the equations 3.1 and 3.2.

$$Mel(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{3.1}$$

$$F(mel) = 700\left(10^{\frac{mel}{2595}} - 1\right) \tag{3.2}$$

After that, the logarithm of the energy from each filter is computed and the Discrete Cosine Transform is applied 3.3, and the cepstral vectors for each analysis window are obtained.

$$C_n = \sum_{k=1}^{K} S_k \cdot cos[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}] \qquad \text{n=1,2,...,L.} \tag{3.3}$$

3.2 Gaussian Mixture Models

Given a segment of speech (an observation O), and a hypothesized person P, the task of a verification system is to determine if O was from the speaker P [2]. This is restated as a hypothesis test (3.4). Using statistical pattern recognition, the optimum test to decide between the two hypothesis is a likelihood ratio test (LLR). The probability density functions of both hypothesis evaluated for the observation O are referred to the likelihood of the hypothesis, and the decision threshold for accepting or rejecting the hypothesis is given by θ.

$$\begin{array}{c} H_0 : O\, is\, from\, person\, P \\ H_1 : O\, is\, \underline{not}\, from\, person\, P \end{array} \qquad \frac{p\,(O \mid H_0)}{p\,(O \mid H_1)} \begin{cases} \geq \theta\ Accept\ H_0 \\ < \theta\ Reject\ H_1 \end{cases} \tag{3.4}$$

In a verification system the aim is to determine a technique to compute this likelihood ration function, usually by finding a method to represent and model the two likelihoods of the two hypothesis.

In some experiments of this thesis it is assumed that a Gaussian Mixture Model (GMM) distribution is representing the distribution of the feature vectors of the hypothesis. For a D-dimensional feature vector, the mixture density used for the likelihood function is defined as 3.5. The density is a weighted linear combination of M unimodal Gaussian densities p_i, each parametrized by a mean vector and a covariance matrix.Given a collection of training vectors, maximum likelihood model parameters are estimated using the iterative expectation-maximization (EM) algorithm, generally 5 to 10 iterations are sufficient for parameter convergence.

$$p(\bar{x}|\lambda) = \sum_{i=1}^{M} w_i p_i(\bar{x}) \tag{3.5}$$

The advantage of GMM as the likelihood function for text-independent speaker verification is that it is a well understood statistical model, and it offers a solid starting point for comparison of other methodologies.

3.2.1 Universal Background Model

A Universal Background Model (UBM) is a large GMM (usually 2048 mixtures) trained to represent the speaker-independent distribution of features. This is done to model alternative hypothesis in the likelihood ration test.

Specifically, speech that is reflective of the expected alternative speech encountered during recognition is selected [7].

The main advantage of this approach is that a single speaker-independent model can be trained and then used for all hypothesized speakers, but it is also possible to tailor specific sets of background models for specific tasks. It should be noted that the data has to be balanced if gender-independent UBM is wanted (same number of male speech and female speech samples), otherwise the UBM will be biased towards the dominant sub-population.

3.2.2 MAP Criterion

Even though the speaker-specific model can be procured with the same method explained before (GMM-EM), the approach to obtain the speaker model in this thesis is slightly different. The speaker model is derived by adapting the parameters of the background model UBM using the speaker's training speech and a form of Bayesian adaptation or maximum a posteriori (MAP) estimation.

The basic idea of this approach is to derive the speaker's model by updating the well-trained parameters in the background model via adaptation. This provides a better coupling between the speaker's model and the UBM.

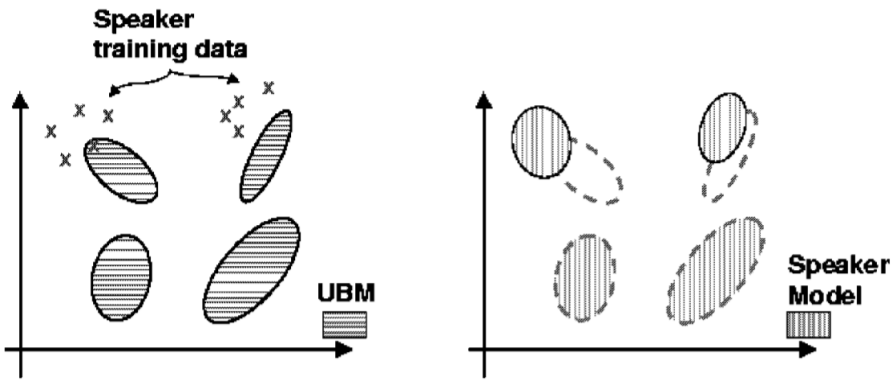

Figure 3.2: MAP adaptation. [2]

The first step of the adaptation model is the same as to the 'expectation' step in the Expectation Maximization, where the sufficient statistics of the speaker's training data area estimated and computed for each mixture in the UBM. In the second step, the new sufficient statistics are combined with the sufficient statistics from the background model mixture parameters, using a data-dependent mixing coefficient.

Given a UBM and training vectors from a speaker, the probabilistic alignment of the training vectors into the background model mixture components is determined with 3.6 for each mixture. Then, the sufficient statistics for the weight, mean and variance are computed: 3.7, 3.8 and 3.9.

$$Pr(i|\bar{x}_t) = \frac{w_i p_i(\bar{x}_t)}{\sum_{j=1}^{M} w_j p_j(\bar{x}_t)} \tag{3.6}$$

$$n_i = \sum_{t=1}^{T} Pr(i|\bar{x}_t) \tag{3.7}$$

$$E_i(\bar{x}) = \frac{1}{n_i} \sum_{t=1}^{T} Pr(i|\bar{x}_t)\bar{x}_t \tag{3.8}$$

$$E_i(\bar{x^2}) = \frac{1}{n_i} \sum_{t=1}^{T} Pr(i|\bar{x}_t)\bar{x}_t{}^2 \tag{3.9}$$

And lastly, the new sufficient statistics from the training data are used to update the old UBM sufficient statistics for each mixture to create the adapted parameters with 3.10, 3.11 and 3.12.

$$\hat{w}_i = [\alpha_i^w n_i/T + (1-\alpha_i^w)w_i]\,\gamma \tag{3.10}$$

$$\hat{\mu}_i = \alpha_i^m E_i(x) + (1-\alpha_i^m)\mu_i \tag{3.11}$$

$$\hat{\sigma}_i{}^2 = \alpha_i^v E_i(x^2) + (1-\alpha_i^v)(\alpha_i^2 + \mu_i^2) - \hat{\mu}_i{}^2 \tag{3.12}$$

where $\{\alpha_i^w \alpha_i^m \alpha_i^v\}$ are the adaptation coefficients that control the balance between the old and new estimates. The parameter updating can be derived from the general MAP estimation equations for a GMM using constrains on the prior distributions [2].

3.3 Extraction of i-vectors

A speaker utterance is represented by a supervector that consists of additive components from a speaker and a channel/session subspace [4]. Said speaker GMM supervector (M) can be defined as:

$$M = m + Vy + Ux + Dz \tag{3.13}$$

where there is a speaker- and a session-independent supervector m, generally from a universal background model (UBM) and V and D define a speaker subspace (eigenvoice matrix and diagonal residual, respectively), and U defines a session subspace (eigenchannel matrix).

The speaker- and session-dependent factors in the subspaces are the vectors $\{xyz\}$, each assumed to be a random variable with a normal distribution. To be able to apply JFA to speaker verification, a first step of estimating the subspaces from appropriately labelled development data and then a second estimation of the speaker and session factors (the vectors) for a given new target utterance. The speaker-dependent supervector is given by:

$$s = m + V_y + D_z \tag{3.14}$$

3.3.1 Total Variability Matrix and i-vectors

The Total Variability Matrix approach uses factor analysis as a feature extractor; and is based on defining only a single space, instead of two separate spaces, which are the classical JFA modeling (speaker space and the channel space). This single space, which is referred to the "Total Variability Space", contains the speaker and channel variabilities simultaneously. It is defined by the Total Variability Matrix that contains the eigenvectors with the largest eigenvalues of the total variability covariance matrix.

In this model, there is no distinction between the speaker effects and the channel effects in GMM supervector space. This new approach is motivated by the experiments that showed that the channel factors of the JFA which normally model only channel effects also contain information about the speaker.

Given an utterance, the new speaker- and channel-dependent GMM supervector is rewritten as 3.15.

$$M = m + Tw \tag{3.15}$$

In this case, the speaker- and channel-independent supervector (which could be the UBM supervector), is a low rank rectangular matrix and w is a random vector having a standard normal distribution. The components said vector are the total factors. These new vectors are referred to as identity vectors or i-vectors.

In this modeling, the speaker-dependent supervector is assumed to be normally distributed with mean vector and covariance matrix. As for training the Total Variability Matrix T, the process is exactly the same as learning the eigenvoice matrix V, except for one important difference: in eigenvoice training, all the recordings of a given speaker are considered to belong to the same speaker; in the case of the total variability matrix however, a speaker's set of utterances is regarded as having been produced by several speakers -i.e. it is pretended that every utterance from a given speaker is produced by different speakers-.

The model can be seen as a simple factor analysis that allows us to project a speech utterance onto the low-dimensional total variability space.

The total factor w is a hidden variable, which can be defined by its posterior distribution conditioned to the Baum–Welch statistics for a given utterance. This posterior distribution is a Gaussian distribution and the mean of this distribution corresponds exactly to the i-vector. The Baum–Welch statistics needed to estimate the i-vector for a given speech utterance are obtained by 3.16 and 3.17.

$$N_c = \sum_{t=1}^{L} P(c|y_t, \Omega) \tag{3.16}$$

$$F_c = \sum_{t=1}^{L} P(c|y_t, \Omega) y_t \tag{3.17}$$

where c corresponds to the Gaussian index and P to the posterior probability of mixture component generating the vector y. In order to estimate the i-vector, we also need to compute the centralized first-order Baum–Welch statistics based on the UBM mean mixture components 3.18. To fully ensure that the inverse of the equation matrix follows all the needed properties, a Cholesky Decomposition for matrix inversion is computed. [8]

$$\widetilde{F}_c = \sum_{t=1}^{L} P(c|y_t, \Omega)(y_t - m_c) \tag{3.18}$$

The i-vector for a given utterance can be obtained using 3.19.

$$w = (I + T^t \sum^{-1} N(u) T)^{-1} . T^t \sum^{-1} \widetilde{F}(u) \tag{3.19}$$

3.4 Scoring and Evaluation

As previously explained, the testing phase of a verification system requires a scoring method. After the system obtains such scores, a comparison between those and the threshold is made and the system decides to accept or reject the identity claim.

Some general scoring techniques of biometric systems offer a simple threshold that is applied to the entire system; but there are some approaches where a model-dependent threshold is estimated. Also, techniques such as Score Pruning –in which some non-representative scores are excluded- can be applied to the testing scores. For i-vector computation, the Cosine Distance is an accurate scoring method.

3.4.1 Speaker-dependent Threshold

As shown in Fig 2.2, there is a need for a threshold to determine if the speaker is accepted or rejected into the overall system.

A priori speaker-dependent (SD) threshold [9] [10] is estimated using only on data from the clients, employing standard deviation and client mean from LLR scores estimations. It uses only client scores from the enrolling phase due to the difficulties in selective impostor-feature samples, since in a real biometric system application, said impostors could become clients in the future.
The client mean estimation is adjusted by means of the client standard deviation estimation as 3.20.

$$\Theta = \mu - \alpha \cdot \sigma \tag{3.20}$$

where α is a constant empirically determined.

3.4.2 Fixed Score Pruning

A problem presented when there are only a few utterances available is that some of them could produce non-representative scores. This is common when an utterance contains background noises or is recorded with different handsets, or simply when the speaker is sick. The presence of outliers can induce to wrong estimations of mean and variance of client scores, leading to wrong speaker-dependent thresholds.

Score Pruning methods [6] for speaker-dependent (SD) threshold estimation calculate the maximum deviation and remove scores out of that interval.

Iterative Score Pruning methods iterate within an algorithm to find the maximum deviation allowed and remove the scores one by one out of an interval while changing the previously estimated mean in each iteration.

Non-iterative or Fixed Score Pruning methods for speaker-dependent threshold are those that employ the most typical scores discards a percentage of α most distant scores with respect to the mean, which is later re-estimated. That mean is used to estimate the speaker-dependent threshold.

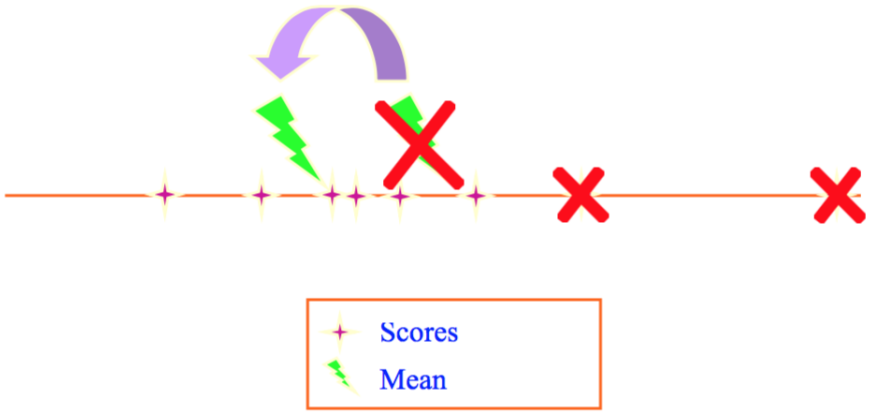

Figure 3.3: Fixed Score Pruning method. [6]

3.4.3 Cosine Distance

As explained in the previous chapter, the Cosine Distance is a scoring technique which uses the value of the cosine kernel between the target speaker i-vector and the test i-vector as a decision score:

$$score(w_{target}, w_{test}) = \frac{< w_{target}, w_{test} >}{\|w_{target}\| \, \|w_{test}\|} \simeq \theta \tag{3.21}$$

This value is then compared to the threshold θ in order to make the final decision. It should be noted that both target and test i-vectors are estimated exactly in the same manner –i.e. there is no extra process between estimating target and test i-vectors- so the i-vectors can be seen as new speaker recognition features.

3.4.4 Evaluation Measures

In order to evaluate the performance of the verification systems, some measures are defined [11]:

- False Acceptance Rate (FAR): measures the number of impostor attempts that have been granted access into the verification system with regard to the total number of impostors attempts. (3.22)

$$FAR = \frac{impostor\ scores\ exceeding\ threshold}{all\ impostor\ scores} \tag{3.22}$$

- False Rejection Rate (FRR): measures the number of clients attempts that have not been granted access into the system with regard to the total number of client attempts. (3.23)

$$FRR = \frac{genuine\ scores\ falling\ below\ threshold}{all\ genuine\ scores} \tag{3.23}$$

- Detection Error Trade-off (DET) Curve: represents FAR and FRR in both axes to fully represent the error trade-off of the system performance.

3.5 Software

3.5.1 Spro 4.0

SPro4 [12] is an open-source speech signal processing toolkit which provides run-time commands implementing standard feature extraction algorithms for speech applications and a C library to implement new algorithms.

This software has feature extraction techniques used in speech applications, such as: filter-bank energies, cepstral coefficients, linear prediction derived representation, etc. Even though the toolkit has been designed as a front-end for speech

applications, the library provides possibilities to implement other feature extraction algorithms. The library, written in ANSI C, provides functions for the following: waveform signal input, low-level signal processing, low-level feature processing and feature I/O.

Basically, the SPro4.0 toolkit [13] can read in an input audio file, process it and extract the feature vector. The most common SPro4.0 parameters that can be set are: Format (input format), Buffer size, Samples, Normalization and Derivatives.

Because the ALIZE / LIA_RAL software does not allow audio feature extraction, Spro4 has to be used to extract feature vectors. The Spro4.0 output feature vector will be the starting point for the LIA_RAL system.

3.5.2 ALIZE / LIA_RAL

ALIZE / LIA_RAL [14] is an open-source platform for biometric authentication with an LPGL license used specifically for speaker recognition. This software was developed by the LIA in the framework of the French Research Ministry Technolangue program at the University of Avignon REF:ALIZE/LIA RAL, France. It is divided into the ALIZE library and the high-level LIA_RAL toolkit; both of them are implemented in C++, and ALIZE uses GNU Autotools1 for platform independence.

LIA_RAL is the package containing all code specific to speaker recognition. The code in the LIA_RAL package can be divided into two categories: the generic library (referred to as Spk-Tools) which contains the training algorithms and some specialization of the statistical functions found in ALIZE, applied on feature vectors; and many small programs, designed to work in a pipeline methodology –i.e. there is one program for each step of a speaker verification system (3.4).

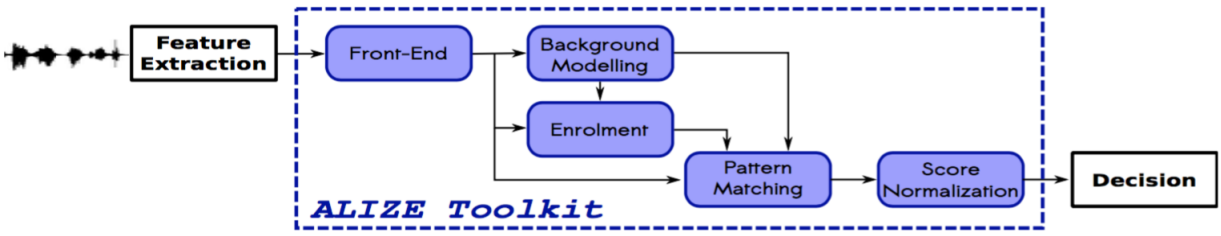

Figure 3.4: ALIZE toolkit system. [15]

On one hand, because the LIA_RAL toolkit is built on top of ALIZE, it is not possible to use only the LIA_RAL. On the other hand, since ALIZE in itself is too generic to be a speaker verification toolkit, the LIA_RAL is needed in order to evaluate the inner workings of ALIZE. The interesting speaker verifications algorithms are located in the LIA_RAL.

Furthermore, as the LIA_RAL package consists of small programs rather than a library for use in another C++ application, the user interaction with ALIZE is

small. In fact, nearly all user actions are wrapped in LIA_RAL structures, the only exception being the configuration file parser, which is found in ALIZE.

The overall software is developed following an object oriented UML method. Its general architecture is based on a split of the functionalities between several software servers. The main servers are the feature server, which manages acoustic data, the mixture server which deals with models (storage and modification) and the statistics server which implements all the statistical computations (such as Viterbi alignment and EM estimations).

Connecting all programs and servers of the pipeline requires reading the documentation, the example configuration files and the various unit-tests that exist for each program. Refer to Annex B to see the configuration files.

3.5.3 SpkIdAPI

The SpkIdAPI is a text-dependent speaker biometrics security software. It was developed by the Center for Language and Speech Technologies and Applications (TALP) [16] at the Universitat Politècnica of Catalunya (BarcelonaTECH), in Spain. It is a software designed for a text-constrained speaker verification system in which the known texts are number series.

The SpkIdAPI software is implemented in C++ and it uses libraries characteristics form C++98. Its structure is based on a library that can be combined into executable programs/binaries that result into the different steps for a speaker verification system. In contrast with the ALIZE/LIA_RAL software, SpkIdAPI has its own feature extractor to obtain the feature vectors.

In this thesis, one of the main goals is to create new methods in the SpkIdAPI software library so it implements text-independent speaker verification methods; more specifically, Total Variability Space methods (T-matrix and i-vector extraction).

Chapter 4

Experiments and Results

For the study of the speaker verification systems and methods, a total of three types of experiments have been carried out. Even though the methodology of each experiment may differ greatly, there are some points in common that ensure a coherent comparison between the methods.

4.1 Database

The database used for all experiments is the BioTech database, a multi-session database in Spanish especially designed for speaker recognition and owned by Biometric Technologies, S.L. It has a total of 184 speakers recorded by phone 106 male speakers and 78 female speakers. The BioTech database was recorded with 520 land-line calls from the Public Switched Telephone Network (PSTN) and 328 from cellphones.

The average number of sessions per speaker is 4.55 while the average time between each session is 11.48 days. In each session, there are several recordings used in the experiments: two utterances of a 4-digit number, six utterances of 8-digit numbers (three numbers repeated twice), and one-minute long utterance of spontaneous speech. The instructions given to the speakers while recording the sessions are included in Annex C.

For the experiments, a client/impostor classification of the BioTech database has been made. It has been established that clients are those speakers who have a total of 5 or more sessions, while impostors have between 2 or 4 sessions. This classification leaves the experiments with a database of 96 clients (57 male clients and 39 female clients).

4.2 Experimental Setup

The feature extraction of all experiments is the same: utterances are processed in 25 ms frames, Hamming windowed and pre-emphasized. The feature vector for each frame is formed by 12th order Mel-Frequency Cepstral Coefficients (MFCC), the normalize energy log, the Delta and the delta-delta parameters (acceleration).

All these features form a 39-dimensional vector for each frame, where Ceptral Mean Substraction (CMS) is also applied.

In all experiments, the male and female speaker results have been computed and compared separately.

4.3 Experiments and Results

4.3.1 Text-Dependent Speaker Verification

The speaker modeling is done with left-to-right HMM models with 2 states per phoneme and 1 mixture component per state for each digit and the silence model is a GMM with 128 Gaussians. It should be noted that world model (UBM) and the client model have the same topology. The UBM and silence models have been estimated from a subset of the respective database.

The speaker verification includes a speech recognizer for connected-digits recognition that during enrolment discards those utterances labeled as 'no voice'.

For the Text-Dependent Speaker Verification two experiments using number series have been carried out: a 4-digit utterances and an 8-digit utterances experiments. The speaker clients use 4 sessions for training – the number of training utterances can vary from 8 to 48- and the rest are for testing. This leaves with a total of 509 client tests and 66418 impostor tests, and 1025 clients and 133122 impostor tests for 8 digits.

Experiments with 4-digit utterances

To obtain a threshold that would accept or decline the identity claim, there has been two approaches: a general threshold for the entire system and a speaker-dependent threshold.

Methodology	**MALE**		**FEMALE**		**TOTAL**	
	FAR	**FRR**	**FAR**	**FRR**	**FAR**	**FRR**
General	0.11	0.03	0.11	0.09	0.11	0.05
SD Threshold	0.04	0.05	0.04	0.08	0.04	0.06

Table 4.1: FAR-FRR 4 digits

It is seen that a general threshold for the system does not offer as good results as a speaker-dependent threshold.

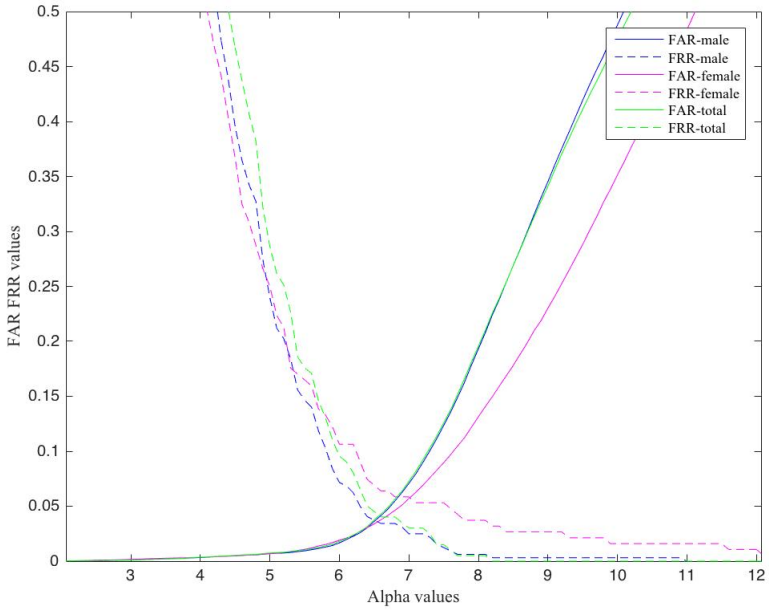

Figure 4.1: Text Dependent $\alpha = 6.5$ FAR-FRR with 4-digit utterances

As explained in Chapter 3, when applying the speaker-dependent (SD) threshold method for threshold estimation, it is required to empirically find the alpha parameter for equation 3.20.

Experiments with 8-digit utterances

The same methodology applied to the 4-digit utterances experiments is applied with the 8-digit utterances. A general threshold is computed for the entire system, and it is compared with the speaker-dependent threshold method.

Methodology	**MALE**		**FEMALE**		**TOTAL**	
	FAR	**FRR**	**FAR**	**FRR**	**FAR**	**FRR**
General	0.10	0.02	0.09	0.07	0.09	0.04
SD Threshold	0.04	0.01	0.05	0.05	0.04	0.02

Table 4.2: FAR-FRR 8 digits

The table shows the same conclusion: a speaker-dependent threshold has a better performance than a general threshold for all speakers in the system.

The same α parameter as before is tested to see if it is still optimal when applying the speaker-dependent (SD) threshold method for threshold estimation in 8-digit utterances.

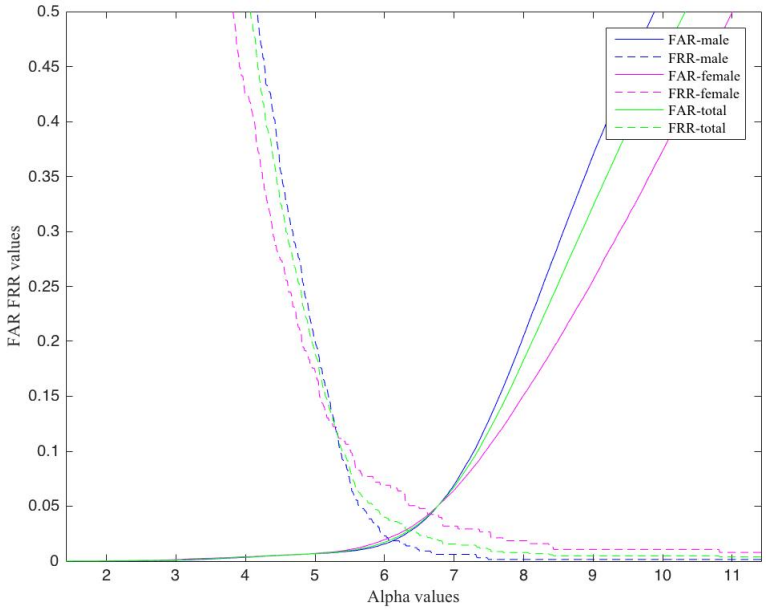

Figure 4.2: Text Dependent $\alpha = 6.5$ FAR-FRR with 8-digit utterances.

Text-Dependent Speaker Verification Results Summary

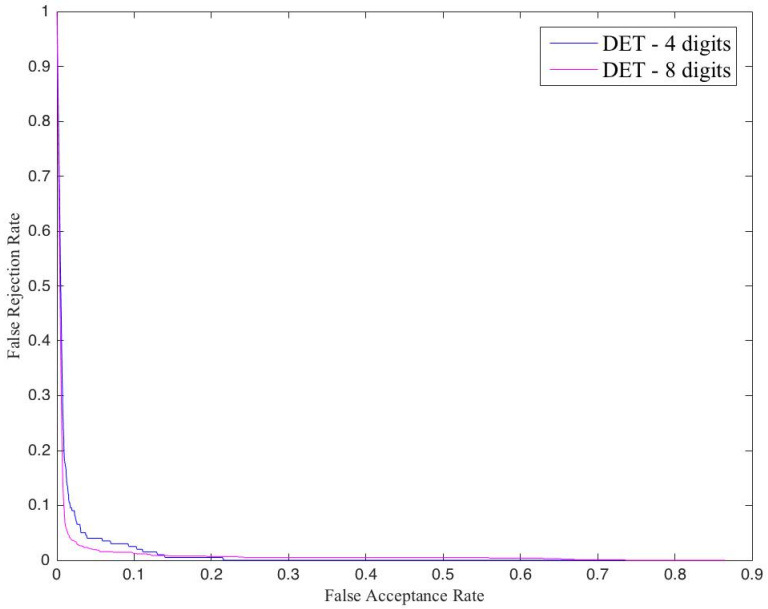

Figure 4.3: Text-Dependent DET Curve.

In the DET Curve above there is a comparison between both experiments. The experiments with 8-digit utterances have a lower FAR/FRR; and therefore offer a better performance. The main difference is in that using 8-digit utterances instead of 4-digit utterances, the system has been considerably more trained.

The overall performance of a text-dependent speaker verification system is very optimal.

4.3.2 Text-Independent Speaker Verification using GMMs

Text-independent verification experiments using GMMs have been carried out using unknown 1-minute non-text-constrained speech utterances. The main goals of this experiment were to prove that a speaker-dependent threshold is still better than a general threshold for the system, and to empirically find the alpha appropriate for the speaker verification system.

A total of 6 UBMs (3 all-male UBMs, 3 all-female UBMs) were created, 40 speakers each, but increasing the number of sessions and utterances for each UBM. It was proven that the results improved as the amount of data of the UBM increased.

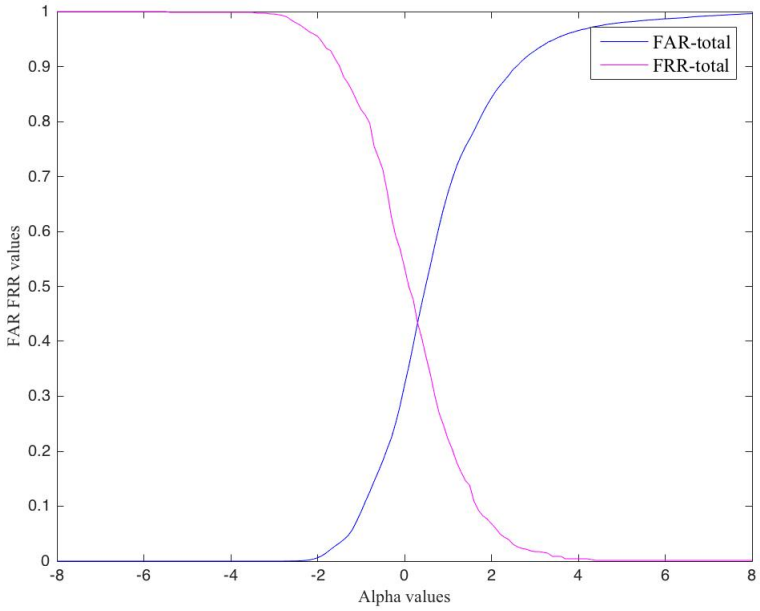

Figure 4.4: Text-Independent $\alpha = 0.3$ FAR-FRR

Since the results are far from optimal, a Fixed Score Pruning technique was applied, and the non-representative scores were removed.

Methodology	**MALE**		**FEMALE**		**TOTAL**	
	FAR	**FRR**	**FAR**	**FRR**	**FAR**	**FRR**
General	0.14	0.72	0.55	0.25	0.43	0.44
SD Threshold	0.38	0.36	0.37	0.36	0.38	0.37
Score Pruning	0.22	0.20	0.22	0.16	0.22	0.19

Table 4.3: Text-Independent methods using GMMs

If taken into comparison the text-dependent speaker verification system previously analyzed, the results are worse. It is proven once again that a speaker-dependent threshold is better than a general threshold for the entire system. It is also expected some worsen in the text-independent verification, since the system does not know a priori what the speaker is going to say.

4.3.3 Text-Independent Speaker Verification using i-vectors

Text-independent verification experiments using i-vectors have been carried out using two softwares: ALIZE/LIA_RAL and the integration of i-vector extraction into the SpkIdAPI library.

The i-vector dimension established is 400 and the number of GMMs used for the UBM is 256. It should be noted that most state-of-art experiments with i-vectors use at least 1028 GMMs for the world model. In this experiments, the number of GMMs was constrained by the technology, given that it takes about 5 full days to compute the UBMs and T-matrix with the aforementioned dimensions.

Much like in the text-independent speaker verification using GMM, a total of 6 UBMs (3 all-male UBMs, 3 all-female UBMs) were created, and consequently 6 T-matrix, to determine the optimal quantity of data to be trained.

Methodology	**MALE**		**FEMALE**		**TOTAL**	
	FAR	**FRR**	**FAR**	**FRR**	**FAR**	**FRR**
ALIZE/LIA_RAL i-vectors	0.13	0.13	0.12	0.12	0.13	0.13
SpkIdAPI i-vectors	0.14	0.14	0.13	0.13	0.14	0.14

Table 4.4: Text-Independent i-vectors method using different software.

Text-Independent Speaker Verification Results Summary

Methodology	**MALE**		**FEMALE**		**TOTAL**	
	FAR	**FRR**	**FAR**	**FRR**	**FAR**	**FRR**
GMMs	0.38	0.36	0.37	0.36	0.38	0.37
GMMs + SP	0.22	0.20	0.22	0.16	0.22	0.19
ALIZE/LIA_RAL i-vectors	0.13	0.13	0.12	0.12	0.13	0.13
SpkIdAPI i-vectors	0.14	0.14	0.13	0.13	0.14	0.14

Table 4.5: Text-independent Results Summary.

The i-vector approach offers unmistakeably better results in comparison with the Gaussian Mixture Model approach. The Score Pruning technique applied at the GMMs scores sets a considerably higher quality for the system, but its performance is still not as good as the i-vector method.

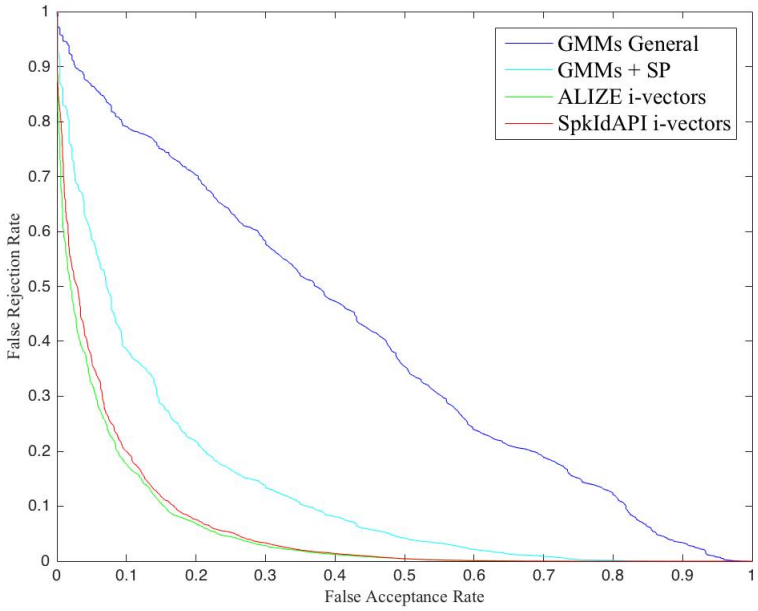

Figure 4.5: Text-independent Results Summary DET Curve.

In Fig. 4.5 it can be seen the different performance DET Curves of the text-independent systems.

It can be stated that the performance between the ALIZE software and the SpkIdAPI i-vector implementation is very similar, but ALIZE shows slightly superior results.

Chapter 5

Budget

5.1 Implementation Costs

In this chapter the project costs are detailed.

5.2 Software costs

The ALIZE / LIA RAL and the Spro4 software are open-source on a LPGL licence, therefore they do not represent an additional cost. The SpkIdAPI software and the BioTech database have been provided by the thesis tutor J. Hernando Pericas at no additional cost nor have intellectual property fees.

The results have been computed with MATLAB, which has a Academinc-Use licence cost of **500 €**

5.3 Development costs

In table 5.1 are separated the development costs in the different phases of the project.

Concept	Hour by ECTS Credit	Credits	Hours	Price/hour	Cost (€)
SpkIdAPI text-dependent	30	24	196	8 €	1568
SpkIdAPI adaptation for independent-speaker experiments (GMMs)			128		1024
ALIZE/LIA_RAL i-vectors study			124		992
Integration of i-vectos in SpkIdAPI			272		2176
Total			**720**		**5760**

Table 5.1: Development costs.

The approximately cost of the overall project is **6260 €**

Chapter 6

Conclusions and Future Development

6.1 Conclusions

This thesis premise was the study of the intricacies of speaker verification systems and the implementation of a text-independent speaker verification system. It centers mostly on the mathematics behind the speaker verification techniques and their implementation in a verification system, followed by a series of tests the systems' performance.

The study of the state-of-the-art techniques and the methodologies for text-constrained and text-independent speaker verification systems have been studied thoroughly. The structure of both systems has been analyzed, as well as the SpkIdAPI and ALIZE/LIA_RAL software.

Furthermore, different methodologies have been tested and compared in the experiments carried out in this thesis, given a broad overview of the different approaches for a text-independent verification system implementation. In addition, an actual implementation of i-vector extraction has been integrated into the SpkIdAPI library.

In this conclusion, one could firmly state that the goals of the thesis have been successfully achieved.

6.2 Future Development

As a future approach, an optimization of the SpkIdAPI library code and an integration of other text-independent methods could be useful to further the study of speaker verification systems and improve biometric security systems.

Another direction that could be followed after this thesis would be the study PLDA training and scoring for i-vectors and determine whether or not it could be implemented into the SpkIdAPI library, since they are also an actual state-of-the-art technique.

Bibliography

[1] Frédéric Bimbot, Jean-François Bonastre, Corinne Fredouille, Guillaume Gravier, and Douglas Reynolds. *A Tutorial on Text-Independent Speaker Verification*. EURASIP Journal on Applied Signal Processing. 430 - 451. 2004.

[2] Douglas A. Reynolds, Thomas F. Quatieri and Robert B. Dunn. *Speaker Verification Using Adapted Gaussian Mixture Models*. Digital Signal Processing 10. 19-41. 2000.

[3] P. Ouellet Patrick Kenny G. Boulianne and P. Dumouchel. *Joint Factor Analysis versus Eigenchannels in Speaker Recognition*. Signal Processing Toolkit, release 4.0. 2007.

[4] Howard Lei. *Joint Factor Analysis (JFA) and i-vector Tutorial*. ICSI.

[5] Niko Brümmer. *Bayesian PLDA*. 2010.

[6] Javier Rodríguez Saeta. *Decision Threshold Estimation and Model Quality Evaluation Techniques for Speaker Verification*. 2005.

[7] Douglas Reynolds. *Universal Background Models*. MIT Lincoln Laboratory, 244.

[8] L. Demanet. *The Cholesky decomposition*. MIT, 18.085 Spring 2014. 2014.

[9] Javier R. Saeta, Javier Hernando, Oscar Manso and Manel Medina. *Securing Certificate Revocation through Speaker Verification: the CertiVer Project*.

[10] Javier R. Saeta, Javier Hernando, Oscar Manso and Manel Medina. *Applying Speaker Verification to Certificate Revocation*.

[11] SYRIS Technology Corp. *Technical Document Abour FAR, FRR and EER*. 2004.

[12] *SPRO4.0*. URL: `https://www.irisa.fr/metiss/guig/spro/`.

[13] Guillaume Gravier. *SPro*. Signal Processing Toolkit, release 4.0. 2003.

[14] Laboratoire Informatique d'Avignon. *ALIZE*. URL: `http://mistral.univ-avignon.fr/`.

[15] Anthony Larcher, Jean-Francois Bonastre and Benoit Favue. *ALIZE 3.0 - Open Source Toolkit for State-of-the-Art Speaker Recognition*. University of Avignon - LIA, France.

[16] *TALP*. URL: `http://www.talp.upc.edu/`.

Appendix A

Work Plan Packages, Milestones and Gantt diagram

A.1 Work Packages

<table>
<tr><td>Project: Background Learning and Specifications</td><td colspan="2">WP ref: 1</td></tr>
<tr><td>Major constituent: Research</td><td colspan="2">Sheet 1 of 6</td></tr>
<tr><td rowspan="2">Short description: Research on the project topic (background knowledge) and the specifications to accomplish the goals.</td><td colspan="2">Planned start date: 15/02/2016
Planned end date: 01/03/2016</td></tr>
<tr><td colspan="2">Start event: 15/02/2016
End event: 01/03/2016</td></tr>
<tr><td>Internal task T1: Study of text-independent and text-dependent speaker verification theory.
Internal task T2: Study of GMM for text-independent speaker verification theory.
Internal task T3: Study of i-vectors methodology for speaker verification theory.</td><td>Deliverables:
D1.1 Project proposal and workplan</td><td>Dates:
01/03/2016</td></tr>
</table>

<table>
<tr><td>Project: Prototype</td><td colspan="2">WP ref: 2</td></tr>
<tr><td>Major constituent: Simulation/Integration</td><td colspan="2">Sheet 2 of 6</td></tr>
<tr><td rowspan="2">Short description: Design the prototype text-dependent verification system, integrate it with libraries and debug. An analysis of the obtained results must be made.</td><td colspan="2">Planned start date: 02/03/2016
Planned end date: 29/03/2016</td></tr>
<tr><td colspan="2">Start event: 02/03/2016
End event: 29/04/2016</td></tr>
<tr><td>Internal task T1: Design the prototype
Internal task T2: Integration with the libraries and development
Internal task T3: Text-dependent speaker verification results analysis</td><td>Deliverables:</td><td>Dates:</td></tr>
</table>

Project: Second Prototype	WP ref: 3	
Major constituent: Simulation/Integration	Sheet 3 of 6	
Short description: Design the prototype text-independent verification system, integrate it with ALIZE software. Perform an analysis of the obtained results.	Planned start date: 30/03/2016 Planned end date: 27/05/2016	
	Start event: 25/04/2016 End event: 12/06/2016	
Internal task T1: Design a second prototype. Internal task T2: Integration with the software and development. Internal task T3: Text-independent speaker verification results analysis	Deliverables: D3.1 Critical Review	Dates: 09/05/2016

Project: Integration	WP ref: 4	
Major constituent: Simulation/Integration	Sheet 4 of 6	
Short description: Integrate with the software libraries, develop and debug the code.	Planned start date: 30/05/2016 Planned end date: 13/06/2016	
	Start event: 13/06/2016 End event: 31/08/2016	
Internal task T1: design the implementation structure. Internal task T2: develop the code.	Deliverables:	Dates:

Project: Final Prototype	WP ref: 5	
Major constituent: Testing	Sheet 5 of 6	
Short description: test the final prototype.	Planned start date: 14/06/2016 Planned end date: 27/06/2016	
	Start event: 20/08/2016 End event: 05/09/2016	
Internal task T1: Tests	Deliverables: D5.1 Bachelor's Degree Thesis	Dates: 22/09/2016

Project: Documentation	WP ref: 6	
Major constituent: Documentation	Sheet 5 of 6	
Short description: document the specifications, planning, decisions made and results obtained in each step of the project.	Planned start date: 15/02/2016 Planned end date: 27/06/2016	
	Start event: 15/02/2016 End event: 22/09/2016	
Internal task T1: Document	Deliverables:	Dates:

A.2 Milestones

WP#	Task#	Short title	Milestone / deliverable	Date (week)
6	1	Background Research		3
6	2	Project Proposal and Work Plan	PPW	3
6	3	Critical Review	CR	13
6	1	Bachelor's Degree Thesis	BDT	19

A.3 Gantt diagram

Task name	Start date	Duratio...	Progr...	Assig...
Text-Independent Speaker Verification	15/02/16 (	7.25		
Text-Independent Speaker Verification Thesis	15/02/16 (	7.25	26%	
Research	15/02/16 (	0.50	100%	
Text-dependent systems	15/02/16 (	0.47	100%	Project
Text-independent systems	15/02/16 (	0.50	100%	Project
Projecto Proposal and Workplan	01/03/16 (			
Prototypes	02/03/16 (	2.25	63%	
First Prototype	02/03/16 (	1.45	100%	Project
First Prototype Tests	04/04/16 (	0.88	0%	Project
Final First Prototype Implementation	14/04/16 ´	0.52	65%	Project
Critical Review	09/05/16 (			
Second Prototype	29/04/16 ´	1.09	37%	Project
Second Prototype Tests	03/06/16 ´	0.34	0%	Project
Final Second Prototype Implementation	01/06/16 ´	0.41	0%	Project
Integration	13/06/16 (	2.62	0%	Project
Final Prototype	20/08/16 (	0.53	0%	Project
Final Prototype Tests	20/08/16 (	1.12	0%	Project
Bachelor's Thesis	22/09/16 (			

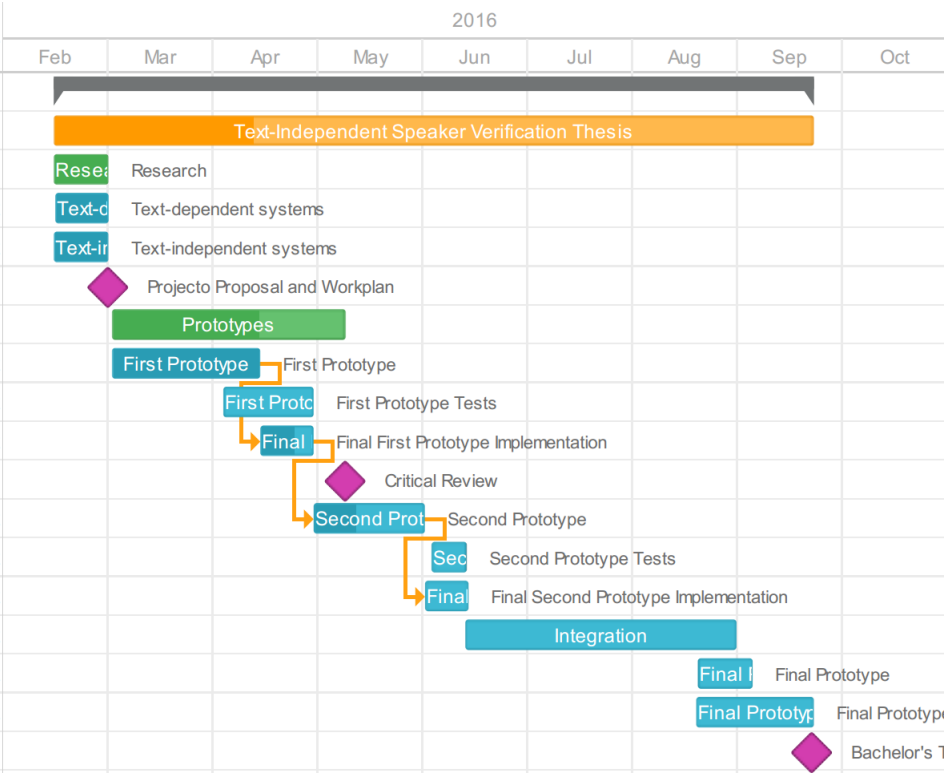

Figure A.1: Gantt

Appendix B

ALIZE Configuration Files

The main configuration files for the i-vector extraction using the ALIZE/LIA_RAL functions are detailed.

```
*** TrainWorld Configuration File
***

numThread                                   4
verboseLevel                                2

*******************************************************************************
*       In & Out
*******************************************************************************
loadFeatureFileExtension                    .norm.prm
loadFeatureFileFormat                       SPRO4
loadMixtureFileExtension                    .gmm
loadMixtureFileFormat                       XML
saveMixtureFileExtension                    .gmm
saveMixtureFileFormat                       XML
*******************************************************************************
*       Paths
*******************************************************************************
featureFilesPath                            data/prm/
labelFilesPath                              data/lbl/
mixtureFilesPath                            gmm/
*******************************************************************************
*       Feature options
*******************************************************************************
loadFeatureFileBigEndian                    false
featureServerMask                           0-11,13-38
*******************************************************************************
*       Computation
*******************************************************************************
maxLLK                                      200
minLLK                                      -200
*******************************************************************************
*       TrainWorld specific options
*******************************************************************************
baggedFrameProbability                      0.3
baggedFrameInitProbability                  0.7
normalizeModel                              true
baggedMinimalLength                         3
baggedMaximalLength                         10
inputFeatureFilename                        lst/UBM.lst
*       INITIALISATION
use01                                       false
nbFrameToSelect                             100
initVarianceFlooring                        0.01
initVarianceCeiling                         10
*       END
mixtureDistribCount                         256
nbTrainIt                                   8
finalVarianceFlooring                       0.01
finalVarianceCeiling                        10
outputWorldFilename                         world
```

Figure B.1: Train World configuration file, for UBM creation

```
*** TotalVariability Configuration file
***

********************************************************************************
*   In & Out
********************************************************************************
loadMixtureFileFormat                                 XML
loadMixtureFileExtension                              .gmm
loadFeatureFileFormat                                 SPRO4
loadFeatureFileExtension                              .norm.prm
loadMatrixFormat                                      DT
saveMatrixFormat                                      DT
loadMatrixFilesExtension                              .matx
saveMatrixFilesExtension                              .matx
********************************************************************************
*   Path
********************************************************************************
labelFilesPath                                        ./data/lbl/
featureFilesPath                                      ./data/prm/
mixtureFilesPath                                      ./gmm/
matrixFilesPath                                       ./mat/
********************************************************************************
*      Feature options
********************************************************************************
loadFeatureFileBigEndian                              false
featureServerMask                                     0-11,13-38
frameLength                                           0.01
********************************************************************************
*      TotalVariability specific options
********************************************************************************
nbIt                                                  10
orthonormalizeT                                       false
minDivergence                                         true
meanEstimate                                          newMeanMinDiv_it
loadAccs                                              false
nullOrderStatSpeaker                                  TV_N
firstOrderStatSpeaker                                 TV_F_X
randomInitLaw                                         normal
saveAllTVMatrices                                     true
saveInitTotalVariabilityMatrix                        true
loadInitTotalVariabilityMatrix                        false
initTotalVariabilityMatrix                            TV_init
totalVariabilityMatrix                                TV
totalVariabilityNumber                                400
checkLLK                                              false
computeLLK                                            1
inputWorldFilename                                    world
ndxFilename                                           ndx/totalvariability.ndx
```

Figure B.2: Total Variability Matrix configuration file, for T-Matrix creation

```
*** TrainTarget Configuration File
***

************************************************************
*   In & Out
************************************************************
saveMixtureFileFormat                          XML
loadMixtureFileFormat                          XML
loadMixtureFileExtension                       .gmm
saveMixtureFileExtension                       .gmm
loadFeatureFileFormat                          SPRO4
loadFeatureFileExtension                       .norm.prm
loadMatrixFormat                               DT
saveMatrixFormat                               DT
loadMatrixFilesExtension                       .matx
saveMatrixFilesExtension                       .matx
vectorFilesExtension                           .y
************************************************************
*      Feature options
************************************************************
loadFeatureFileBigEndian                       false
addDefaultLabel                                false
defaultLabel                                   speech
labelSelectedFrames                            speech
featureServerMask                              0-11,13-50
************************************************************
*       Path
************************************************************
mixtureFilesPath                               ./gmm/
matrixFilesPath                                ./mat/
saveVectorFilesPath                            ./iv/raw/
featureFilesPath                               ./data/prm/
labelFilesPath                                 ./data/lbl/
************************************************************
*      Computation
************************************************************
maxLLK                                                 200
minLLK                                                 -200
************************************************************
*      TrainTarget specific Options
************************************************************
minDivergence                                  true
meanEstimate                                   newMeanMinDiv_it
loadAccs                                       false
nullOrderStatSpeaker                           TV_target_N
firstOrderStatSpeaker                          TV_target_F_X
totalVariabilityMatrix                         TV
totalVariabilityNumber                         400
inputWorldFilename                             world
targetIdList                                   ndx/ivExtractor.ndx
```

Figure B.3: i-vectors extractor configuration file, for i-vector extraction

Appendix C

BioTech Database Speaker Session Form

The speaker session form for the BioTech Database is shown below:

Bienvenido al sistema de grabación de voz de Biometric Technologies. Para proceder a la grabación de los datos, recuerde que tiene que pronunciar los números dígito a dígito, sin pausas forzadas entre ellos. Si se equivoca, continúe igualmente. Y recuerde que ha de comenzar a hablar después de oír la señal. ¿Realiza su llamada desde un teléfono móvil o desde un fijo?

Diga su DNI dígito a dígito									
Diga su DNI dígito a dígito al revés									
Diga un número aleatorio de 4 cifras dígito a dígito		.....		.....		.....		.....	
Diga el número 1 dígito a dígito	Número 1	9		0		1		4	
Diga el número 2 dígito a dígito	Número 2	4	5	3	2	7	0	8	6
Diga el número 3 dígito a dígito	Número 3	3	7	1	5	9	2	6	8
Pronuncie las siguientes palabras :		*BODEGA*							
		PETACA							
		LLORAR							
		LECHUZA							
		JEFES							
		ROMÁNTICO							
A continuación, lea las siguientes frases:	Frase 1	-Los tiempos felices en la humanidad son las páginas vacías de la historia							
	Frase 2	-El genio es un rayo cuyo trueno se prolonga durante siglos.							
	Frase 3	-En la pelea se conoce al soldado y en la victoria al caballero							
	Frase 4	-Para obtener éxito en el mundo, hay que parecer loco y ser sabio.							
	Frase 5	-El miedo es para el espíritu tan saludable como el baño para el cuerpo.							
Lea el texto de su hoja de instrucciones.		A la desertización y la deforestación les sigue la contaminación química, que cada año provoca la muerte de millones de animales y plantas. Esta contaminación es causa del efecto invernadero: la temperatura media del planeta ha aumentado entre uno y dos grados en los últimos 100 años. Además, la enorme cantidad de residuos radiactivos o no biodegradables han convertido grandes extensiones en vertederos incompatibles con la vida. Todo ello destruye los ecosistemas. Se trata de una de las causas principales, junto al crecimiento demográfico y a la caza furtiva, de que en poco más de 20 años se hayan extinguido 500 especies animales. Las pérdidas, a las que muy pronto se podrían sumar el buitre negro, el lince ibérico, el águila pescadora y un tipo de esturión, no se detienen. En los próximos 30 años pueden desaparecer de la faz de la Tierra una cuarta parte de las especies animales y vegetales, a un ritmo de 100 diarias.							
Hable durante un minuto (aprox.) sobre el tema que usted desee. Por ejemplo sobre lo que ve a su alrededor, qué ha hecho el fin de semana, el último libro que ha leído o la última película que ha visto, etc.		....							
Diga su DNI dígito a dígito									
Diga su DNI dígito a dígito al revés									
Diga otro número aleatorio de 4 cifras dígito a dígito		.....		.....		.....		.....	
Diga el número 1 dígito a dígito	Número 1	9		0		1		4	
Diga el número 2 dígito a dígito	Número 2	4	5	3	2	7	9	8	6
Diga el número 3 dígito a dígito	Número 3	3	7	1	5	9	2	6	8

Su sesión ha concluido. Muchas gracias por su colaboración.

Figure C.1: BioTech Database Speaker Session

Glossary

A
ANSI: American National Standards Institute
C
CMS: Cepstral Mean Substraction
D
DET: Detection Error Trade-off
E
EM: Expectation Maximization
F
FAR: False Acceptance Rate
FFT: Fast Fourier Transform
FRR: False Rejection Rate
G
GMM: Gaussian Mixture Model
H
HMM: Hidden Markov Model
J
JFA: Joint Factor Analysis
L
LLR: Likelihood Ratio
LPGL: Lesser General Public Licence
M
MAP: Maximum A Posteriori
MFCCs: Mel-Frequency Cepstral Coefficients
MLE: Maximum Likelihood Estimation
P
PIN: Personal Identification Number
PLDA: Probabilistic Linear Discriminant Analysis
PSTN: Public Switched Telephone Network
S
SD: Speaker Dependent
SP: Score Pruning
SVM: Support Vector Machines
T
TALP: Language and Speech Technologies and Applications
U
UBM: Universal Background Model
UML: Unified Modeling Language
UPC: Universitat Politècnica de Catalunya