UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

SPEAKER TRACKING SYSTEM USING SPEAKER BOUNDARY DETECTION

Master's Thesis

Submitted to the Faculty of the

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Umair Khan

In partial fulfillment

of the requirements for the degree of

MASTER IN TELECOMMUNICATIONS ENGINEERING

Advisor: Professor Francisco Javier Hernando Pericás

Barcelona, November 2016

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

telecom BCN

Title: **Speaker Tracking System Using Speaker Boundary Detection.**

Author: **Umair Khan**

Advisor: **Professor Francisco Javier Hernando Pericás**

Abstract:

This thesis is about a research conducted in the area of Speaker Recognition. The application is concerned to the automatic detection and tracking of target speakers in meetings, conferences, telephone conversations and in radio and television broadcasts. A Speaker Tracking system is developed here, in collaboration with the Center for Language and Speech Technologies and Applications (TALP) in UPC. The main objective of this Speaker Tracking system is to answer the question: When the target speaker speaks? The system uses training speech data for the target speaker in the pre-enrollment stage. Three main modules have been designed for this Speaker Tracking system. In the first module an energy based Speech Activity Detection is applied to select the speech parts of the audio. In the second module the audio is segmented according to the speaker turning points. In the last module a Speaker Verification is implemented in which the target speakers are verified and tracked. Two different approaches are applied in this last module. In the first approach for Speaker Verification, the target speakers and the segments are modeled using the state-of-the-art, Gaussian Mixture Models (GMM). In the second approach for Speaker Verification, the identity vectors (i-vectors) representation is applied for the target speakers and the segments. Finally, the performance of both these approaches is compared for the results evaluation.

Dedicated to my Parents and my Sister.

Acknowledgments

First of all I would like to thank Almighty ALLAH, who has always blessed me. Because of His blessings I got the confidence to manage this thesis and all other hardships in my life. I think this thesis wouldn't be possible without the supervision and ideas of my advisor, Francisco Javier Hernando Pericás. I would like to thank him for his support and cooperation which lead me to this point. Working with him in TALP research center, has given me a lot. It helped me in developing skills to work in a competitive environment and enhancing my interests in the area of Speaker Recognition.

I would like to thank the staff of TSC department for their support and cooperation. In TALP research center, I would like to thank Miquel Angel India Massana who helped me whenever I was stuck in the programming part of this thesis. Also I would like to appreciate the advises from Abraham Woubie, who helped me with many issues related to the Speaker Recognition terminologies. I am thankful to Omid Ghahabi for helping me in the Speech Activity Detectors and ALIZE-3.0 toolkit. Carla Cortillas has also helped me in some configuration parameters for this toolkit, for which I am very thankful to her.

Finally, I am extremely thankful to my parents and my sister who have always supported me and always prayed for my success. I am grateful to all my friends who are always there whenever I feel down in life.

Revision History and Approval Record

Revision	Date	Purpose
0		
1		
2		
3		

Written by		Reviewed and Approved by	
Date	November 22, 2016	Date	November 22, 2016
Name	Umair Khan	Name	Francisco Javier Hernando Pericás
Position	Thesis Author	Position	Thesis Supervisor

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter is a brief introduction of the thesis. It describes the context and motivation of the author that inspired him to do this thesis. The main objectives to be achieved in this thesis are listed here. This thesis is done in Universitat Politècnica de Catalunya (UPC) Center for Language and Speech Technologies and Applications (TALP). It is a great opportunity to work in collaboration with this research group. Speaker Recognition is one the main working areas in this group. As, the main task of this thesis is Speaker Tracking, the idea is to perform a two step approach i.e: Speaker Segmentation and then Speaker Verification. Speaker Tracking is somehow related to Speaker Diarization. In Speaker Tracking tasks the audio recording is passed through segmentation step and then the verification. While in Speaker Diarization the verification step is replaced by clustering.

1.1 Context and Motivations

Speaker Recognition is one of key the applications of speech processing that can be used as a modern biometric system. Speaker Recognition includes both Speaker Identification and Speaker Verification. Humans possess unique acoustic characteristics which are useful features for the recognition process. In Speaker Recognition task a person speaking is identified/verified by using his/her voice characteristics. Nowadays, a big amount of data is communicated through television and internet in meetings and conferences where multiple speakers are speaking. This opens a new era for the Speaker Recognition systems. Speaker

Recognition is a vast area of research depending upon the application scenario. For example: Speaker Identification, Speaker Verification, Speaker Segmentation, Speaker Clustering, Speaker Diarization and Speaker Tracking etc. In some applications, a specific/target speaker or only *a person of interest* is to be identified by using his voice. This is basically a Speaker Tracking task. To identify, *when the target speaker speaks,* in the conference or meeting, is termed as Speaker Tracking.

This thesis mainly focuses on Speaker Tracking by using a simple technique of speaker turn points detection for Speaker Segmentation. The first step is Speaker Segmentation, in which the points in time are detected where there is a speaker change in the audio. In the next step the speaker turn points are re-confirmed and finalized. Once these are confirmed, the audio signal is segmented according to the finalized speaker turn points. This, literally, means that every segment belongs to different speakers appearing in the audio. In the second step, i.e: Speaker Verification, the different segments from the previous step are tested against the target speakers, which are pre-enrolled in the database. Thus the target speaker/speakers is/are tracked in the whole audio signal in the test. Nowadays, most of the speaker identification/verification systems use Mel-Frequency Cepstral Coefficients (MFCC) as features and Gaussian Mixture Models (GMM) as modeling technique as s state-of-the-art technique. In this thesis, the MFCC features are used both for speaker turn points detection and confirmation. For speaker modeling, GMM modeling using the Expectation Maximization (EM) algorithm is used and the Identity Vectors representation is used for performance improvement.

1.2 Objectives

As this thesis give emphasis on Speaker Tracking, the main objectives are divided into two main categories. The first category is Speaker Segmentation in which the audio is segmented according to different speakers. The second category is Speaker Verification where the target speaker/speakers are tested against the speaker appearing in different segments from the segmentation step. Keeping these two main points in mind, the following points are set as the main objectives of this thesis:

- **Speaker Boundary Detection:** The Speaker Segmentation step of this system relies on the speaker boundary. Thus the first goal is to detect those points, where the speaker changes, with the help of a simple Divergence Distance measure.

- **Speaker Segmentation:** The next objective is to segment the audio with respect to the speaker boundaries detected. Before this, the speaker boundaries has to be confirmed with a confirmation algorithm using a Universal Background Model.

- **Speaker Verification:** In order to perform the Speaker Tracking task, which is the ultimate goal, a Speaker Verification has to be implemented. The target speakers has to be pre-enrolled and then tested against the segments from the last step. For this purpose, again an advantage of the Universal Background Model has to be taken into account. Two different approaches are implemented in this last step:

  1. **GMM Modeling:** GMM models of both the target speakers and all the test segments has to be developed for Speaker Verification.
  2. **Identity Vectors Representation:** Identity vectors or simply **i-vector** models both for target speakers and all the test segments has to be developed. A performance comparison has to be done in both these cases.

1.3 Thesis Outline

This thesis consists of five chapters in the area of Speaker Segmentation and Speaker Verification. Following is the order and description of each chapter:

1. **Chapter 2** describes the state of the art in Speaker Recognition and specifically in Speaker Tracking tasks. Different Speaker Modeling approaches are briefly discussed here. Two of the main application tasks in Speaker recognition, i.e: Speaker Diarization and Speaker Tracking are explained in detail. Various steps in these two tasks are discussed.

2. **Chapter 3** explains all the details of the proposed Speaker Tracking system. This chapter tells how the proposed Speaker Tracking system is developed. It contains the

detailed formulation and theoretical description of every step of this thesis. The divergence shape distance computation for Speaker Segmentation and Speaker Tracking is explained. Then the speaker modeling techniques i.e: GMM and i-vectors, with illustration examples, and UBM training are explained in this chapter.

3. **Chapter 4** contains details about the experimental setup, database and the experiments carried out for Speaker Segmentation and Speaker Verification, in the evidence of this thesis. The results of these experiments are analyzed and discussed in the lights of different modeling techniques and their performances as described above. The evaluation metrics for Speaker Tracking tasks are also discussed in this chapter.

4. **Chapter 5** concludes the whole thesis with respect to system's performance and results, giving future recommendations in this task.

Chapter 2

State of the Art

This chapter explains the state of the art in the area of Speaker Recognition systems. It defines the different categories of a Speaker Recognition system which are Speaker Identification and Speaker Verification. This chapter contains a detailed explanation of different modeling approaches for Speaker Identification and Speaker Verification, for example Gaussian Mixture Models, Hidden Markov Models, Artificial Neural Networks, Deep Neural Networks, Support Vector Machines, Supervectors and Identity Vectors. Applications of Speaker Recognition system like Speaker Diarization system and Speaker Tracking system are also discussed in this chapter. Different stages in Speaker Diarization and Tracking systems are explained in detail.

2.1 Speaker Recognition

Speaker Recognition is one of key the applications of speech processing that can be used as a modern biometric system. Humans possess unique acoustic characteristics which can be extracted as useful features for the identification process. In Speaker Recognition task a person speaking is identified by using his/her voice characteristics. Speaker Recognition is further categorized into Speaker Identification and Speaker Verification. In a Speaker Identification system a person who is speaking is to be identified. Simply it answers *"Who is this person?"* In general the speaker, in this case, is from a known set of speakers and the system has to find the speaking person out of them [13]. On the other hand, in Speaker

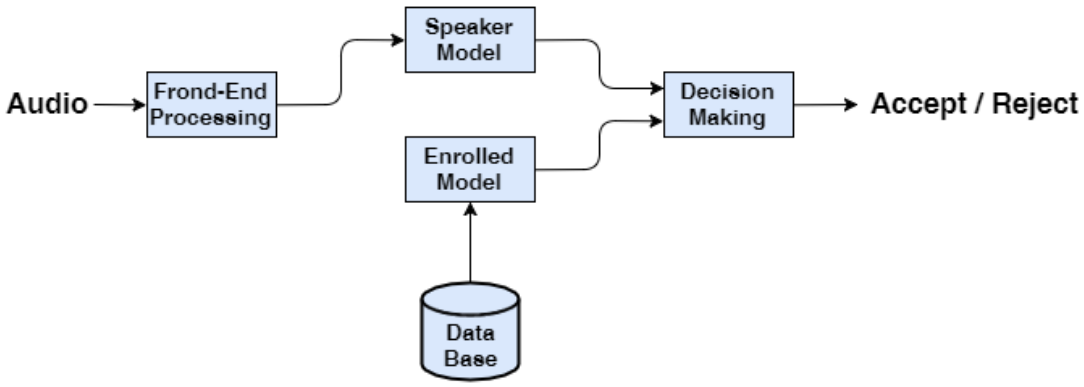


Figure 2.1 – *A Simple Block Diagram of Speaker Recognition System*

Verification a person who claims an identity is verified whether he is correct or not. This system answers, for example, *"Is this the voice of Nimra?"* Thus the only difference between Speaker Identification and Speaker Verification is the number of test experiments. In Speaker Identification, the number of tests experiments is the same as the available events. In Speaker Verification the task is rather a simple one, either the claimed identity is accepted with proof or rejected in case of not having sufficient score. Figure 2.1 shows a simple block diagram of a Speaker Recognition system. The speech is passed through a Front-end Processing module where the speech parts are separated from the non-speech parts and the acoustic features are extracted. The features are, then, modeled by using any modeling technique, to develop a speaker model. The model is matched with a verifying model from the database. These models are supposed to be pre-enrolled and stored in the database. A decision is taken in the scoring part and the speaker model is either accepted or rejected depending on the threshold parameter. The threshold setting depends on the application sensitivity.

In General, a Speaker Recognition system has two phases, i.e: The Training Phase and The Testing Phase. In the training phase, the speakers models are developed with enough amount of training speech data. Features are extracted from the samples and a specific speaker model is trained. The model is labeled with the speaker identity and is stored in the database. Similarly a large amount of data base is developed with the speakers models and their respective identities. In the testing phase, an unknown speaker is tested against the models stored in the data base. This phase depends on the problem, whether its a Speaker

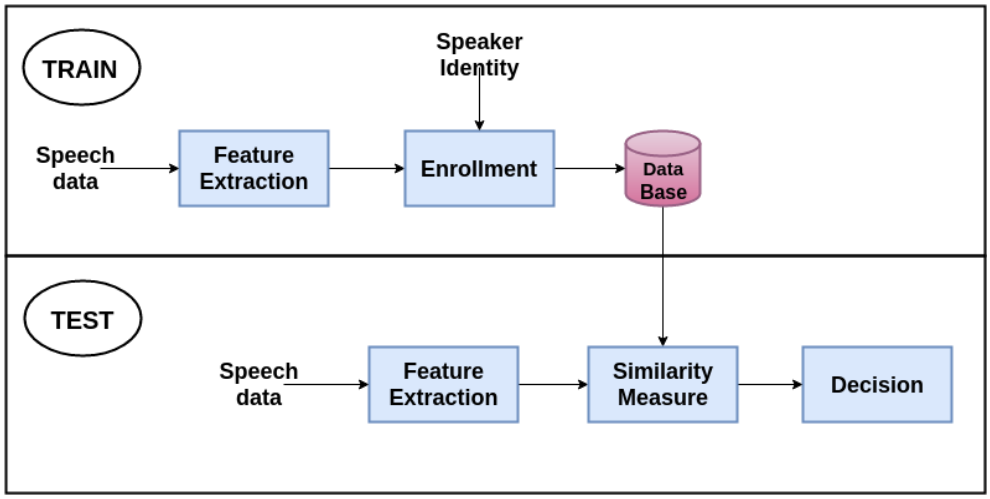


Figure 2.2 – *Block Diagram of the Two Phases of a Speaker Recognition System*

Identification case or Speaker Verification case? In Speaker Identification case, the Features are extracted from the speech samples of the testing speaker. A speaker model is developed and then it is identified against all the speakers models in the data base. This is a long process, as it depends on the number of the speaker models stored in the data base. At the end, the testing speaker is assigned to one of the speaker identities from the data base. It is also possible to label the testing speaker as unknown or new speaker if it does not matches any of the labels in the data base. On the other hand, in a Speaker Verification case, the testing speaker claims an identity. In this case, it is easy for the system to match the testing speaker only with the claimed identity model from the data base, and not with all the data base. The matching score is compared with a threshold depending upon the application sensitivity and the testing speaker is either accepted or rejected.

In either phase, Training or Testing, and in either case, Speaker Identification or Speaker Verification, the importance of speaker modeling can not be underestimated. In the next section, the different modeling approaches are discussed in detail.

2.2 Speaker Models

The segments or clusters are to be modeled in an efficient way so as to be used for Speaker Recognition tasks. However, modeling technique is really important in this task. Various

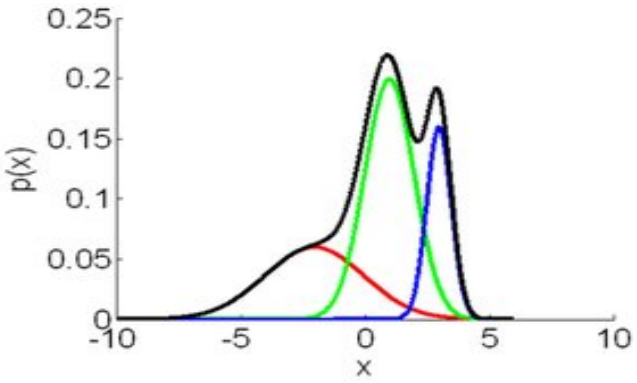


Figure 2.3 – *Gaussian Mixture Models (Expectation Maximization of Gausian Mixture Models in VTK 2010)*

approaches are applied, so for. Most of these approaches assume some data structure to some extent for example its statistics or the probability density function. Some of them are:

- **Gaussian Mixture Models (GMM):** GMM is considered to be the most common and state-of-the-art approach for speaker modeling. This approach is based on a weighted sum of Gaussian component densities as the parametric probability function of a model. A GMM is a parametric probability density function represented as a weighted sum of Gaussian component densities. This is graphically shown in Figure 2.3.

$$P(x/\mu_i, \Sigma_i) = \sum_{n=i}^{k} w_i g(x/\mu_i, \Sigma_i) \tag{2.1}$$

Where x is a N-Dimensional feature vector, w_i is the weight of ith Gaussian component and $g(x/\mu_i, \Sigma_i)$ is the ith Gaussian component density. $g(x/\mu_i, \Sigma_i)$ is given by:

$$g(x/\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)) \tag{2.2}$$

Where μ_i and Σ_i are the mean vector and covariance matrix of the ith Gaussian component respectively. The mixture weights w_i must satisfy the constraint: $\sum_{n=i}^{k} w_i = 1$. The GMM components are represented by the three parameters as:

$$\lambda_i = \{w_i, \mu_i, \Sigma_i\};\ i = 1, ..., k \tag{2.3}$$

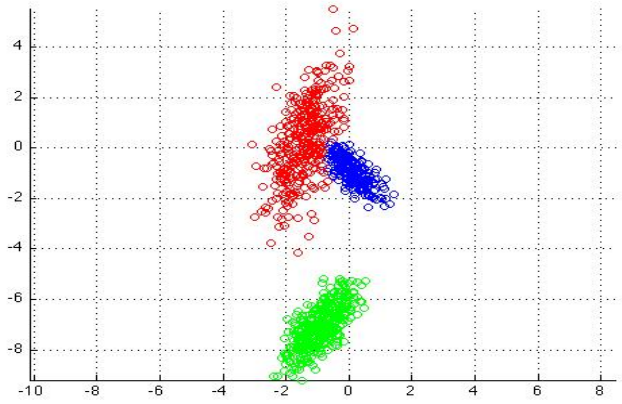


Figure 2.4 – *Gaussian Mixture Models using Expectation Maximization (EM Algorithm for Gaussian Mixture Model, MathWorks, 2016)*

GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features in a biometric system, such as vocal-tract related spectral features in a speaker recognition system. Maximum likelihood (ML) parameter estimations are obtained by using a few iterations of EM algorithm. In this case, each model is built independently by using the training utterances provided by the registering speaker. Figure 2.4 shows an illustration of the clustering by using EM. Each cluster is a GMM component having a particular weight [10].

A speaker may be modeled by using either a decoupled GMM from training data or by means a Maximum A Posteriori (MAP) estimation, a form of Bayesian adaptation. In this case, also termed GMM-adaptation, each model is the result of adapting a general model, which represents a large population of speakers, to better represent the characteristics of the specific speaker being modeled. This general model is usually referred to as world model or universal background model (UBM). An UBM is a large GMM model used in a biometric verification systems to represent general, person independent feature characteristics to be compared against a model of person-specific feature characteristics when making the accept or reject decision [1].

- **Hidden Markov Models (HMM):** HMM is a stochastic model and is a type of Bayesian network, normally used for modeling applications where the observations are

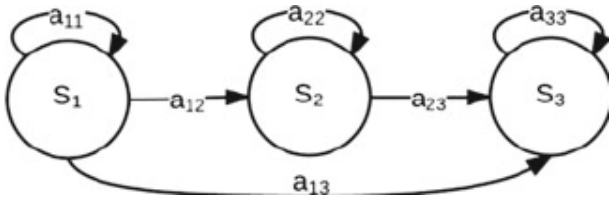


Figure 2.5 – *A 3-State Hidden Markov Model [28]*

a probabilistic function of the state. Basically HMM acts as a finite-state machine and has each state associated with an event that can be observed deterministically and the observations (features) are stochastic function of the state [1]. A probability density function or feature vector stochastic model is associated with each state of the HMM. The probability that a sequence of speech frames was generated by this model is found by using Baum–Welch decoding [8]. This likelihood is the score for L frames of input speech given the model.

$$p(x(1;L)/model) = \sum_{All-States} \prod_{i=1}^{L} p(x_i/s_i)p(s_i/s_{i-1}) \tag{2.4}$$

Where $p(x_i/s_i)$ is the probability distribution function associated with state s_i. The states are connected by a transition network, where the state transition probabilities are $a_{ij} = p(s_i/s_j)$. A classification is performed with the help of this scores. A hypothetical three-state HMM is illustrated in Figure 2.5. a_{11}, a_{12}, a_{13}, a_{22}, a_{23} and a_{33} are the transition probabilities between states S_1, S_2, and S_3.

- **Artificial Neural Networks (ANN):** Artificial neural networks are also used in speaker recognition applications. The kind of neural networks used are feed-forward neural networks, where the information moves only in forward direction from the input nodes, through the hidden nodes, if any, and to the output nodes. Commonly, a feed-forward neural network is created for each known speaker, and each network contains one output that is trained to be active only for its speaker. In the testing phase, an input feature vector is fed forward through each network, and the identification is determined by the network with the highest accumulated output values. In the speaker

verification mode, the input vectors of the unknown user are fed forward through the network belonging to the claimed speaker. If the average output value is bigger than a threshold, the speaker is accepted [1].

- **Deep Neural Networks (DNN):** Deep learning methods are machine learning methods using multiple processing layers or levels of abstraction. A successful application of deep learning technologies consists in selecting a good architecture for the neural network as well as an effective training procedure to learn the parameters of the network. In recent years, modeling using neural networks has emerged very strongly.

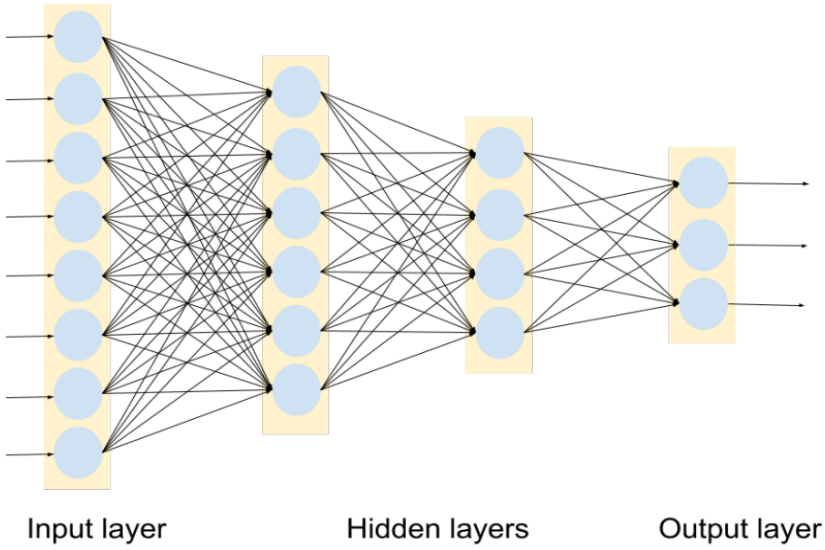


Figure 2.6 – *Architecture of Deep Neural Network [29]*

Other important factors of this renaissance are the availability of higher computing power and large databases which is necessary to train multilayer structures with a large number of parameters. Although its widespread use started a few years ago, and despite the difficulty of analyzing the behavior of deep learning algorithms, the impact of deep learning is already very important in areas as image, speech and text processing in research and commercial applications. In speech recognition we have now systems based on a simple generic deep learning architecture that outperform traditional speech recognition systems with many speech-specific processing modules. An architecture of

a DNN system is depicted in Figure 2.6. It has an input layer, three hidden layers and an output layer. The layers are connected with the help of connectors.

- **Support Vectors Machine (SVM):** This approach relies on stacking a huge number of speech features in a vector (supervector) which is finally modeled by a Support Vector Machine. This strategy is known to be a high performance speaker recognition approach. The SVM model relies on two assumptions. First, transforming data into a high-dimensional space may convert complex classification problems (with complex decision surfaces) into simpler problems that can use linear discriminant functions. Second, SVMs are based on using only those training patterns that are near the decision surface assuming they provide the most useful information for classification. A common

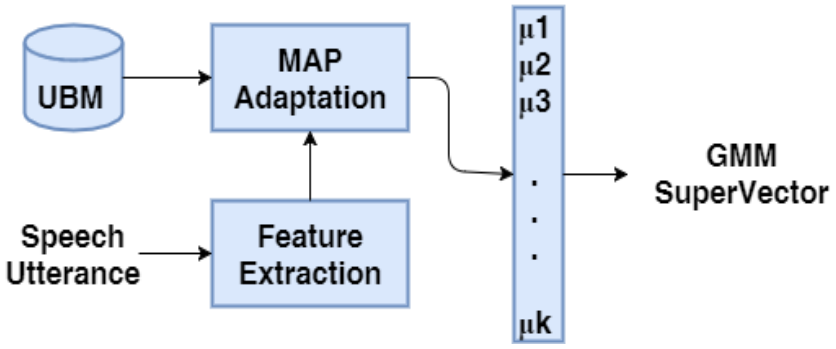


Figure 2.7 – *Architecture of Gaussian Supervector Modeling*

way to combine SVM with GMM is the so-called Gaussian Super-Vector (GSV). Figure 2.7 depicts the architecture of this system. Normally, a MAP adapted Gaussian Model of the speaker is developed and the components means are stacked in a high dimensional vector and is fed to the SVM. This is called a Supervector. A Joint Factor Analysis (JFA) says that a supervector, for a speaker, must be split into speaker dependent, speaker independent, channel dependent and residual parts. Each of these components can be represented by a low-dimensional set of factors. Thus, for a given speaker, a GMM supervector can be split into these components as follows:

$$s = m + Vy + Ux + Dz \tag{2.5}$$

Where vector m is a speaker independent supervector which obtained from the UBM,

V is the eigenvoice matrix, y is the speaker factors vector, U is the eigenchannel matrix, x is the channel factors vector, D is the residual matrix which is a diagonal matrix and z is the speaker specific residual factors vector. For a GMM-UBM system of complexity equal to 512 mixtures, the practical and recommended dimensions of each Joint Factor Analysis components, discussed above, are as follows:

- V is a 20,000 by 300 matrix
- y is a 300 by 1 vector
- U is a 20,000 by 100 matrix
- x is a 100 by 1 vector
- D is a 20,000 by 20,000 matrix
- z is a 20,000 by 1 vector

- **Identity Vectors (i-vectors):** Super-vectors can further be transformed to lower dimensional vectors called Identity Vectors or i-vector [21]. I-vectors are actually a compact representation of speech signals and have been the-state-of-the-art over the last few years. Suppose a supervector is decomposed as:

$$s = m + Tw \tag{2.6}$$

Where s is the source side supervector of a speaker, m is a speaker independent supervector which is obtained from UBM, T is a low rank Total Variability Matrix and w is the i-vector. Given s, m and a trained T matrix, an i-vector w can be easily extracted. An i-vector system uses a set of low-dimensional Total Variability Factors (w) to represent each speaker. Each factor controls an eigen-dimension of the low ranked Total Variability Matrix (T), and are known as the i-vectors. An i-vector is extracted for each speaker and then a cosine distance score is computed for matching. The cosine score between i-vector w_i and i-vector w_j is given by:

$$CosineScore(w_i, w_j) = \frac{w_i^* * w_j}{||w_i^*|| * ||w_j||} = \cos(\theta_{w_i,w_j}) \tag{2.7}$$

If the i-vectors of two speakers point in the same direction, the cosine distance score takes highest value up to 1. If they point in opposite directions, the cosine distance score takes lowest value of up to -1.

Research Activities

The state-of-the-art systems usually utilize statistical modeling algorithms in their training phases as statistics better characterize the speaker-specific information. The fast growing of the improvements in the modeling stage during the past few years shows the importance and the high attention of the research groups to this stage. The most widely applied approach to speaker representation is based on Gaussian mixture models. Maximum likelihood model parameters are estimated by the iterative Expectation Maximization algorithm. In GMM based speaker recognition, a Universal Background Model is first trained with the EM algorithm from long duration of speech data gathered from a large number of speakers [4]. The background model represents speaker-independent distribution of feature vectors. When enrolling a new speaker to the system, the parameters of the background model are adapted to the feature distribution of the new speaker. The adapted model is then used as the model of that speaker. In this way, the model parameters are not estimated from scratch, with prior knowledge being utilized instead. In the recognition mode, the MAP-adapted model and the UBM are coupled, and the recognizer is commonly referred to as GMM-UBM. The match score depends on both the target model and background model through average log likelihood ratio. How to represent utterances having a varying number of feature vectors using a single vector, a so-called super-vector, is an issue. Since the UBM is included as a part in most speaker recognition systems, it provides a natural way to create super-vectors [20]. This lead to a hybrid classifier where the generative GMM-UBM model is used for creating feature vectors for the discriminative Support Vector Machine (SVM). Super-vectors can further be transformed to lower dimensional identity vectors referred to as i-vector [21]. i-Vectors are actually a compact representation of speech signals and is capturing the place of state-of-the-art in last years. Also the success of deep learning in speech recognition inspired the community to make use of those techniques in speaker recognition as well. Deep Belief Networks (DBN) have been used in [22] as unsupervised feature extractors for speaker

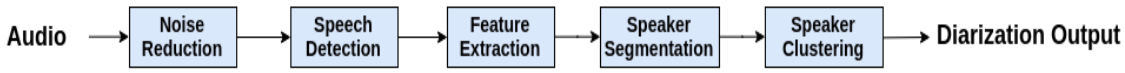


Figure 2.8 – *Block Diagram of a Speaker Diarization System*

identification. Different combinations of Restricted Boltzmann Machines (RBM) have been used in [23] to model i-vectors. RBMs have also been used to extract pseudo-i-vectors from acoustic features and i-vectors [24]. Deep Neural Networks (DNN) are used in an adaptation process to model target and impostor i-vectors discriminatively [25]. They have also been used to extract Baum-Welch statistics for super-vector and i-vector extraction. Nowadays, the BottleNeck Features (BNF) are used in Deep Learning techniques for Speaker Recognition tasks. In [27] it has been implemented to have outperformed the baseline system. In Speaker Recognition tasks, Speaker Diarization and Speaker Tracking are two of the major applications. They are briefly discussed in the following sections.

2.3 Speaker Diarization

Speaker Diarization refers to identify which speaker speaks when, in a conference recording or meeting. Speaker Diarization answers the question: Who Speaks When? Speaker Diarization systems refers to the systems that performs Speaker Segmentation of the speech signal and then Speaker Clustering of the developed segments into homogeneous groups. All these steps are performed within the same input stream. The Speaker Diarization task assumes no prior knowledge about the speakers' identities or how many speakers are participating in the conference. This is a step by step process which involves some Front-end Processing, Speaker Segmentation and Speaker Clustering followed by some hypothesis result. Figure 2.8 shows a block diagram of various steps involved in a Speaker Diarization system.

2.3.1 Front-end Processing

The Front-End Processing usually, includes several processes such as, Speech Enhancement and Noise Reduction, Speech Activity Detection and Feature Extraction.

1. *Speech Enhancement and Noise Reduction:* Normally, the speech signal is noisy because

of communication limitations. The noise part should be suppressed in order to enhance the output Signal to Noise ratio. This can be achieved by using the Wiener's filtering approach [1].

2. *Feature Extraction:* There exist various parametrization features for the diarization process such as MFCC (Mel Frequency Cepstral Coefficients), LPC (Linear Predictive Coding), LFCC (Linear frequency cepstral coefficients) and PLP (Perceptual Linear Predictive) etc but the most common features, these days used in voice recognition are the MFCCs, which are also the features used in the UPC Speaker Diarization System [1].

3. *Speech Activity Detection:* The audio is not always a full time speech signal. There are, sometimes, small gaps in-between the speech frames called silence or non-speech frames, or even music. In order to reduce the computational complexity of the process, these silence frames are removed to avoid processing useless frames. This process is called speech activity detection (SAD) or Voice Activity Detection (VAD). An energy based SAD is used in the UPC Speaker Diarization system which counts the energy contents of each frame. A threshold is set to decide the speech and non-speech frame. Energy feature is extracted for each frame and is compared with the threshold. If the value crosses the threshold, the frame is considered speech frame otherwise a non-speech frame.

2.3.2 Speaker Segmentation

The goal of audio segmentation is to detect the points in time, in the audio streams at changes between different speakers or acoustic environments. It is better to segment the audio and make homogeneous regions with respect to the changes in speaker, conditions of the environment and channel. In order to detect target speakers in the audio stream the audio is segmented in this way. The content in the audio also have to be considered. For example the audio portions of different contents and nature must be handled differently. There are portions of music and noise which can be deleted. The task might be to design a separate recognition system for telephone speech. Since a same speaker may appear multiple times

in several conditions it is not easy to create a correct segmentation. Many systems, these days, are based on the Bayesian Information Criterion (BIC) but there exist many other segmentation algorithms [1]. So, Speaker Segmentation produces segments of the audio at those points where there appears significant changes between different speakers. There are various methods to perform segmentation such as energy based segmentation, model based segmentation (for example Gaussian Mixture Model GMM) and metric based segmentation for example Generalize Likelihood Ratio (GLR) and Bayesian Information Criterion (BIC). The energy-based segmentation only detects boundaries at silence/non-silence positions. In general, this idea has no direct dependency with the acoustic changes in the speech data. The other two approaches, i.e: the model-based and the metric-based segmentation algorithms, are common in relying on putting a threshold to the meaningful measurements. This strategy has a lack in stability and robustness. The important drawback in these two is that there is no generalization to acoustic conditions that are hidden or unseen [1].

2.3.3 Speaker Clustering

In this step the segments from the previous step are clustered according to their homogeneity. The criteria is defined by some sort of distance measure or likelihood measure and iteratively the clusters are merged or split depending upon the clustering approach. The segmentation and clustering can be done in this step by step approach or in a one-loop system where both are performed in one single iteration. The second approach uses Viterbi realignment in which the audio is re-segmented based on the current results from the clustering thus avoids errors [1]. There exists many approaches to the Speaker Clustering task for example: Vector Quantization (VQ), Self-Organizing Maps (SOM) and Spectral clustering (SC). However, in the UPC Speaker Diarization system, clustering is done by using Agglomerative Hierarchical Clustering (AHC). Hierarchical Clustering is based on iteratively splitting or merging clusters until an optimum number of clusters is reached. When the optimal stage is reached, the system stops any more iterations and gives an output hypothesis. An example of the Speaker Diarization output is shown in Figure 2.9. AHC can be approached in two different ways depending upon splitting or merging clusters:

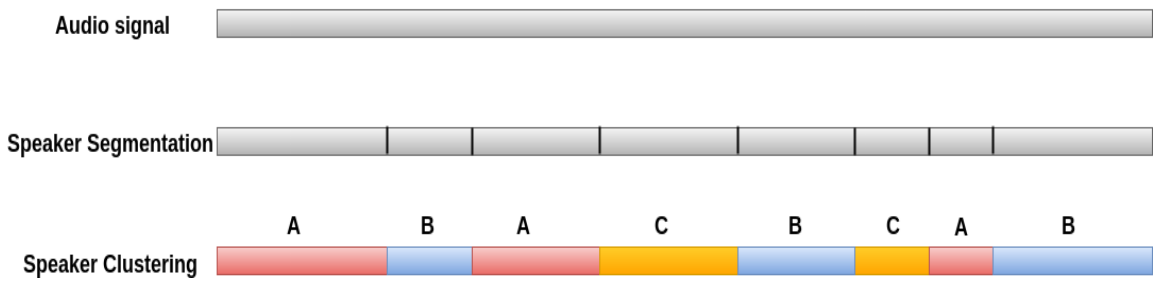


Figure 2.9 – *Speaker Diarization Output*

- **Bottom-Up Agglomerative Hierarchical Clustering:** The system starts with maximum number of clusters and keeps merging them iteratively, according to some criteria for example the BIC criteria. It is necessary to define the initial number of clusters in order to initiate the algorithm. An example of Bottom-Up AHC is shown in Figure 2.10.

- **Top-Down Agglomerative Hierarchical Clustering:** This is opposite to the first approach. Here the system starts with minimum number of clusters and iteratively splits them into new clusters until the optimal stage is reached. An example of Top-Down AHC is shown in Figure 2.10.

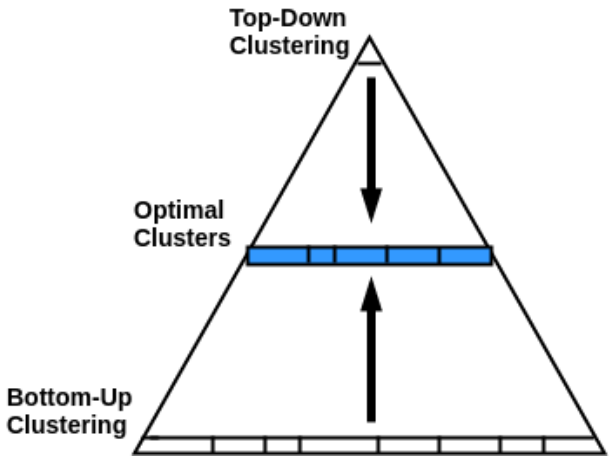


Figure 2.10 – *Agglomerative Hierarchical Clustering*

2.4 Speaker Tracking

In general, a Speaker Tracking system is a Speaker Diarization system followed by a Speaker Verification module. This is because the output of a Speaker Tracking system is to find out the position and duration where the target speaker speaks in the audio. As, the Speaker Diarization system gives the information about different speakers' clusters, it is simply to track the target speaker by verifying the clusters against the target speaker. This means that a unlike Speaker Diarization system, Speaker Tracking system need a prior knowledge of the target speakers. For this purpose, a pre-enrollment of the target speakers is performed and the target speaker models are stored in a database. Figure 2.11 shows this schematics for Speaker Tracking. The output of a Speaker Tracking system does not only contain information about different clusters but also it tells about the identity of the speaker speaking in that particular cluster.

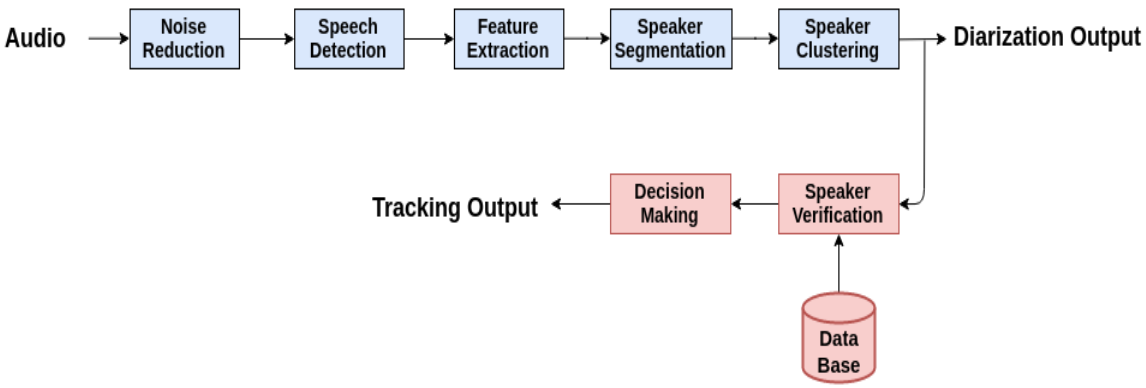


Figure 2.11 – *Block Diagram of a Speaker Tracking System*

In the past decade, various Speaker Tracking systems are developed, so for, based on different segmentation and modeling strategies. In [2] Speaker Tracking system based on speaker turn detection is presented. In this system, the segmentation step is performed based on speaker turn points. The speaker turn points are detected be a Generalized Likelihood Ratio (GLR) as used in [14] and [15]. Initially small segments are developed from the speech data and then fro every two adjacent segments two hypothesis are assumed:

- H_0: Both the segments belongs to a similar speaker.

$$X_{12} = X_1 \cup X_2 \sim \mathcal{N}(\mu_{12}, \Sigma_{12}) \tag{2.8}$$

- H_1: Both the segments are uttered by two different speakers.

$$X_1 \sim \mathcal{N}(\mu_1, \Sigma_1) \tag{2.9}$$

$$X_2 \sim \mathcal{N}(\mu_2, \Sigma_2) \tag{2.10}$$

Then the Generalized Likelihood Ratio is computed as:

$$R = \frac{L(X_{12}, \mathcal{N}(\mu_{12}, \Sigma_{12}))}{L(X_1, \mathcal{N}(\mu_1, \Sigma_1)).L(X_2, \mathcal{N}(\mu_2, \Sigma_2))} \tag{2.11}$$

A dissimilarity distance is then computed as:

$$d = -log(R) \tag{2.12}$$

Finally a Speaker Verification is performed. In this step, GMM models have been used using the EM algorithm.

In [16], a similar kind of Speaker Tracking system is developed. The algorithm is based on speaker change detection in real time applications for broadcast news. It detects the speaker changing points in the audio and then performs a segmentation. GMM models are used to model the speakers. An automatic real time updating of speaker models and cluster merging strategy is applied here. For the Speaker Verification part, a fusion strategy is applied between the MFCC and LSP features. The decision is made using a Bayesian Decision function. In [18], a multimodal person discovery system in broadcast news is developed by UPC Image Processing Group and Speech Processing Group. In this system, three modalities are fused together to track a person in the broadcast news i.e: text, audio and video detection. For the audio part of this system, an Agglomerative Hierarchical Clustering Speaker Diarization is applied. The clusters are modeled by using GMM models and a Bayesian Information Criterion is applied for decision taking. The system output is used in fusion with the other two modalities to make a final hypothesis. In [19], another multimodal person discovery system in

broadcast news is developed by UPC Image Processing Group and Speech Processing Group. In time, two modalities are fused together to track a person in the broadcast news. One is face detection and the other is voice detection. For voice detection, a Speaker Tracking system is developed. This system uses a speaker segmentation by using a dissimilarity measure between overlapping segments. They use i-vector modeling for with cosine similarity score. The output of this Speaker Tracking system is used in intersection with visual detection of the target person. So, in short, there are various strategic approaches to develop a Speaker Tracking system. Some systems use Speaker Segmentation and Speaker Verification, while some systems run a Speaker Verification after a Speaker Diarization output. All these strategies may/or may not differ in modeling approaches. Because, in either strategy, speaker modeling is necessary. Thus, in terms of speaker modeling approaches, there have been a significant research in the past decade, discussed in Section 2.2.

Chapter 3

Proposed Speaker Tracking System

This chapter explains the detailed theory and formulation of the proposed Speaker Tracking system. The chapter is divided it three main sections. The first section explains the theory of Speech Activity Detection. The second section explains initial considerations and features extraction. The theory of Speaker Segmentation module of the system is discussed in detail. It explains various steps for speaker turn detection and final segmentation. The third and last section of this chapter explains two different approaches for Speaker Verification module of the system. The first approach relies on Gaussian Mixture Models of the candidates and the second approach uses their i-vectors representation.

In this thesis, a simple but convenient Speaker Tracking system has been developed for the applications of recorded audios like meetings, conferences, television talk shows, NEWS bulletins and other multiple speakers scenarios. The objective is to answer the question *'when the target speaker speaks?'* For this purpose, it is necessary to pre-enroll the target speaker in the system, in order to perform the Speaker Verification. This means that the system must have a prior knowledge of the target speakers. This step differs a Speaker Tracking system from a Speaker Diarization system, where the system does not necessarily require any prior knowledge of the speaker appearing in the audio. In the proposed Speaker Tracking system, the recorded audio is segmented according to the speaker turn points. The speaker turn points are detected using the divergence shaped distance used in [5]. In the next step, these points are passed through another confirmation test where some of the false detected

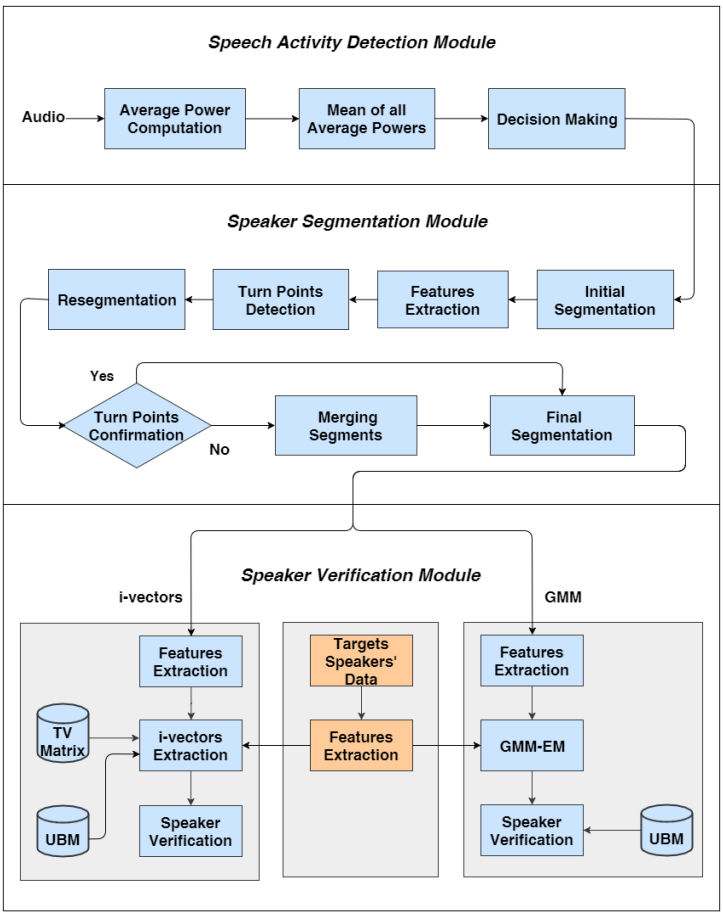


Figure 3.1 – *A Brief Flowchart of the Proposed Speaker Tracking System*

turn points are dropped. Now the system is left with optimal number of speaker turns. The corresponding segments are clustered and new segments are formed. This, literally means that every adjacent segment belongs to a different speaker. After this, a Speaker Verification of the target speakers against the segments, is performed in order to know *'to which target speaker the current segment belongs?'* Gaussian Mixture Models are developed using the Expectation Maximization algorithm for all the segments. In the meanwhile, target speaker are also enrolled in the system and Gaussian Mixture Models are developed using the same EM algorithm. Thus, all the segments are tested against all the target speakers using the state-of-the-art GMM models. Further more, for this last step, identity vectors, as used in [6], are developed for all the segments and target speakers. ALIZE 3.0, a free toolkit for Speaker Recognition tasks, is used to perform this step [7]. A brief flowchart of the proposed system

is shown in Figure 3.1. It is mainly composed of three main modules, i.e. Speech Activity Detection, Speaker Segmentation and Speaker Verification. In the following Sections, these modules are explained in detail.

3.1 Speech Activity Detection

There are, often, silence or non-speech frames which needs to be removed to avoid useless computations in the process. This process is called Speech Activity Detection (SAD) or Voice Activity Detection (VAD). Usually, a Speech activity Detection is important before every speaker recognition system. In the tasks of speaker segmentation and speaker verification, the removal of non-speech frames gives more accurate results. In [9] an energy based Speech Activity Detection and a hybrid Speech Activity Detection are implemented. The signal statistics are important choosing the type of Speech Activity Detection. For signal which has a high Signal to Noise Ratio, an energy based Speech Activity Detection is recommended. In this speaker tracking system, an energy-based Speech Activity Detection is performed which counts the energy content of every frame as implemented in [10]. It compares the average power of each frame with the mean of all the frames average power. A threshold value is defined in order to classify speech and non-speech parts. The block diagram is shown in Figure 3.2 . In the first step, the system computes average power of each frame using a 30ms length Hamming window each 10ms. This is computed using the following expression:

$$P_x(k) = \frac{1}{N_{length}} \sum_{n=0}^{N_{length}-1} |x(n - k.N_{shift}).v(n)|^2 \tag{3.1}$$

Where the $x(n)$ is the audio signal, N_{length} is the sample length of the $v(n)$ Hamming window and N_{shift} is the number of samples that correspond to the 10ms shift parameter.

In the second step, the system computes the mean of power averages of all the frames. In the third step, a ratio between the average power of each frame and the mean of power averages of all frames is computed. If this ratio is higher than the predefined threshold, the frame is considered as a speech frame otherwise the frame is considered as a non-speech frame.

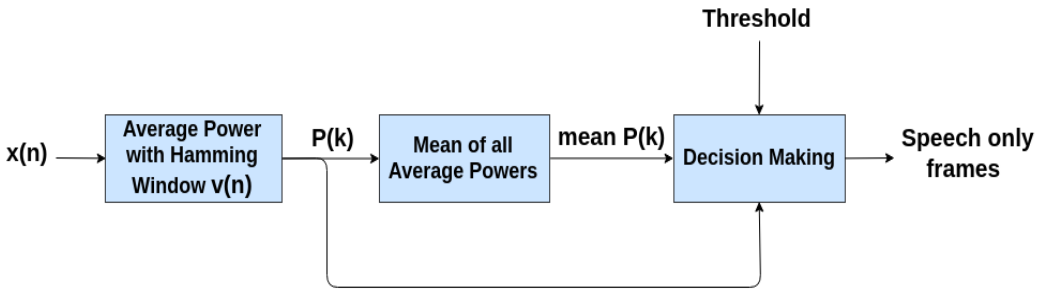


Figure 3.2 – *Block Diagram of Speech Activity Detection*

3.2 Speaker Segmentation

Speaker Segmentation is one of the main modules in this Speaker Tracking system. The idea is to, first, make segments of the audio where it is assumed to have different speakers and then verify the segments against the target speakers. Thus, after the Speech Activity Detection, the second step is to segment of the audio depending on the speaker changing points. The audio signal is cut into segments on those points where the speaker changes. The goal is to answer *in which portions of the audio different speakers appear?* For segmentation, it is necessary to detect the positions where there is a speaker change. The speaker changing points in time are marked to ensure accurate segmentation. There is an initial unsupervised segmentation in the very first step. This helps in marking the speaker turn points. The audio is segmented according to this. In the next step, the speaker turn points are confirmed and some of false detection are dropped. As a result the corresponding segments are merged. A re-segmentation is applied after merging some of the segments. In the following subsections, all the steps for Speaker Segmentation are explained in detail.

3.2.1 Initial Segmentation

Once the system detects the speech frames, they are, initially segmented into small segments. This is a uniform segmentation. The idea is to detect the speaker changing point after every small possible duration. Keeping enough speaker data for processing, the size of small segments is set to three seconds. The small segments have an overlapping of 2.75 seconds with each other. Thus the resolution is 250 ms. This means that the system looks for a speaker

turn point or speaker change after every 250 ms. Figure 3.3 shows initial segmentation of the audio. Starting from the first sample, small Segment 1 lasts up to 3 seconds. Small Segment 2 starts from 0.25 seconds and lasts up to 3.25 seconds. Similarly small Segment 3 starts from 0.5 seconds and lasts up to 3.5 seconds. In this way, all the speech parts of the audio are segmented into small segments of three seconds each.

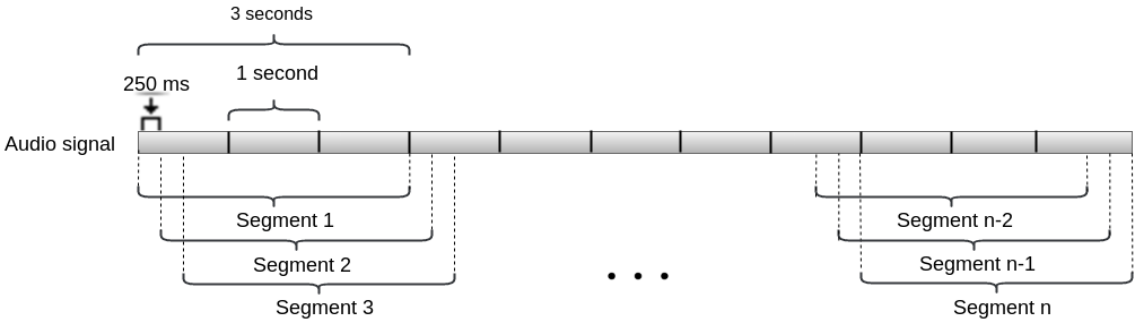


Figure 3.3 – *Initial Segmentation with an Overlap of 2.75 Seconds*

3.2.2 Feature Extraction

The next step is to extract useful features for each of the small segments in the process. Human voice can be characterized by within speaker variability called intra-speaker variability. This is a kind of variation in which two speech signals from same speaker are wrongly classified as from two different speakers. The other type of variability which classifies different speakers is called inter-speaker variability. Because of the variations in features of human voice, a large research has been done on choosing the types of features. The characteristic to keep is that the features must be capable to more or less characterize a defined speaker. There exist various parametrization features for the Speech/Speaker Recognition tasks, such as Mel Frequency Cepstral Coefficients (MFCC), Line Spectral Frequency (LSF) features, Linear Prediction Coefficients (LPC), Linear Frequency Cepstral Coefficients (LFCC) and Perceptual Linear Prediction Coefficients (PLPC) etc. The most common features, these days, used in voice recognition are the MFCC which are also used in the UPC Speaker Identification System [1]. In this Speaker Tracking system, unlike [5], only MFCC are used as the main features for modeling the target speakers and the segments.

Figure 3.4 – *MFCC Feature Extraction*

Mel Frequency Cepstral Coefficients (MFCC)

Mel Frequency Cepstral Coefficients features are computed as a vector of coefficients that represent the short-term power spectrum of a speech signal. It is based on a linear cosine transform of a log power spectrum on a mel-scale of frequency [11]. Figure 3.4 shows the full block diagram of MFCC features extraction. The speech signal is passed through a windowing block which frames it into 25ms frames. A Hamming window is applied here with an overlapping of 10ms. Then a magnitude squared Discrete Fourier Transform (DFT) is computed.The frequencies are then wrapped by applying a mel-scale filter bank. Finally the Discrete Cosine Transform (DCT) of the log filter-bank energies is computed and MFCC features are extracted at the output.

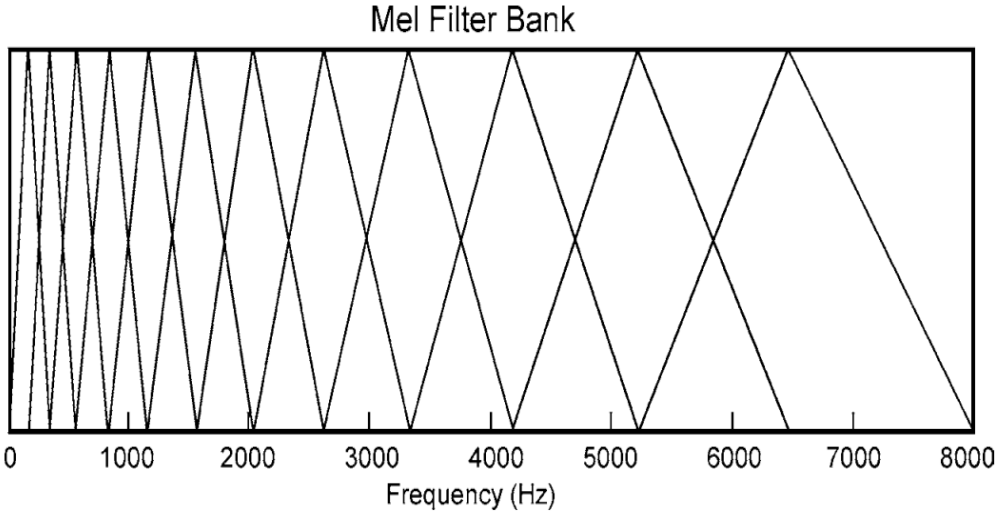


Figure 3.5 – *Mel-Scale Filter Bank [10]*

The mel-scale of frequency, which is applied trough a filter bank shown in Figure 3.5, is implemented as an approximation of the auditory human system performance, where the capacity of discerning the difference between two closely spaced frequencies decreases on the

highest frequencies. The mel-scale maps the frequency of a tone, or pitch, onto a linear scale. The scale is linear up to 1000 Hz and logarithmic between 1000 Hz and 8000 Hz. Thus, more importance is given to the lower frequencies as compared to the higher frequencies.

3.2.3 Speaker Turn Points Detection

A crucial task is to find the positions where there is a speaker change in the audio. The goal is to mark the speaker turn points and segments the audio accordingly. For this purpose, a dissimilarity measure between the MFCC features of consecutive small segments is computed [8]. As a resolution of 250 ms second is kept in the initial segmentation step, it is guaranteed that the dissimilarity is checked after every 250 ms. Suppose C is the estimated covariance matrix of the the MFCC features of an initial small segment. Then the dissimilarity measure between two adjacent small segments, $Segment_1$ and $Segment_2$, is given by:

$$D = \frac{1}{2} tr[(C_1 - C_2)(C_2^{-1} - C_1^{-1})] \tag{3.2}$$

Where C_1 is the covariance matrix of the features of $Segment_1$ and C_2 is the covariance matrix of the features of $Segment_2$. So, in general, the dissimilarity measure between two adjacent small segments, $Segment_i$ and $Segment_{i+1}$, is given by:

$$D = \frac{1}{2} tr[(C_i - C_{i+1})(C_{i+1}^{-1} - C_i^{-1})] \tag{3.3}$$

Where C_i is the covariance matrix of the features of $Segment_i$ and C_{i+1} is the covariance matrix of the features of $Segment_{i+1}$.

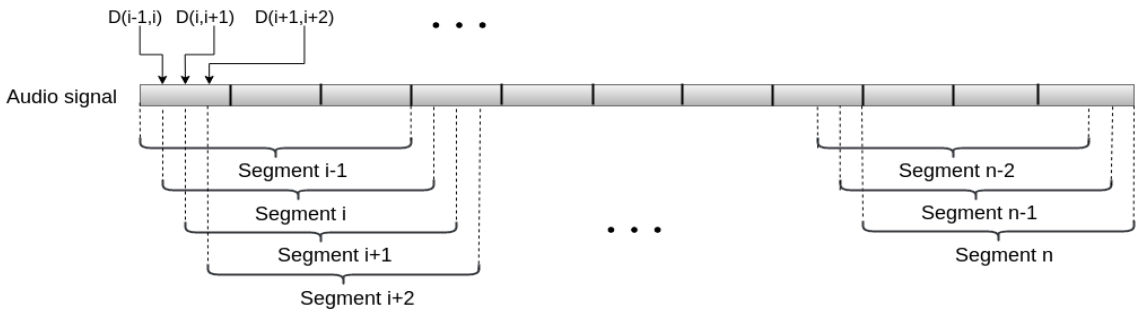


Figure 3.6 – *Divergence Distance between Adjacent Small Segments*

This is called the divergence shape distance measure [8]. Thus, the divergence shape distance is computed for every two adjacent small segments with a resolution of 250 ms. Suppose $D(i-1,i)$ is the distance between $Segment_{i-1}$ and $Segment_i$, $D(i,i+1)$ is the distance between $Segment_i$ and $Segment_{i+1}$ and $D(i+1,i+2)$ is the distance between $Segment_{i+1}$ and $Segment_{i+2}$, as shown in Figure 3.6. In order to detect the speaker turn point, the distances of three adjacent small segments are compared. For a speaker turn point at *ith* small segment, the following conditions must be satisfied:

$$D(i,i+1) > D(i,i+2) \tag{3.4}$$

$$D(i,i+1) > D(i-1,i) \tag{3.5}$$

$$D(i,i+1) > Threshold \tag{3.6}$$

This means if the dissimilarity between $Segment_i$ and $Segment_{i+1}$ is greater than one previous distance, one next distance and a constant factor, a speaker turn point is marked here. Figure 3.7 shows a graphical representation of the distance measure between adjacent segments.

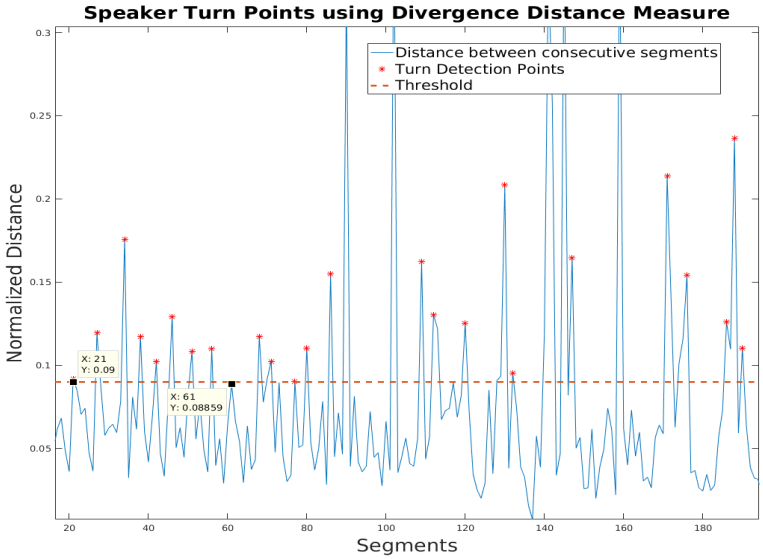


Figure 3.7 – *Graphical Representation of Divergence Distance Between Adjacent Small Segments, with Constant Threshold Value.*

The horizontal axis represents segment number and the vertical axis shows the amplitude of max-normalized distances. The red crosses represent a speaker turn point. The fixed threshold value is shown in dotted line. The amplitude of distances vary abruptly and it is difficult to detect all the important ones. Sometimes, a higher threshold value will miss some points and a lower one will have false detection. For example the two points shown in Figure 3.7 and 3.8 respectively. With a fixed threshold value, the first point is detected but the second

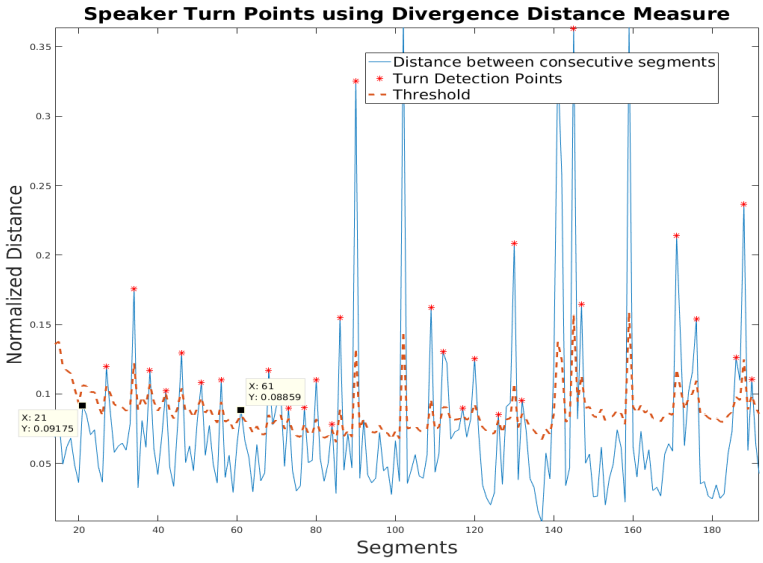


Figure 3.8 – *Graphical representation of Divergence Distance between adjacent small segments, with adaptive threshold value.*

one is missed. In order to avoid this kind of problem, a threshold adaptation is done, as in [5]. The threshold value is no more fixed, but is defined by the average of previous distance values. Suppose $Threshold_i$ is the threshold value at *ith* small segment, then generally the $Threshold_i$ is defined as:

$$Threshold_i = \frac{\alpha}{N} \sum_{n=0}^{N} D(i-n-1, i-n) \tag{3.7}$$

Where α is a scaling factor and is to be tuned for good performance according to the statistics of data. In Figure 3.7 the first point at $x = 21$ is detected and the second point at $x = 62$ is missed. Solving the same problem with an adaptive threshold method as in equation 3.7, the

two points are detected the other way around. This can be seen in Figure 3.8. The horizontal axis represents segment number and the vertical axis shows the amplitude of max-normalized distances. The red crosses represent a speaker turn point. Adaptive threshold value is shown in dotted line.

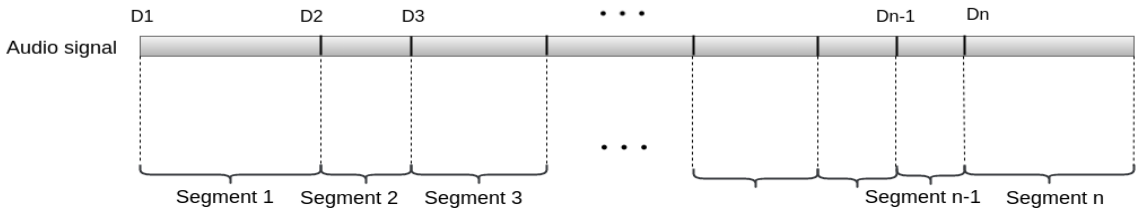


Figure 3.9 – *Re-Segmentation of the Audio on Speaker Turn Points*

The audio signal is now re-segmented according to the speaker turn points detected in the previous step. This is a supervised segmentation as the system uses the positions detected before. Thus the size of the segments after this step are not necessarily equal. The size and even the number of segments depends on the speaker turn points detected. Figure 3.9 shows the re-segmentation step.

3.2.4 Final Segmentation

At this stage, the speaker turn points detected may not be accurate. Because the α parameter for threshold adaptation is normally kept on a lower side in order not miss any speaker change. So there is a high probability of false detection. Thus a confirmation stage is applied to see if the point is an actual speaker turn or a false detection. For this purpose, a universal background model (UBM) is trained and the GMM models [3] of the segments are developed. A UBM represents a large population of speakers, to better represent the characteristics of the background speakers. This general model is referred as universal background model [4]. The Expectation Maximization (EM) algorithm for developing the UBM and the GMM models is used in this system. A dissimilarity check is performed between the segments (with each other) and with the UBM. A summation of the distances between the first 3 seconds of a segment and the previous segment is computed, according to the following formula [5]:

$$D_{new} = \sum_{i=1}^{k} w_i D(C_i, C_m) \tag{3.8}$$

Where D_{new} is the new divergence distance with the previous segment, w_i is the weight of the Gaussian component i of the previous segment, k is the total number of components in the Gaussian Model of the previous segment, C_i is covariance of the ith component of the previous segment and C_m is the covariance of the first 3 seconds of a segment.

In a similar fashion, A summation of the distances between the first 3 seconds of a segment and the UBM components is computed, according to the following formula [5]:

$$D_{newUBM} = \sum_{i=1}^{k} w_i D(C_i, C_m) \tag{3.9}$$

Here, D_{newUBM} is the new divergence distance with UBM, w_i is the weight of the Gaussian component i of the UBM, k is the total number of components in the UBM, C_i is covariance of the ith component of the UBM and C_m is the covariance of the first 3 seconds of a segment.

A test is run, if the current segment is from the same speaker as of the previous one or it belongs to the UBM. For this purpose a ratio of the two distance summations, i.e: D_{new} and D_{newUBM} is computed. If the ratio is higher than a threshold λ, the speaker turn point is accepted as a true point, otherwise it is dropped.

$$\frac{D_{new}}{D_{newUBM}} = \begin{cases} > \lambda & accept \\ \leq \lambda & drop \end{cases} \tag{3.10}$$

At this stage the system is optimal about the final speaker turn points. Those segments for which the turn points are dropped in the previous step, are merged together to form final segments. So on the basis of the accepted turn points, the system has segmented the audio into final segments. Each consecutive segment belongs to a different speaker and is having information of the corresponding speaker. At this point, the system has information about the size of the final segments and their starting and ending positions in time. The segmentation step is crucial because in the next step, the speaker verification process relies on this step. An error in the segmentation will become even bigger, in the verification step. This will act like a snowball which increases in size as it rolls down. It is better to avoid the

miss detection in segmentation stage rather that the false detection. The reason behind this is, if the system misses a speaker turn point, in the verification process there will be an error. The segments/data from different speakers will be processed in a same segment. On the other hand, if the system has a false detection in the segmentation step, it means that the same speaker data is segmented into two. But it will be overcome in the verification stage, as it will label the two segments to the same speaker.

3.3 Speaker Verification

This is the final step of our system. In this speaker tracking system, the strategy is to segment the audio properly on speaker turn points, and then perform a speaker verification test on the segments. The target speakers are modeled and pre-enrolled in the system. Thus the final segments, which are the results of the last step, are used for verification. The goal is to answer *to which target speaker, the segments belong?* Two different speaker verification strategies, based on the modeling technique, are applied here. They are compared in order to improve the performance of the speaker tracking system. In the first strategy Gaussian Mixture Models are developed both for target speakers and segments of the audio. GMM is the state-of-the-art technique in speaker verification, nowadays [1]. On the other hand i-vectors are extracted both for the target speakers and segments of the audio.

3.3.1 Gaussian Mixture Models

In this approach, the system first enrolls the target speaker by developing the Gaussian Mixture Models of all the target speakers. Then All the final segments are modeled using the same technique. Finally a Speaker Tracking is applied on both the data sets for final decision. Following is the detail of every step:

- **GMM for Target Speakers:** In this tracking system, the target models are pre-enrolled in the database. For, experimental purpose, the information about target speaker segments is taken from the manual transcriptions in the database used for experiments. GMM models of the target speaker are developed by using EM algorithm as it is the state-of-the-art modeling technique in speaker verification and identification

these days [1]. The complexity of the GMM models depends on how much speaker data/frames we have in a particular segment [1], according to the automatic model complexity selection given as:

$$k = \left\lfloor \frac{N}{Rcc} + \frac{1}{2} \right\rfloor \tag{3.11}$$

Where k is the number of components in the Gaussian Mixtures Model, N is the number of frames belonging to the segment and Rcc is a constant factor. Here, $Rcc = 7$ is used as in [1].

- **GMM for Segments Speakers:** The segments of the final segmentation step are modeled using the same EM algorithm for Gaussian Mixture Models (GMM). The complexity of the GMM models, again, depends on how much speaker data/frames we have in a particular segment defined in 3.11 used in [1].

- **Speaker Tracking:** target speakers' models are tested against the segments speakers' models for speaker tracking. For this step, a summation of the distance measures is computed, for measuring the dissimilarity between target speaker and all the segments. For every target speaker, the distance is given by:

$$D^m_{Tracking} = \sum_{i=1}^{k} w_i^m D(C_i^m, C_{final}) \tag{3.12}$$

Where $D^m_{Tracking}$ is the divergence shape distance between a final segment and a target speaker m, w_i^m is the weight of the ith Gaussian component of the mth target speaker, k is the total number of components in the Gaussian Model of the mth target speaker, C_i^m is covariance of the ith component of the mth target speaker and C_{final} is the covariance of a final segment (a final segment is obtained from the final segmentation step). In this way all the target speakers are tested against all the final segments. Also, all the final segments are tested against a UBM, which is already trained in the segmentation step. Thus, a distance measure is computed using the following expression:

$$D_{Tracking-UBM} = \sum_{i=1}^{k} w_i D(C_i, C_{final}) \tag{3.13}$$

Here, $D_{Tracking-UBM}$ is the divergence distance between a final segment and the UBM, w_i is the weight of the ith Gaussian component of the UBM, k is the total number of

components in the UBM, C_i is covariance of the ith component of the UBM and C_{final} is the covariance of a final segment. After this, a ratio of $D^m_{Tracking}$ and $D_{Tracking-UBM}$ is computed, which decide if the final segment in consideration belongs to the target speaker or the UBM.

$$Ratio = \frac{D_{Tracking}}{D_{Tracking-UBM}} \tag{3.14}$$

The process can be represented in a matrix form. Suppose R_{mxn} is the matrix obtained after computing equation 3.14. The m rows of R represent the target speakers and the n columns represent the final segments.

$$R = \begin{bmatrix} R_{T1,S1} & R_{T1,S2} & R_{T1,S3} & \dots & R_{T1,Sn} \\ R_{T2,S1} & R_{T2,S2} & R_{T2,S3} & \dots & R_{T2,Sn} \\ R_{T3,S1} & R_{T3,S2} & R_{T3,S3} & \dots & R_{T3,Sn} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_{Tm,S1} & R_{Tm,S2} & R_{Tm,S3} & \dots & R_{Tm,Sn} \end{bmatrix} \tag{3.15}$$

In matrix R, the mth row has information of dissimilarity between mth target speaker and nth segment. A Segment n is assigned to a target m if the following condition is satisfied:

$$min(R_{T1,Sn}, R_{T2,Sn}, R_{T3,Sn} ..., R_{Tm,Sn}) < \lambda_1 \tag{3.16}$$

Where λ_1 is a threshold to decide if the segment does not belong to any of the target speaker. Here, the strategy is to chose the closest target for a segment using the minimum distance. The target with minimum dissimilarity is a potential candidate for the segment under test. Then, the threshold limit decides if the dissimilarity is low enough. If the distance is high enough to cross $\lambda 1$, the corresponding segment does not belong to the target speaker to be tracked and vice versa.

An Example: An example of the tracking is illustrated in Figure 3.10. The ratio scores are depicted in every column for a segment against every target. In this example, there are 10 segments and 6 target speakers. In the first column, segment $S1$ is tested against all targets and it is seen that a minimum value appears for target $T2$ (colored in yellow). A threshold is set to a value of 8, in the red row. As, the minimum value, (5 for $T2$)

is less than the threshold, the corresponding segment, $S1$ is assigned to $T2$. Similarly, all the segments are assigned to their corresponding minimum valued Tn (colored in green), provided that it satisfies the threshold condition. In column 5, the minimum value for segment $S5$ is 9, which, in this case, does not satisfies the threshold condition, so $S5$ is not assigned to any target. In this case, the segment is labeled by **NO Target** as shown in the red colored block of the last row of Figure 3.10.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
T1	10	6	15	11	9	5	13	11	14	8
T2	5	9	9	7	11	3	8	8	9	11
T3	8	8	7	11	15	7	6	10	7	7
T4	9	10	15	7	10	8	13	19	9	6
T5	7	12	16	5	18	11	15	6	8	17
T6	14	12	9	9	19	9	10	13	9	5
min	5	6	7	5	9	3	6	6	7	5
Thr	8	8	8	8	8	8	8	8	8	8
Result	T2	T1	T3	T5	NO Target	T2	T3	T5	T3	T6

Figure 3.10 – *Illustration Example of Target Speakers Tracking Using GMM Models.*

The horizontal axis represents ratio scores of segments **Sn** with target **Tm** in the vertical axis. The minimum values are shown in row **min**.

3.3.2 Identity Vectors

A second approach for speaker verification, is the use of identity vectors (i-vectors). In this approach, the system uses the i-vectors for representing a target speaker or the final segments. As, it has been proved in [6] that i-vectors out-performs the state-of-the-art, Gaussian Mixture Models approach for speaker recognition tasks. In this Speaker Tracking system, for the speaker verification step, i-vectors representation of the speakers has been implemented. ALIZE-3.0, a free toolkit [7] is used for extracting and testing of i-vectors.

- **i-vectors for Target Speakers:** As the target speakers must be enrolled in the system, the first step is to extract i-vectors for the target speakers. First, the MFCC features are extracted for target speaker data, and then it is fed into the *i-vectors extractor* module

of ALIZE-3.0 to extract i-vectors for each target speaker. Before extracting i-vectors, the Total Variability Matrix is trained for the system. The rank of Total Variability Matrix is kept same as the size of i-vectors.

- **i-vectors for Segments Speakers:** The final segments from the segmentation step are also represented in i-vectors form, by using the same ALIZE-3.0 toolkit. First, MFCC features are extracted for the final segments and then it is fed to the *i-vectors-extractor* module of ALIZE-3.0 to extract the corresponding i-vectors, using the trained Total Variability Matrix.

- **Speaker Tracking:** Once the system extracts i-vectors, it is ready to perform the verification of all the final segments against all the target speakers. In the meanwhile, a UBM is trained by the *Train-World* module of ALIZE-3.0 for performing the i-vectors test. Then the speaker verification is performed by the *i-vectors-Test* module of ALIZE-3.0, where it computes various type of scoring techniques. The segments' i-vectors are tested if they resemble one or more i-vectors from the target speakers. In this system, only cosine scoring technique (see Equation 2.7) for i-vectors test is used. The cosine score represents the resemblance between i-vectors in a score of range -1 to $+1$. The more the score is closer to 1, the higher is the resemblance and the more likely is the segment to belong to this target. The more the score is closer to zero, the lower is the resemblance and the less likely is the segment to belong to this target. Similar to the speaker tracking described in Section 3.3.1, the system computes a matrix of cosine scores between all segments and all target speakers. Suppose C_{mxn} is the matrix obtained after computing the cosine scores. m represent the target speakers and n represent the segments. In matrix C, the mth row has information of cosine score (similarity) between mth target speaker and nth segment.

$$C = \begin{bmatrix} C_{T1,S1} & C_{T1,S2} & C_{T1,S3} & \dots & C_{T1,Sn} \\ C_{T2,S1} & C_{T2,S2} & C_{T2,S3} & \dots & C_{T2,Sn} \\ C_{T3,S1} & C_{T3,S2} & C_{T3,S3} & \dots & C_{T3,Sn} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{Tm,S1} & C_{Tm,S2} & C_{Tm,S3} & \dots & C_{Tm,Sn} \end{bmatrix} \tag{3.17}$$

A Segment n is assigned to a target m if the following condition is satisfied:

$$max(C_{T1,Sn}, C_{T2,Sn}, C_{T3,Sn}..., C_{Tm,Sn}) > \lambda_2 \tag{3.18}$$

Where λ_2 is a threshold to decide if the segments does not belong to any of the target speakers. A potential candidate in target speakers is selected for a segment under test, by chosing the maximum cosine score. Then the cosine score is compared with λ_2. If the score is low and couldn't cross λ_2, then the corresponding segment does not belong to the target speaker to be tracked and vice versa.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
T1	0.2	0.3	- 0.3	0.7	0.9	- 0.5	0.2	0.5	0.6	0.1
T2	0.5	0.6	0.2	0.5	- 0.5	0.2	- 0.9	0.1	0.1	- 0.8
T3	0.8	- 0.5	- 0.1	0.3	- 0.5	- 0.7	0.1	- 0.8	- 0.9	- 0.1
T4	- 0.9	- 0.1	0.4	- 0.9	- 0.9	0.8	- 0.9	0.6	- 0.9	- 0.9
T5	- 0.3	0.5	- 0.8	- 0.3	0.6	- 0.1	0.6	- 0.3	0.8	- 0.3
T6	- 0.4	- 0.7	- 0.4	0.4	- 0.4	- 0.7	0.5	0.4	- 0.7	0.6
max	0.8	0.6	0.4	0.7	0.9	0.8	0.6	0.6	0.8	0.6
Thr	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Result	T3	T2	NO Target	T1	T1	T4	T5	T4	T5	T6

Figure 3.11 – *Illustration Example of Target Speakers Tracking Using I-Vectors.*

The horizontal axis represents cosine scores of segments **Sn** with target **Tm** in the vertical axis. The maximum values are shown in row **max**.

An Example: An example of the tracking using cosine scoring, is illustrated in Figure 3.11. The cosine scores are depicted in every column for a segment against every target. In this example, there are 10 segments and 6 target speakers. In the first column, segment $S1$ is tested against all targets and it is seen that a maximum cosine score value appears for target $T3$ (colored in yellow). A threshold is set to a value of 0.5, in the red row. As, the maximum value, (0.8 for $T3$) is greater than the threshold, the corresponding segment, $S1$ is assigned to $T3$. In this way, all the segments are assigned to their corresponding maximum valued Tn (colored in green), provided that it satisfies

the threshold condition. In column 3, the maximum cosine score value for segment $S3$ is 0.4, which, in this case, does not satisfies the threshold condition, so $S3$ is not assigned to any target. In this case, the segment is labeled by **NO Target** as shown in the red colored block of the last row of Figure 3.11.

Chapter 4

Experiments and Results

This chapter explains the speech database used and the experiments carried out in this thesis. The first section is about the experimental setup and database. Then the actual experiments carried out are explains with the corresponding results analysis. The results are, mainly, shown with the help of graphical representations. This Speaker Tracking system has three major modules, i.e Speech Activity Detection, Speaker Segmentation and Speaker Verification. The Speech Activity Detection module is not under the experimental research of this thesis. An energy based Speech Activity Detection has been used as in [10]. The experiments carried out aims at the other two modules of the system, i.e Speaker Segmentation and Speaker Verification. Thus, this chapter contains two sets of experiments. The first set of experiments is carried out for the Speaker Segmentation part. Various parameters are under consideration in these experiments. The second set of experiments is carried out for Speaker Verification, with analysis of different parameters. The experiments for Speaker Verification, further, has two different approaches depending upon the modeling technique used for target speakers and the audio segments. For the first approach, which is the state-of-the-art in Speaker Recognition (Gaussian Mixture Models for Speaker Verification), various experiments are performed while considering different parameters. On the other hand, different experiments are carried out for the second approach, i-vectors representation, for Speaker Verification. For this approach different parameters are considered in importance for better performance of the system.

4.1 Experimental Setup and Database

In this speaker tracking system, all the experiments are performed using audios from Agora database. This database contains the recordings of 34 TV shows of Catalan public broadcast TV3. The shows are highly moderated debates with a high variation in topics and invited speakers. In total the database consists of 68 files with a total audio duration of 43 hours. Each audio file corresponds to half show of an airing day with an average duration of 38 minutes. In this Speaker Tracking system, 38 files are used with an approximate length of 24 hours. The transcription follows the general guideline generated within the TC-STAR project for European Parliament Plenary Sessions but it was extended to include additional information as the language, background condition, silence/voice segmentation, speaker segmentation and acoustic events. The transcriptions have four layers. Transcriptions follow the TRS format produced by the Transcriber transcribing tool. The whole database recordings contain segments from 871 adult Catalan speakers. Of them, 441 are male speakers, 113 are female speakers and 317 are unknown speakers.There are 157 adult Spanish speakers. Of them, 83 are male speakers, 29 are female speakers and 45 are unknown speakers. These speakers may originate from different accents. Speakers are unbalanced in gender favoring male speakers in total duration. All the shows were performed in a closed TV studio.

4.2 Evaluation Metrics

In the context of observations and experiments, this Speaker Tracking system has two modules i.e: Speaker Segmentation and Speaker Verification. These modules are evaluated in their respective terms in order to know the system's performance. Following is the detail for evaluation metrics of each of them.

Speaker Segmentation

The performance for Speaker Segmentation module is evaluated in terms of False Alarm Rate (FAR), Miss Detection Rate (MDR), Precision, Recall and F1 Measure [17]. As the segmentation module relies on finding the speaker turn points, the evaluation is directly based on how accurately the speaker turn points are detected. The detected turn points are

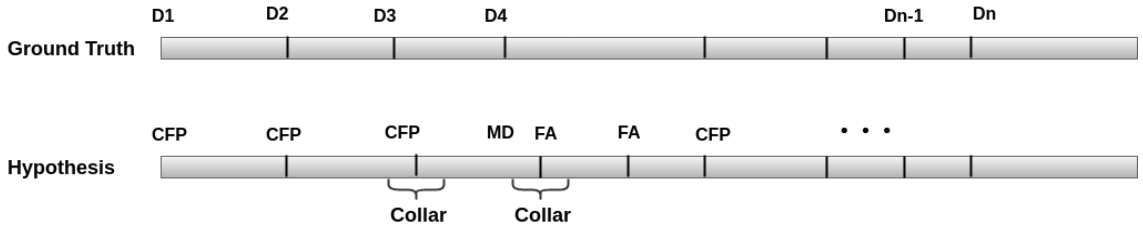


Figure 4.1 – *Evaluation Metrics for Speaker Segmentation*

categorized into Correctly Found Points (CFP) and False Alarms (FA). The are also some points which are not detected by the system but actually exist. They are termed as Miss Detection (MD). Figure 4.1 shows the CFP, FA and MD for a test example. In this example D1 and D2 of the Ground Truth are detected correctly and labeled as CFPs. D3 and D4 has shifts but as it is clear from the figure that D3 falls under the range of the collar, it is labeled as CFP. Unlike D3, D4 does not fall under this range, so it is labeled as MD. And as it appears in a position where there is no turn point in the Ground Truth, it is labeled as FA. In this way the rest of the terms are calculated as:

$$FAR = \frac{FA}{GT + FA} \tag{4.1}$$

$$MDR = \frac{MD}{GT} \tag{4.2}$$

$$Precision = \frac{CFP}{CFP + FA} \tag{4.3}$$

$$Recall = \frac{CFP}{CFP + MD} \tag{4.4}$$

$$F1Measure = 2\frac{(Precision)(Recall)}{Precision + Recall} \tag{4.5}$$

Where GT is the Ground Truth points.

Speaker Verification

The tracking module gives a hypothesis result. This hypothesis is evaluated against a Ground Truth in order to evaluate the system performance. Basically the tracking module labels the audio recording according to the target speakers. Figure 4.2 depicts an example. In this example Target 1 is the speaker of interest. There is a ground truth which shows the duration where Target 1 appears in the audio. This is shown in pink. In the hypothesis, the duration of target 1 found by the system are shown in light green. In the labeling the True Positive (TP) duration is shown in dark green. False Positive (FP) and False Negative (FN) are shown in red while True Negative (TN) is shown in white. In this way, all these four terms are computed for all target speakers appearing in the audio. The overall terms are computed as the weighted sum of all the individual terms as follows:

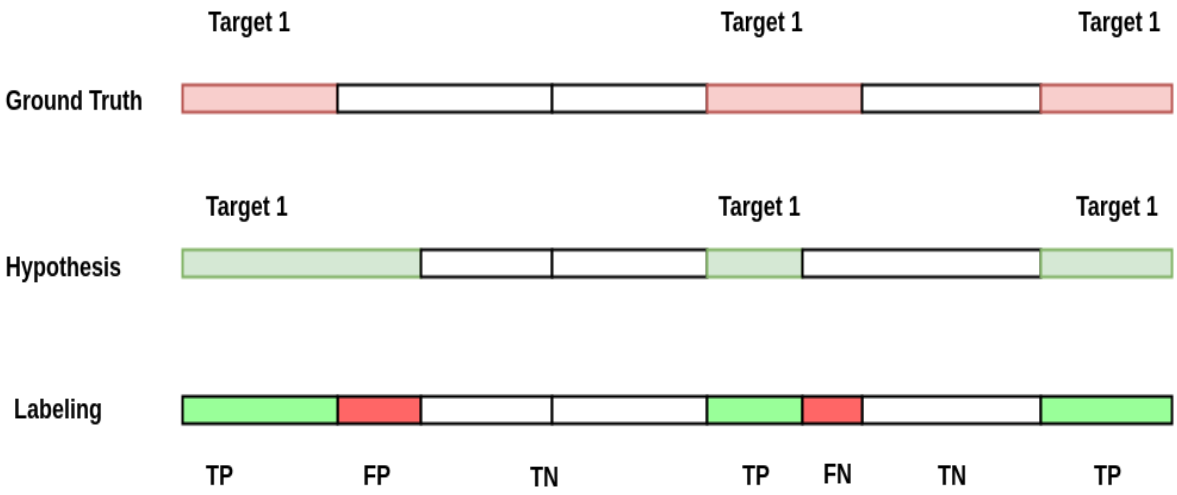


Figure 4.2 – *Evaluation Metrics for Speaker Verification*

$$TP = \sum_{i=1}^{k} \frac{dur_i}{dur_{all}} (TP_i) \tag{4.6}$$

Where k is the total number of target speakers, dur_i is the duration of the $Target_i$ in the audio, dur_{all} is the total duration of the audio and TP_i is the True Positive duration found in hypothesis for $Target_i$. Similarly the other three terms are calculated as:

$$FP = \sum_{i=1}^{k} \frac{dur_i}{dur_{all}}(FP_i) \tag{4.7}$$

$$TN = \sum_{i=1}^{k} \frac{dur_i}{dur_{all}}(TN_i) \tag{4.8}$$

$$FN = \sum_{i=1}^{k} \frac{dur_i}{dur_{all}}(FN_i) \tag{4.9}$$

In this thesis, the speaker verification is evaluated in terms of False Acceptance Rate (FAR), False Rejection Rate (FRR), Precision, Recall and F1 Measure. These terms are computed as follows:

$$FAR = \frac{FP}{TP + FP} \tag{4.10}$$

$$FRR = \frac{FN}{TN + FN} \tag{4.11}$$

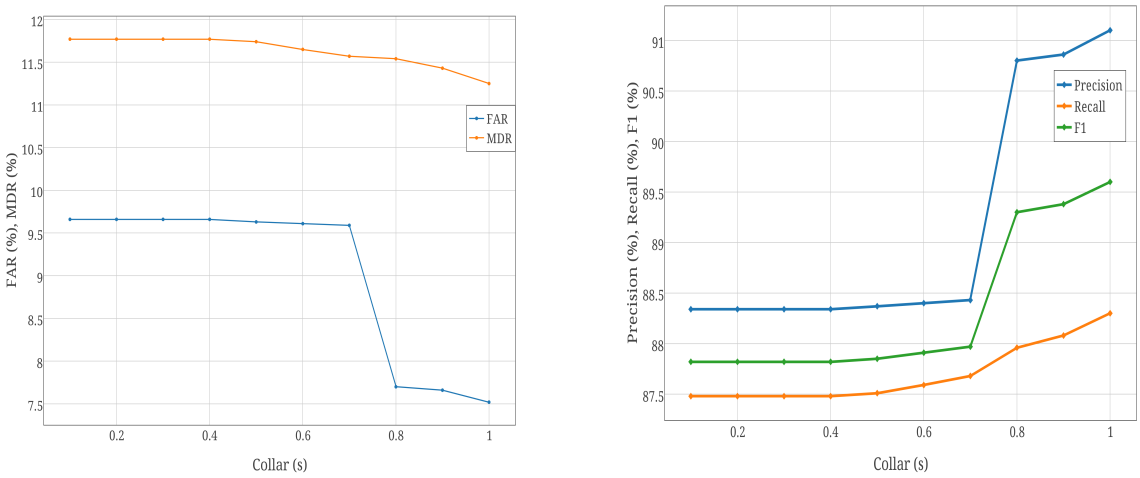
$$Precision = \frac{TP}{TP + FP} \tag{4.12}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.13}$$

$$F1Measure = 2\frac{(Precision)(Recall)}{Precision + Recall} \tag{4.14}$$

4.3 Speaker Segmentation

Various experiments are performed for the Speaker Segmentation task. In these experiments an energy based Speech Activity Detection, as discussed in Section 3.1, is used for



(a) Collar Against FAR and MDR

(b) Collar Against Precision, Recall and F1

Figure 4.3 – *Speaker Segmentation Results*

detection the speech frames for the audios used. As discussed in Section 3.2.2, MFCC features of order 20 are used for the experiments. A frame length of 25 ms with an overlap of 10 ms is used to extract the MFCC features. Thus the system extracts 20 features after every 10 ms. In this case, for a small segment of 3 seconds, there are 300 frames. Every frame is represented by 20 coefficients. Different parameters are set, in order to achieve best performance. For the initial segmentation discussed in Section 3.2.1, segments of 3 seconds each are recommended here that give best performance. This is a moderate size for initial segmentation. The system has enough speaker data in 3 seconds. This means that the system has a small segment of a minimum duration of 3 seconds. At the output of Speech Activity Detection module, those speech parts which are smaller than 3 seconds are not considered in the audio and are discarded. The adjacent small segments are overlapped with each other by 2.75 seconds. Furthermore, in Equation 3.6, the α parameter (a scalar for threshold adaptation) is set to 8 while the λ parameter (a threshold for speaker turn points confirmation), in Equation 3.10, is set to 0.65, for best performance for this particular database. With these set of parameters the Speaker Segmentation experiments are performed.

An offset or collar is tuned with respect to the evaluation terms for Speaker Segmentation.

The collar value is important because this system relies on detecting the speaker turn points. If the turn point detected by the system lies within the acceptable range of collar value, it is taken as Correctly Found Point or True Positive. If the turn point is not in range of collar, and it is supposed to be there as per the Ground Truth, then this is a Miss Detection. Similarly, if a turn point is detected and it is not present in the Ground Truth, then it is a False Detection. In this experiment, several collar values are tested to evaluate the performance of the speaker segmentation step. Figure 4.3 shows the evaluation results for Speaker Segmentation task with respect to different collar values. These results are taken as a weighted average of the individual results for 38 audio files from the database used. The weight of each file depends on the duration of the file. Starting from a collar value of 0.1 second to 1 second, Figure 4.3a shows the results in terms of False Alarm Rate (FAR) and Miss Detection Rate (MDR) in percentages respectively. The goal is to minimize the FAR and MDR. Figure 4.3b shows the corresponding Precision, Recall and F1 Measure in percentages respectively. In this case the goal to maximize Precision Recall and F1 Measure. It is seen from the figures that as the collar value increases, the performance improves. This is because increasing the collar value means accepting more tolerance. So, the more tolerance is accepted, the more distinct speaker turn point is accepted as Correctly Found Point, so the better is the result and vice versa.

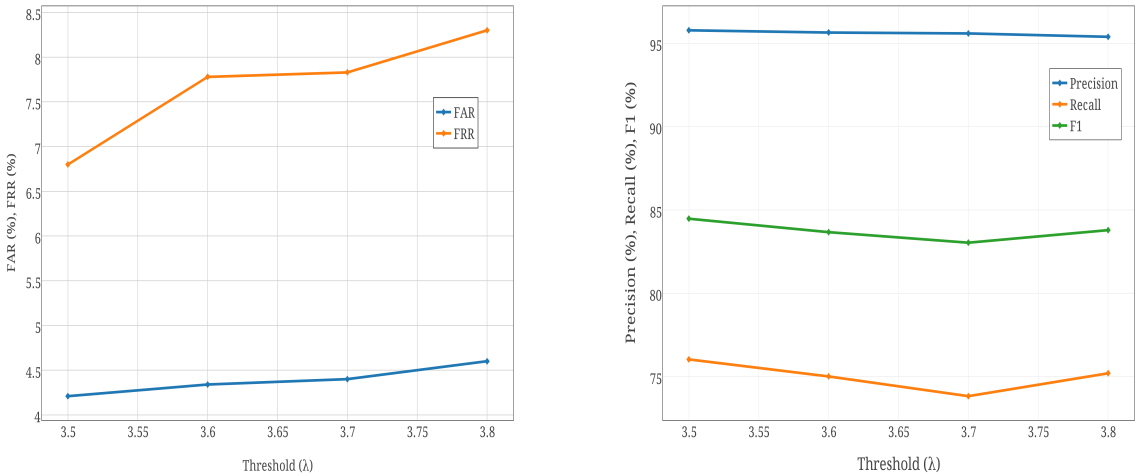
For a collar value in range 0.1 to 0.4, the system performance is the same, which means that some of the speaker turn points are not in the vicinity of collar value up to 0.4 seconds but beyond that. Usually, in biometric systems, the system performance is dependent on application sensitivity. For high security applications, False Detection is more avoided than Miss Detection. On the other hand, for collar value in range 0.5 to 0.7, there comes a slight improvement in False Alarm Rate and Miss Detection Rate. Beyond 0.7 the performance further improves which correspondingly give good Precision and Recall. In Figure 4.3, it is seen that a collar value of 1 second gives the best performance. Thus the collar value selection depends on how much False Detection and Miss Detection are acceptable in the process.

4.4 Speaker Verification

In this thesis, as discussed in Section 3.3, two different approaches are applied for Speaker Verification. The first approach is considered as the state of the art approach for speaker modeling, from the last decade. The second approach is an emerging technique from last few years for representing different speakers. In this thesis experiments are performed in order to implement both the approaches for the best performance. Following is the detail of experiments performed in both the cases:

Gaussian Mixture Models Approach

Different experiments are performed for the tracking task using Gaussian Mixture Models of the target and segments speakers. Two parameters are tuned in these experiments for best performance. A UBM of complexity 512 is used for these experiments, as per used in [12]. Figure 4.4 shows the average performance of the system for 38 audio files of the database. The results are depicted in terms of False Acceptance Rate (FAR), False Rejection Rate (FRR) in Figure 4.4a. It is seen that the FAR is more consistent and is on a lower



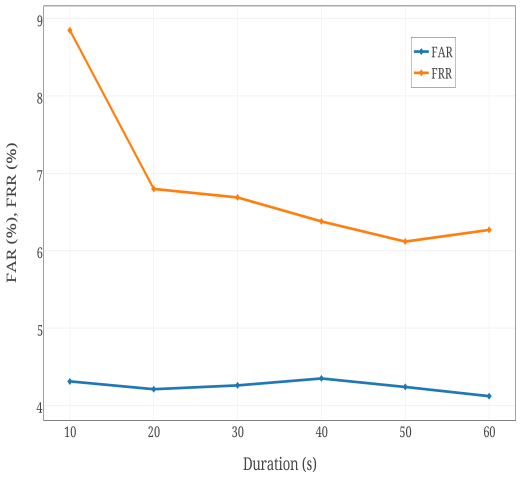
(a) Threshold (λ) Against FAR and FRR

(b) Threshold (λ) Against Precision, Recall and F1

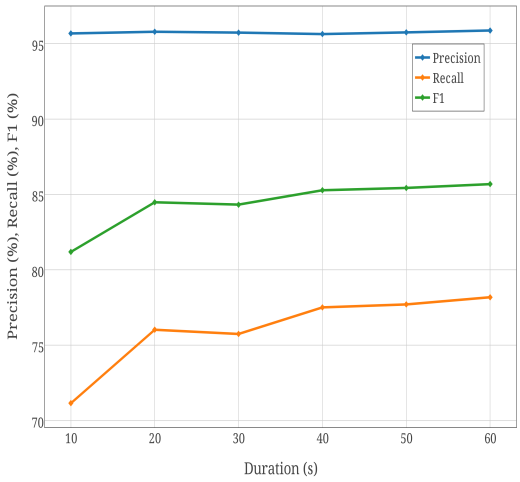
Figure 4.4 – *Speaker Tracking Results Using GMM (λ_1 Selection)*

side as compared to FRR. Thus a lower point on the FAR graph will give good results and can be an operating point. In this experiment different values of the threshold λ_1 are tested and it is seen that for $\lambda_1 = 3.5$, the both the FAR and FRR are at their corresponding lower positions. Similarly, on the other hand, in Figure 4.4b, the Precision, Recall and F1 Measure are depicted in percentages respectively. It is seen from the figure that, that the corresponding Precision, Recall and F1 Measure are on higher side for $\lambda_1 = 3.5$. Generally the threshold depends on the statistics of the audios in the database used. With this value of λ_1 the system has a False Acceptance Rate of 4.2% and a False Rejection Rate of 6.8%. The corresponding Precision, Recall and F1 Measure are 95.79%, 76.03% and 84.48% respectively. At this stage, the system performs with a good precision but the recall is in the average range. This experiments is performed using only 20 seconds of training data for target speakers pre-enrollment. This duration of the target speakers pre-enrollment is not a global standard and it depends on the application sensitivity. The idea is to minimize the training data and thus reduce computational cost but not the performance.

Another experiment is performed for selecting how much training data should be enough for target speakers pre-enrollment. The system has been tested for different amount of training data for target speakers. The results are depicted in Figure 4.5. FAR and FRR are plotted against the training data duration, in Figure 4.5a. It is seen in the figure that as the training data increases, the performance of the system improves and vice versa. Similarly the graphs for Precision, recall and F1 Measure against the training data duration goes on increasing when the training data duration increases. This is shown in Figure 4.5b. This is very obvious because the more data is available for training the system, the better the system performs. On the other hand, a system with more training data takes long time to train and thus the computational cost is also high. So there is trade-off between these two terms. A system of high security application may require more training data with low False Acceptance Rate. In this experiment, the False Acceptance Rate is in a very low variation range, as the duration is increased but the False Rejection Rate gets a significant improvement for the training data of duration 60 seconds. As compared to a training data of duration 20s, the False Acceptance Rate drops to 4.12 from 4.21 while the False Rejection Rate drops to 6.27 from 6.80. Thus it improves the recall by almost 3%.



(a) Target Speakers Duration Against FAR and FRR



(b) Target Speakers Duration Against Precision, Recall and F1

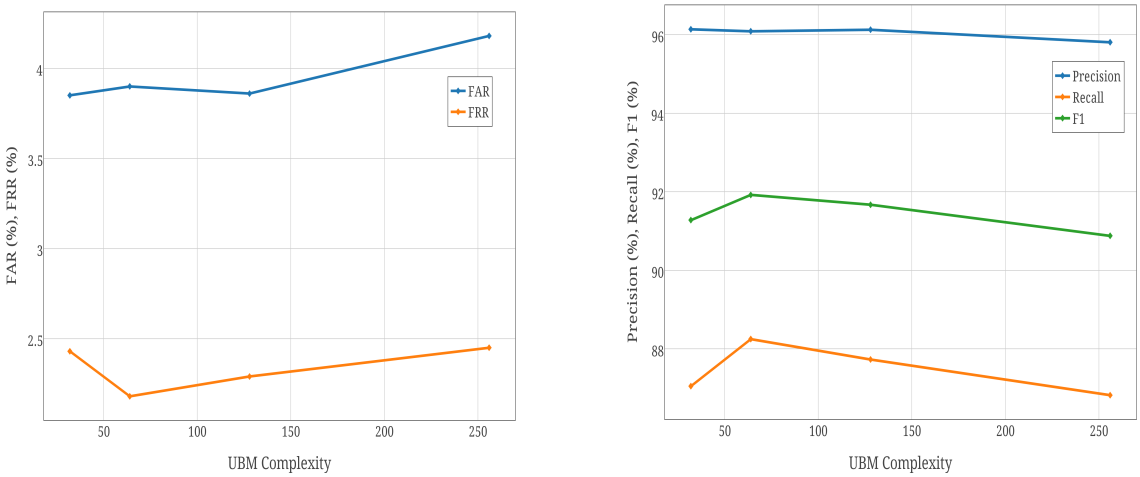
Figure 4.5 – *Speaker Tracking Results Using GMM (Target Speakers Duration)*

The corresponding values of Precision, Recall and F1 Measure are 95.87%, 78.18% and 85.69% respectively. From Figure 4.5b, it is seen that a training data duration of 60 seconds shows best performance in these terms. The graph of Precision is more consistent throughout the experiment but the graph of Recall has a significant improvement at this optimal point. Thus, it is recommended to train the target speaker models by as much data as the system can afford providing that it is not degrading the system in terms of computational cost and processing time issues.

Identity Vectors Approach

The second approach for speaker representation, implemented in this thesis, is the identity vector representation approach. For this approach ALIZE-3.0 toolkit is used for experiments. ALIZE-3.0 has different configuration parameters which needs to be tuned for better performance of particular cases. These parameters are, normally, selected according to application scenario. In this thesis, two of these parameters are tested for the i-vector representation of speakers. The first parameter is to chose the complexity for the *Train_World* module. This

defines the complexity of the UBM. Figure 4.6a and 4.6b depicts the results of this tracking system using i-vectors representation for different UBM complexity values. The results are



(a) UBM Complexity Against FAR and FRR

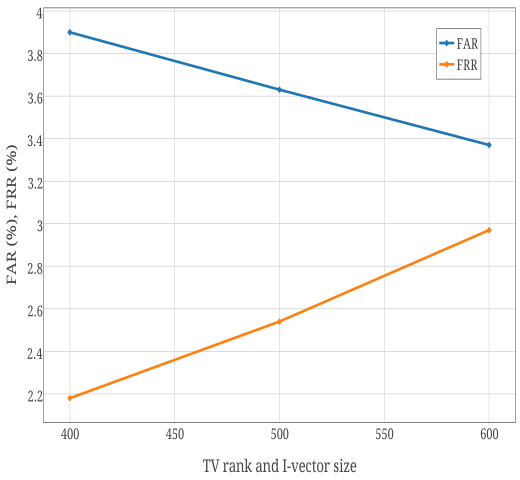
(b) UBM Complexity Against Precision, Recall and F1

Figure 4.6 – *Speaker Tracking Results Using I-Vectors (UBM Complexity)*

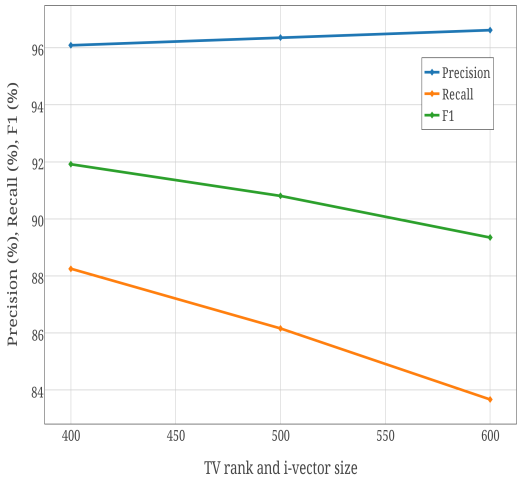
expressed in the terms of False Acceptance Rate (FAR), False Rejection Rate (FRR), Precision, Recall and F1 Measure. In this experiment the cosine similarity of the i-vectors tests is used for decision making as discussed in Section 3.3.2. Different values of UBM complexity have been used here. From the Figure it is seen that for a UBM complexity of 32, 64 and 128, the False Acceptance Rate has a small variation range which correspondingly plots the Precision in a small variation range. On the other hand there is an improvement in False Rejection Rate for UBM complexity of 64 which gives the best Recall of 88.25%. At this point, the corresponding values of False Acceptance Rate and False Rejection Rate are 3.90% and 2.18%. These results clearly out-perform the results obtained using GMM models.

The second parameter which is tested, in this thesis, is the rank of Total Variability Matrix and the size of i-vectors. For this purpose another experiment is performed keeping the best UBM complexity from the previous experiment, which is 64. Normally, both the rank of Total Variability Matrix and the size of i-vectors, are kept between 400 and 600 for Speaker

Verification tasks. In this experiment the same idea is respected. Figure 4.7a depicts the results, in terms of FAR and FRR. The corresponding plots for Precision, Recall and F1 Measure are also shown in Figure 4.7b.



(a) TV Rank and I-Vector Size Against FAR and FRR



(b) TV Rank and I-Vector Size Against Precision, Recall and F1

Figure 4.7 – *Speaker Tracking Results Using I-Vectors (TV Rank and I-Vector Size Selection)*

From the figures it is seen that as these two parameters increase, the FAR decreases. On the other hand the FRR increases. So there is a trade-off between FAR and FRR here. The selection depends on the application scenario. It is seen that the lowest FAR is achieved for a Total Variability and i-vector size of 400. The corresponding graphs for Precision, recall and F1 Measure in Figure 4.7b. It is seen that the highest Recall is achieved at this point. Thus the best recall, in these experiments, is 88.25%, which is far better than the GMM approach.

4.5 Results Comparison

Comparing both the approaches, GMM and i-vectors representation for speaker verification, the results show that the later approach outperforms the former one in this speaker tracking system. Individually, the different evaluation metrics terms are improved while using the i-vectors approach. The comparison is shown in Table 4.1. It is seen that the FAR drops

Approach	FAR(%)	FRR(%)	Precision(%)	Recall(%)	F1(%)
GMM	4.12	6.27	95.87	78.18	85.69
i-vectors	3.90	2.18	96.09	88.25	91.92
Improvement	**5.34**	**34.76**	**0.23**	**12.88**	**7.27**

Table 4.1 – *Comparison of Tracking Results Using GMM and I-Vectors Approaches*

to 3.90% from 4.12% with an improvement of 5.34%. The FRR has a significant decrease and drops to 2.18% from 6.27% with an improvement of 34.76%. Similarly, the Precision, Recall and F1 Measure drops to 96.09%, 88.25% and 91.92% from 95.87%, 78.18% and 85.69% respectively with improvements of 0.23%, 12.88% and 7.27% respectively. Thus, for best combination of parameters for i-vectors representation as compared to the best combination of parameters for GMM, the overall performance of the system improves by using i-vectors representation approach.

Chapter 5

Conclusion

In this thesis, a simple Speaker Tracking system is developed, with the goal to answer, for example, *'where does Nimra speak in the audio?'* In some applications, there might be a person of interest to be tracked in this manner, in an audio recording or conference meeting. In this context the person of interest is termed as target speaker. In this work, a target speaker is tracked in an audio recording. The system finds the time stamps where the target speaker appears in the recording. For this purpose the audio recording is first segmented into different speaker segments. Then the segments are verified against the target speaker and thus the goal is achieved.

In the first step, the audio recording is segmented. For this purpose, the points in time are detected where there might be a speaker change. This change is measured in terms of a dissimilarity measure between adjacent segments. The audio recording is segmented with respect to the speaker turn points. Then the initial segmentation is re-confirmed and some of the segments are clustered. For this step, Gaussian Mixture Models of the audio segments are developed using the Expectation Maximization algorithm. Finally the system is left with a final hypothesis of segments which corresponds to different speakers. The segmentation results are evaluated in terms of False Detection Rate, Miss Detection Rate, Precision, Recall and F1 Measure. For a collar value of 1 second, the best results of a False Detection of 7.52%, Miss Detection of 11.25%, Precision of 91.10%, Recall of 88.30% and F1 Measure 89.60% is achieved.

In the second step, the segments are verified against the target speakers for tracking. Two different approaches are applied for this step i.e Gaussian Mixture Models and identity vectors representation of the target and segments of the audio.

The first approach develops Gaussian Mixture Models for the target and segments of the audio. The system uses Expectation Maximization algorithm for this. A dissimilarity measure is computed for verifying the target speakers against the segments. The segments are also verified against a UBM using the same dissimilarity measure. A decision threshold is fixed for best performance of the system. The tracking results are evaluated in terms of False Acceptance Rate, False Rejection Rate, Precision, Recall and F1 Measure. For a decision threshold of 3.5 and training data of 60 seconds for target speakers, the best results of a False Acceptance Rate of 4.12%, False Rejection Rate of 6.27%, Precision of 95.87%, Recall of 78.18% and F1 Measure 85.69% is achieved.

The second approach represents the target and segments of the audio by identity vectors. The system uses ALIZE-3.0 for this. A similarity measure is computed for verifying the target speakers against the segments. In this system a cosine similarity is used for taking decision. A decision threshold is fixed for best performance of the system. The tracking results are evaluated in the same terms as the first approach. For an i-vector size of 400 , TV matrix rank of 400 and a training data of 30 seconds for target speakers, the best results of a False Acceptance Rate of 3.90%, False Rejection Rate of 2.18%, Precision of 96.09%, Recall of 88.25% and F1 Measure 91.92% is achieved. This approach out-performs the first approach with significant improvements of 34.76% in the False Rejection rate and 12.88% in the Recall.

Future Directions

This thesis aims on Speaker Tracking task in audio recordings. The main strategy is Speaker Segmentation using GMM models and Speaker Verification using both GMM and i-vectors approaches. The results, from Chapter 4, clearly indicates the the later approach out-performs the former approach by a big significant amount. Though, very few but acceptable, parameters are included in consideration in experiments for both these approaches. In addition, for the segmentation step, i-vectors can be tested and implemented for discrimina-

tion purpose. On the other hand, different i-vectors scoring techniques can be used which are available in ALIZE-3.0 toolkit. A Probabilistic Linear Discriminant Analysis (PLDA) test can also be considered for performance improvement of the system, in both the Speaker Segmentation and Speaker Verification steps. Also, there are different i-vector normalization approaches available in the ALIZE-3.0 toolkit. One can take advantage of this and this may add something to the system performance.

References

[1] J. Luque, "Speaker diarization and tracking in multiple-sensor environments", Ph.D. dissertation, Department of Signal Theory and Communications, Universitat Politècnica de Catalunya, Spain, 2012.

[2] J. Bonastre, P. Delacourt, C. Fredouille, T. Merlin, C. Wellekens. "A speaker tracking system based on speaker turn detection for NIST evaluation." *In Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, vol. 2, pp. II1177-II1180.

[3] D. A. Reynolds. "Speaker identification and verification using Gaussian mixture speaker models". *Speech communications*, vol. 17, no. 1, pp. 91-108, August 1995.

[4] D. A. Reynolds, T. F. Quatieri, R. B. Dunn. "Speaker verification using adapted Gaussian mixture models". *Digital signal processing*, vol. 10, no. 1, pp 19-41, January 2000.

[5] L. Lie, H. Jiang, H. Zhang. "A robust audio classification and segmentation method." *In Proceedings of the ninth ACM international conference on Multimedia, ACM*, October 2001, pp. 203-211.

[6] A. Nautsch, "Speaker verification using i-vectors, Evaluation of text-independent speaker verification systems based on identity-vectors in short and variant duration scenarios", M.S. thesis, Department of Computer Sciences, Hochschule Darmstadt University of Applied Science, Germany, 2014.

[7] A Larcher, J. F. Bonastre, B. G. Fauve, K. A. Lee, C. Lévy, H. Li, J. S. Mason, J. Y.

Parfait. "ALIZE 3.0-open source toolkit for state-of-the-art speaker recognition". *In Interspeech*, 2013, pp. 2768-2772.

[8] J. P. Campbell. "Speaker recognition: a tutorial." *Proceedings of the IEEE*, September 1997, vol. 85, no. 9, pp. 1437-1462.

[9] X. Anguera, M. Aguilo, C. Wooters, C. Nadeu, J. Hernando. "Hybrid speech/non-speech detector applied to speaker diarization of meetings". *In IEEE Odyssey- The Speaker and Language Recognition Workshop 2006*, June 2006, pp. 1-6.

[10] I. M. Miquel Angel, "UPC System for the 2015 MediaEval Multimodal Person Discovery in Broadcast TV Task", M.S. thesis, Department of Signal Theory and Communications, Universitat Politècnica de Catalunya, Spain, 2015.

[11] S. Furui. "Cepstral analysis technique for automatic speaker verification". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 2, pp. 254-272, April 1981.

[12] D. A. Reynolds, T. F. Quatieri, R. B. Dunn. "Speaker verification using adapted Gaussian mixture models". *Digital signal processing*, vol. 10, no. 1, pp 19-41, January 2000.

[13] D. A. Reynolds. "An overview of automatic speaker recognition". *In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2002, pp. 4072-4075.

[14] H. Gish, M. H. Siu, R. Rohlicek. "Segregation of speakers for speech recognition and speaker identification". *In Proceeding of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 1991, vol. 2, pp. 873-876.

[15] H. Gish, M. Schmidt. "Text-independent speaker identification". *IEEE signal processing magazine*, vol. 11, no. 4, pp. 18-32, October 1994.

[16] L. Lu, H. J. Zhang. "Speaker change detection and tracking in real-time news broadcasting analysis". *In Proceedings of the tenth ACM international conference on Multimedia*, December 2002, pp. 602-610.

[17] M. Kotti, V. Moschou, C. Kotropoulos. "Speaker segmentation and clustering". *Signal processing*, vol. 88, no. 5, pp. 1091-1124, May 2008.

[18] V. D. González, V. B. Vilaplana, J. R. Morros, H. Javier. "UPC system for the 2015 MediaEval multimodal person discovery in broadcast TV task." *In MediaEval 2015 Multimedia Benchmark Workshop*, 2015.

[19] I. Miquel, G. Martí, C. Carla, G. Bouritsas, E. Sayrol, J. R. Morros, H. Javier. "UPC system for the 2016 MediaEval multimodal person discovery in broadcast TV task." *In MediaEval 2016 Multimedia Benchmark Workshop*, 2016.

[20] W. M. Campbell, D. E. Sturim, D. A. Reynolds. "Support vector machines using GMM supervectors for speaker verification". *IEEE signal processing letters.* vol. 13, no. 5, pp. 308-311, May 2006.

[21] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, P. Ouellet. "Front-end factor analysis for speaker verification". *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788-798, May 2011.

[22] H. Lee, P. Pham, Y. Largman, A. Y. Ng. "Unsupervised feature learning for audio classification using convolutional deep belief networks". *In Advances in neural information processing systems*, 2009, pp. 1096-1104.

[23] T. Stafylakis, P. Kenny, M. Senoussaoui, P. Dumouchel. "Preliminary investigation of Boltzmann machine classifiers for speaker recognition". *In Odyssey*, 2012, pp. 109-116.

[24] O. Ghahabi, J. Hernando. "Restricted Boltzmann machine supervectors for speaker recognition". *In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 4804-4808.

[25] O. Ghahabi, J. Hernando. "Global impostor selection for DBNs in multi-session i-vector speaker recognition." *Advances in Speech and Language Technologies for Iberian Languages. Springer International Publishing*, November 2014, pp. 89-98.

[26] G. E. Dahl, D. Yu, L. Deng, A. Acero. "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition". *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30-42, January 2012.

[27] F. Richardson, D. Reynolds, N. Dehak. "Deep neural network approaches to speaker and language recognition". *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671-1675, October 2015.

[28] K. Sgouropoulos, E. Stergiopoulou, N. Papamarkos. "A dynamic gesture and posture recognition system". *Journal of Intelligent & Robotic Systems*, vol. 76, no. 2, pp. 283-296, November 2014.

[29] R. Rangslang. "Segment phoneme classification from speech under noisy conditions: Using amplitude-frequency modulation based two-dimensional auto-regressive features with deep neural networks". M.S. thesis, Department of Signal Processing and Acoustics, School of Electrical Engineering, Aalto University, Helsinki, Finland, 2016.