Speaker Diarization

BY

Nirali Naik

14MCEC16

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

MAY 2016

Speaker Diarization

Final Year Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

By

Nirali Naik

14MCEC16

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

MAY 2016

Certificate

This is to certify that the thesis entitled "Speaker Diarization" submitted by Ms. Nirali Naik (14MCEC16), towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this thesis, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. Priyank Thakkar
Associate Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad

Prof. Sapan H Mankad
Assistant Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Priyanka Sharma
Professor,
Coordinator M.Tech - CSE
Institute of Technology,
Nirma University, Ahmedabad.

Dr Sanjay Garg
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad

Dr P. N. Tekwani
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Nirali Naik**, Roll. No. **14MCEC16**, give undertaking that the Major Project entitled "**Speaker Diarization**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Date:

Place:

Endorsed by

Dr. Priyank Thakkar

Prof. Sapan H. Mankad

Abstract

We live in the century, where information is strength. Information can be saved in the form of text/multimedia. One such form is Speech audio files. To access those file easily and efficiently, Speaker Diarization is the best option. Speaker Diarization is all about "who spoke when?". This system takes in the input in form of an audio file. The diarization sytem makes the segments of speech file such that each segment is homogeneous. Here, homogeneous segment means that those region of speech contains speech from only one speaker. These segments will be given to the clustering module. This module will merge all the identical segments. To identify the segments from the same speaker, system has to build models for each speaker. These models are constructed using GMMs,using EM algorithm. Feature extraction techniques are applied for extracting speaker specific information. VOice activity detection algorithms are used to differentiate between speech/non-speech reginons and identifying speaker change points. Diarization Systems can be used in Movie analysis, Automatic speech segmentation, Rich transcription, Audio archiving and monitoring, Audio indexing and retrieval. This system doesn't know the number of speakers involved in the system in advance. We just know is the domain of audio file(which type of recording is this, i.e. telephone conversation, meeting room conversation etc.). The evaluation measure used for speaker diarization systems is Diarization Error Rate. This is computed using, Miss, false alarm and confusion. The ideal output expected using diarization system is the speech regions related to speakers, and the speaker lables.

Acknowledgements

It is my privilege and duty to acknowledge the kind of help and guidance received from several people including my faculty members, my well wishers, my friends and colleagues in preparation of this report. It would not have been possible to prepare this report in this form without their valuable help, cooperation and guidance. At the end of the journey, it is a pleasant task to express my thanks to all those who contributed in many ways to the success of this study.

At this moment of accomplishment, first of all I pay homage to my guides, Prof. Sapan H Mankad and Dr. Priyank Thakkar. This work would not have been possible without their guidance, support and encouragement. Under their guidance I learned a lot. Despite of their busy schedules, they used to review my progress, give their valuable suggestions and guidelines. I can only say a proper thanks to them through my future work.

I wish to record my sincere gratitude to Management of this college and to our respected Director, Dr. P N Tekwani, for making available library and laboratory facilities needed to prepare this report.

I take this opportunity to express gratitude to all of the Department faculty members for their help and support. I also thank my parents for the unceasing encouragement, support and attention.

I also place on record, my sense of gratitude to one and all, who directly or indirectly, have lent their hand in this venture.

- Nirali Naik

14MCEC16

Abbreviations

MFCC	Mel-Frequency Cpestral Coefficients.
RPC	Real Cepstral Coefficients
LPC	Linear Predictive Coding
LPCC	Linear Predictive Cepstral Coefficients
ASR	Automatic Speech Recognition
DER	Diarization Error Rate
VAD	Voice Activity Detection
ACP	Average Cluster Purity
ASP	Average Speaker Purity
DCT	Discrete Cosine Transform
SNR	Signal-to-Noise-Ratio
TDOA	Time Difference Of Arrival
BIC	Bayesian Information Criterion
GLR	Generalized Likelihood Ratio
RCL	Recall
PRC	Precision

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Definition

Speaker Diarization

This is the era of knowledge. Knowledge is built when we process the information. We, human beings, communicate mostly through speech. So, information is conveyed through speech. Speech can also be seen as continuous signals. And hence, it is necessary that we store and process these speech signals efficiently. With each passing day, capacity of storage, processing capabilities, transmission facilities are improving. And this gives rise to recorded speech. We try to keep a backup of every conversation, be it broadcast news or meetings. This gives a new direction to research domains. It will be helpful to get the knowledge from the recordings if we can somehow know who were the speakers involved in these recordings, at what time which particular speaker was speaking, etc. This is known as Speaker Diarization. Speaker Diarization thus can be summed up as "Who Spoke When?", which means converting the speech file into homogeneous segments, merging identical segments, and then labeling them. An audio may include non-speech parts like advertisements, noise, etc. The simplest level of the diarization process can be speech-nonspeech labeling. More advance version can be, marking who was speaking at which particular instant in the file, and

the duration of his/her speech. This also helps in rich transcription with the help of speech processing.

1.2 Motivation

Sometimes We face some situations like :

- What if a part of speech/conversation is missed ?
- What if something important I want to revise ?
- What if I want to know how many speakers are there in this audio file?
- What if I want to jump to a particular instant of time in an audio file ?
- What if I had left a meeting in between and then I want to hear remaining part from recording?

All these questions lead us to the same direction. If we somehow make a note of the speakers present in the speech conversation, and the time-instant of their speech, we do not need to traverse whole file manually, and can jump to the audio segment we are interested in. This gives the basic idea of doing work in speaker diarization process.

1.3 Issues in Speaker Diarization Process

In speaker diarization systems, we do not know the number of speakers involved in the audio file, even we neither know speakers' identities, nor their voice samples are given to the systems in advance.

For audio file of meeting room conversations, there are some issues existing related to overspeech detection and handling. The other isuue arises when there multiple distant microphones present in the meeting room. So, audio file of this room setup have to be processed in some different way.

Chapter 2

Literature Survey

2.1 Speaker Diarization as a Process

Speaker diarization process has certain stages and these stages collectively form the final diarized output. The output of the diarization system is graphically represented in the Fig.2.1 . All the stages of speaker diarization system have their specific tasks. The order in which they are performed and the algorithms they use can vary according to the requirements of the system to be developed. It is also possible to perform two stages concurrently. For each stage a separate section is dedicated. The common approaches used in each stage will be listed in the section dedicated to that stage. We give audio file to the system. And Diarization system returns multiple homogeneous speaker segments labeled as shown below. If the system has information about speakers' identity then the labels will be replaced by speakers' names. The white parts in the output are non-speech parts. And speech segments are colored and labeled as per various speakers.

The Fig. 2.2 shows the step by step working of the whole diarization process.

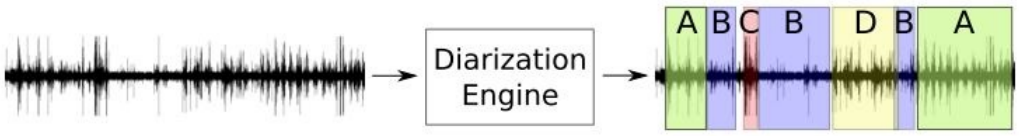

Figure 2.1: Overall Diarization process [2]

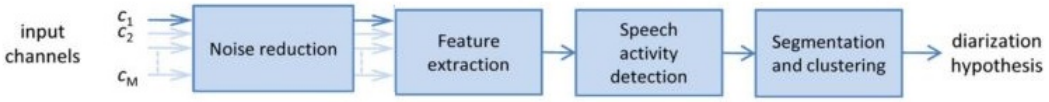

Figure 2.2: Steps involved in Speaker Diarization process [3]

2.1.1 Signal Processing as Pre-processing step

Noise Reduction

While working with real-life signals, we have to deal with the noisy signals also. The noise present in the signal may affect the processing step and may degrade the result somehow. To optimize the output result it in necessary to reduce the noise present in the file. So, the first step can be noise reduction if the audio file has higher noise volume. To filter out the noise,the solution can be *Wiener filtering*. Wiener filtering uses linear time-invariant filtering of the noisy signal and gives the estimate for desired signal. Wiener filtering is capable of giving the optimal solution by filtering out the noisy signal components, and gives the L^2-norm reconstruction [4].

Acoustic beamforming

As mentioned above, Speaker diarization takes place in meeting domain also. Meetings generally have more people involved compared to telephone conversations. And if the meeting involves large number of people, there may be the case that rooms have more than one microphones or the microphone arrays. These distant microphones will have their own recorded signals. To process this recorded audios, there is the need to deal with multiple channels. The characteristics and locations of the

microphones may vary. But this should not affect the diarizarion output.There are various approaches proposed to deal with this.

One approach proposed in [5], is to perform diarization on each microphone channel, and then decide frame-by-frame by grouping the labels given by those multiple channels.And then do the re-segmentation step. The second way to deal with this problem is proposed in [6] to find out the most clear or say most dominant signal from these multiple channels and consider this as the single channel ignoring the rests. The most dominant signal can be the one with the highest signal to noise ratio. But, other than this, one well known approach is Acoustic Beamforming. Acoustic beamforming means that one may derive a single channel and then proceed accordingly. In [7], this is done through weighted some of those microphones channels used in meeting room. The weights-estimation associated with this sum is done according to signal quality measure based on SNR(signal-to-noise-ratio). Other proposed algorithms involve automatically selecting the reference channel, computing the channel delays existing between various microphone channels of the meeting room, selecting the optimum delay, and weight assignment to reduce the penalty values associated with negative impact of channels,two-pass Viterbi decoding for smoothing spurious TDOA values. For the researchers who are not very comfortable with signal-array processing, a toolkit is available called BeamformIt.

2.1.2 Feature Extraction

For every signal processing application, feature extraction performs the significant task.

For speech and speaker recognition domains, the extracted features conveys the information about speaker's characteristics and in turn speaker's identity. This can help in identifying speakers correctly, authenticating speakers, verifying speakers, etc.

Which features should be used? The answer depends on the system one is building, the type and amount of data available, resources available, etc. Features can be categorized into various categories like, *Short-term spectral features, voice source*

features, spectro-temporal feaatures, prosodic features and high-level features. Short-term spectral and voice-source features generally gives better performance because they provide information about speaker's physiology depending upon size of the vocal folds, length and dimention of the vocal-tract and are easy to extract. These features do not depend on the phonetic content. Pitch, rhythm, duration, temporal feaures are the prosodic and spectro-temporal features. Pronounciation, accent, idiolect, are the high-level features. They depends upon pesonality type, parental influence, etc. The long-term features are more robust but difficult to extract.

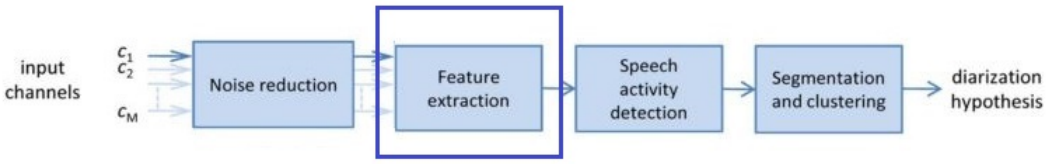

Figure 2.3: Feature Extraction step in Speaker Diarization[3]

This speaker-specific information will help in separating speaker segments from different speakers. How many features should be used that also depends on which algorithm one is going to use to model the speaker. One should keep in mind that not all algorithms can accommodate high-dimensional data. Moreover, Features should have certain characteristics such that they can give the desired information about speaker and one can achieve optimum results.

It is preferred that features should have properties listed below [1]:

- Have large between-speaker variability and small within-speaker variability.
- Be robust against noise and distortion.
- Occur frequently and naturally in speech.
- Be easy to measure from speech signal.
- The number of features should be relatively low.

From speech signals two types of features can be extracted.

a. Short-term cepstral features:

They are considered more accurate as they fetch the physiological information of the speakers. They generally behave like our cochlea and process the information generated by the audio file(vocal tract). They are more accurate. They do not depend on what a person is speaking. They only concerns the voice characteristics of speakers. Such features are Mel-frequency cepstral coefficients (MFCCs), Linear Predictive Coding (LPC) , Linear Predictive Cepstral Coefficients (LPCC) , Perceptual Linear Predictive Cepstral Coefficients (PLPCC) , zero-crossing rate (ZCR), etc.

b. Long-term prosodic features:

These features are high-level features. They possess phonetic information, prosodic information. etc. These features are comparatively difficult to extract. Unlike short-term cepstral features, these features identify characteristics for the longer time period as it contains the general habit of speaking like pronounciation, idiolect, accent, etc. Even disfluensicies like hesitations, repetations are also considered as long term, high level prosodic features.

c. Fusion of Short-term and long-term features

If we take fusion of these two types of features, then we can optimize the feature extraction phase. Because high-level features provides higher robustness to the variation existing in channel/recording. The reason behind this is, these long term features remains quite static for the longer time period and are immune to acoustic changes. So, the accuracy of short-term features and robustness of high-level features can make a better combination for feature extraction phase.

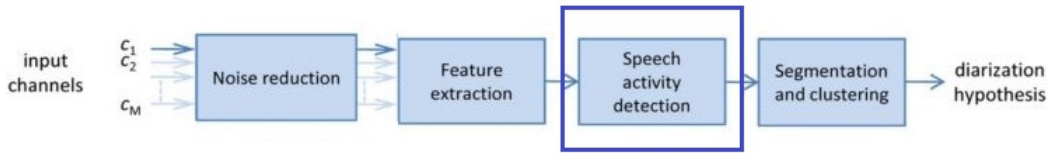

Figure 2.4: VAD step in Speaker Diarization
[3]

2.1.3 Voice Activity Detection

Voice activity detection is also said as Speech activity detection. This is the main part in front-end of whole diarization framework. This activity is followed by segmentation of speech samples. To correctly identifying speech segments, it is necessary that the output of this step is more accurate. The main idea of this step is identifying speech regions. An audio file may contain non-speech parts as silence, music, room noise, door slam, laugh, etc. They all needs to be identified prior to the actual diarization process. Silence can be eliminated using energy level of the speech.

2.1.4 Segmentation

Segmentation is the task of breaking the audio signals such that each segment has only one speaker in it. They are called homogeneous speakers. This is one of the two fundamental tasks(Segmentation & clustering) in Speaker Diarization. Segmentation detects the points where speakers are changing. Now, speakers are identified according to models generated for them. Now, as segmentation senses the speaker change points, a window is used to traverse through the file and find those changes. Two hypothesis are built, one is, the consecutive segments matches the same speaker models, while the 2nd inverse is, the consecutive two segments follow the two different speaker models. These are implemented using some threshold values to satisfy one of those hypothesis. One should make it clear that each segment should consist of only one speaker, otherwise this may lead to higher error rate. The aim of the segmentation step is to identify such points where the acoustic changes occurs. Segmentation and clustering steps combined forms speaker diarization system. The

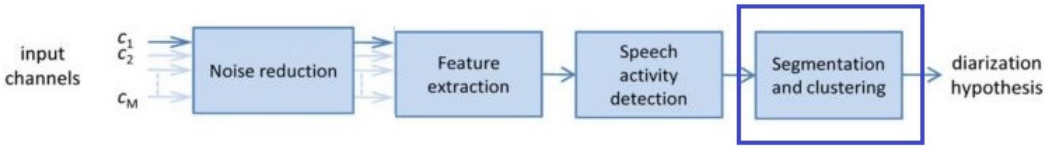

Figure 2.5: Segmentation step in Speaker Diarization [3]

errors in segmentation step will be forwarded to clustering step and so, if the performance of segmentation step is degraded, the performance of the clustering step will degrade consequently. So, to obtain better results one needs to do segmentation with minimal errors.

For speaker diarization evaluation, the mainly used measure is Diarization Error Rate(DER). DER is computed using the number of Miss, False Alarm and Confusion. Where,

- **Miss:** Speech labeled as non-speech (Speech is missed).
- **False alarm:** Non-speech labeled as speech (Noise regions are classified as speech).
- **Confusion:** One speaker's speech is labeled as different speaker's speech.

Segmentation step needs to take care that number of these erroneous segments should be as low as possible.

Generally, two approaches are used for speaker segmentation : one is single pass speaker change detection and second is multiple pass re-segmentation steps. One typical and more general approach for speaker diarization is to use two windows and comparing speaker models for both windows (X and Y). The segmentation output is decided using two hypothesis. One states that both windows have the same acoustic information(Z) and hence belong to same speaker and one speaker segment model can represent them. While the second one is inverse of this, and states that both the windows have different acoustic information and so represent two different speakers

and hence two speaker segment models(Z1 and Z2) are build. The comparison is done using distance measures/thresholds.

From the known segmentation algorithms used, Bayesian information criterion is the most popular.Bayesian information criterion (BIC) and its associated delta-BIC metric [8], the modified BIC criterion [8], cross-BIC and introduced in [9] and, different techniques for likelihood normalization are presented in [10]. Alternative for BIC & its variations is generalized likelihood ratio (GLR) [11]. There is one approach called KullbackLeibler(KL) divergence [12].

2.1.5 Speaker Clustering

Speaker clustering refers to merging of segments belonging to same speaker. Hierarchical clustering algorithms are applied, which forms clusters for speakers. This phase starts instantly after segmentation phase and rests when all the segments from a single speaker is merged altogether. Clustering needs to merge all the segments from same speaker irrespective of their location in the audio file. The ideal result will have one cluster for each speaker. Here, all the distance-measures mentioned above can also be used to check for the similarities between segments, too.

For evaluating clustering step, we have Average Cluster Purity(ACP) & Average Speaker Purity(ASP). ACP degrades when segments from two different speakers are merged as a single speaker. And, when the segment belonging to single speaker is split between more than one clusters, ASP degrades.

Errors occurred in clustering are more sensitive than errors occurred during specifying segment boundaries. Which means, merging two segments from two different speakers, or leaving two different clusters for the same single speaker are more erroneous to the whole diarization system.

2.1.6 Two clustering Approaches

As already mentioned above, for speaker diarization applications, the number of speakers present in the audio file is not known in advance. The output of the seg-

mentation phase is various homogeneous speech segments belonging to single speaker, located anywhere in the whole audio file.And the task of clustering phase is to merge all the segments belonging to the same speakers into one cluster. The segments belonging to different speakers are given to two different speakers. So, to start forming clusters there has to be some hypothesis, and an initial number of clusters, each cluster representing single speaker. For such type of merge/split task two approaches are used as follows.

Top-Down Clustering Approach

This approach starts with the lower number of speaker-clusters and end with the larger number of speaker-clusters. At starting point, some hypothetical number of clusters are made. Then, while traversing the whole file, whenever the speaker change is detected, the new cluster is made for further speakers. This approach is more expensive, as we need to compare all the possible splitting points in the audio file.

Bottom-up Clustering Approach

In contrast to above described approach, this approach starts with larger number of speaker-clusters. And, gradually reduces the number of speaker-clusters, by merging the two clusters belonging to the same single speaker. And, the algorithm stops mering the clusters, when there are no more clusters left to be merged.

Comparison of top-down and Bottom-up approach

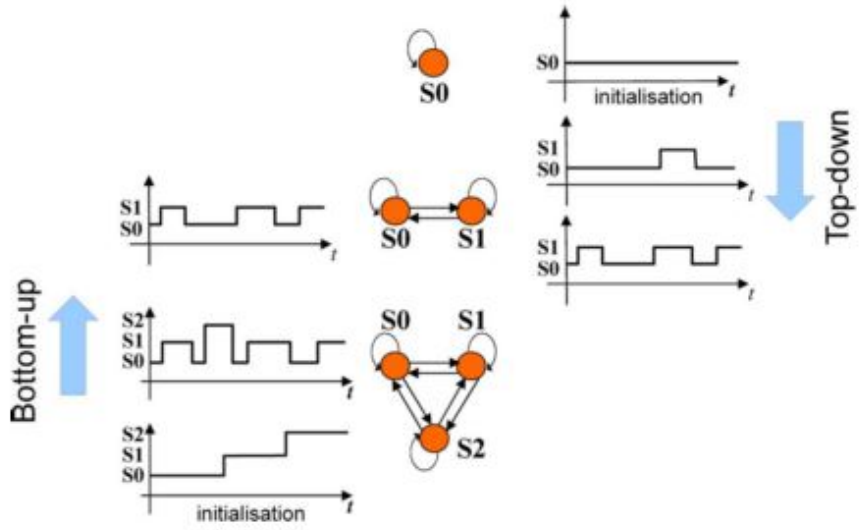

Figure 2.6: Comparison of Top-down and Bottom-up approach [3]

2.2 Different approaches for Speaker Diarization

As mentioned earlier, segmentation and clustering needs to be more accurate to improve efficiency of the system. Otherwise the performance will degrade. Now suppose, there are some errors occurred during segmentation, than those erroneous output will be input for the clustering phase. And then there is no scope of improvements. This situation can be solved if the final output of clustering can be revised. This gives the idea of integrated approach for speaker diarization.

2.2.1 Step-by-Step Approach

In this approach, the flow is strictly followed in the vertical direction. That is, first the segmentation will be completed and then the clustering phase will perform its tasks. There is no way to overcome errors occurred during segmentation or clustering phase. One such state-of the-art algorithm is, CLIPS [13]. Steps to be followed in this approach are mentioned below and depicted in shown in the Fig. 2.7 .

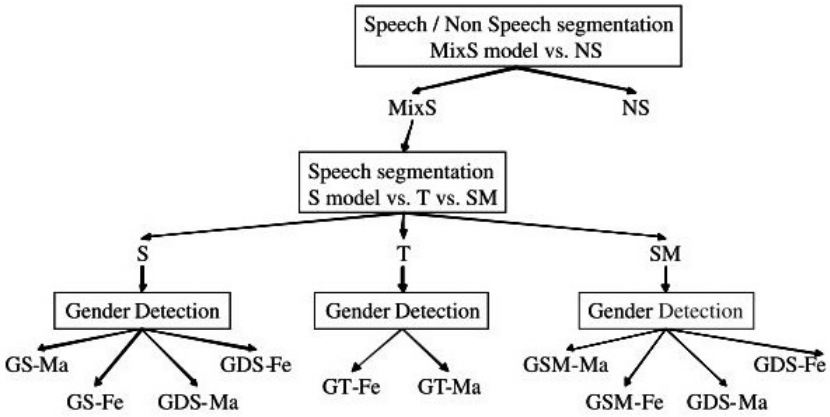

Figure 2.7: Step-by-Step approach for Acoustic Modeling [13]

First, speaker change detection is done. For this task, we use GLR to find the relation between two speech sequences. That is, if we have two acoustic sequences, the using GLR distance measurement, we can infer whether those sequences belong to same speaker or two different speakers. Second step is clustering. Here also the distance measure used is GLR. And lastly, Third step is estimating number of speakers.

2.2.2 Integrated Approach

The improvised and better one is, integrated approach. Here, the control goes back to the segmentation phase once again, to adjust the newly adapted models. That means if there has been any errors in the previous step those can be revised and eliminated or reduced. The flow diagram of this type of approach is as shown in Fig. 2.8 . The output of clustering step is again given to the segmentation module for re-segmentation phase. One such algorithm is LIA algorithm [13].

In the initialization step, single speaker will be trained. Then, one-state HMM will be used for segmentation. In second step, new speaker is identified. And so, the newly identified speaker is eliminated from the initial model, and the newer one is an adapted one, second model is built for that. Then, this same thing will happen for the next newly adapted speakers. This is called as adapting the speaker changes in

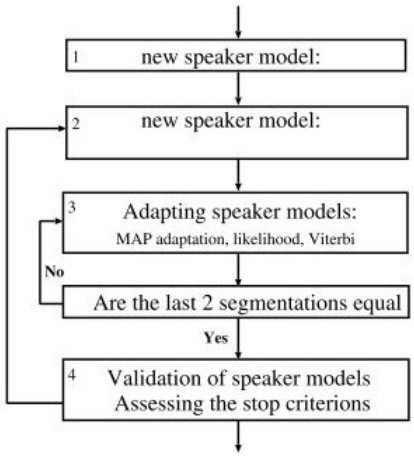

Figure 2.8: Integrated approach for Acoustic Modeling [13]

the file. And, lastly the model validation will be done. Stopping criterion is assessed then. The algorithm stops, when the stopping criterion is reached, or when there is no more speech segment present to be given to the new speaker.

2.2.3 Fusion of step-by-step and Integrated approaches

It is advised that the initial segmentation is done more accurately. But still, the better way is the fusion of both above described approaches. It is observed that, the results obtained using a fusion of both systems where LIA, the integrated approach is applied after the CLIPS step-by-step segmentation system gives the better results compared to the individual approaches. The fusion system obtained a speaker diarization error rate of 12.9% against 19.3% for the CLIPS system used on its own. The fusion system also showed a relative 33% error-reduction compared to the performance of the integrated system taken alone (from 16.9% to 12.9%) [13]

2.3 Speaker Diarization: From Broadcast news to Meetings

Speaker Diarization is applied to mainly three areas, Broadcast news, Telephone conversations, Meeting-room conversations(the most recent one).

Comparison of the three domains

Broadcast news	Telephony conversations	Meeting speech
Number of speakers could be 10 persons or more	Number of speakers limited to 2 or at most 3 persons	The number of speakers is limited to the capacity of the meeting room
Some parts of the file may contain music or commercials	Music and other audio contents do not exist	music or commercials does not exist
The recording condition of each speaker may vary	Recording channel and environment do not usually change	Variations in recording quality, including impulse noises, reverberation and variable speech levels may exist
The recording channel and environment may be different for each speaker	The recording channel and environment are different for each speaker	All the conversations take place in one place
Average speaker change duration is longer	Average speaker change duration is usually so short	Average speaker change duration may be short
Normally there is no overlapping regions between speaker utterances	Normal existence of overlapping regions where two or more speakers speak simultaneously	Normally there are overlapping regions between the speech of two speaker

Table I: Comparison of Broadcast, telephone and meeting domain[1]

2.3.1 Speaker Diarization in Meeting Room Conversations

The presence of overlapped, or co-channel, speech in meetings is a common occurrence and a natural consequence of the spontaneous multiparty conversations which arise within these meetings. This speech, in addition, presents a significant challenge to automatic systems that process audio data from meetings, such as speech recognition and speaker diarization systems. In the case of speaker diarization, current state-of-the-art systems assign speech segments to only one speaker, thus incurring missed speech errors in regions where more than one speaker is active.

The two different situation for meeting rooms can be

- single distant microphones
- multiple distant microphones

Feature extraction need to be modified according to the recording conditions. For multiple distant microphones it will be helpful to use TDOA features.

As with any detection scheme, the overlap system is susceptible to errors of two types: false alarms and misses. These errors impact the diarization system quite differently, with false alarms carrying through to increase the diarization false alarm error and misses having no effect on the baseline diarization error. Because of this difference, the overlap detector is optimized for low false alarms, which corresponds to a high precision (and possibly low recall) operating point."

2.4 Evaluation measures

2.4.1 Segmentation evaluation measures

The widely known segmentation evaluation measures are [1]:

- Recall: percentage of truly detected speaker boundaries (RCL).
- Precision: percentage of candidate speaker boundaries which are the actual speaker change points (PRC).

2.4.2 Clustering evaluation measures

For clustering, it is important to ensure that each cluster has only one speaker assigned and a speaker can be only in single cluster. The evaluation measures for clustering are:

- ACP:
  Average Cluster Purity(ACP) reduces when a cluster includes segments from two or more speakers

- ASP:
  Average Speaker Purity (ASP) reduces when speech of a single speaker is split to more than one cluster

2.4.3 Evaluation measure for Speaker Diarization System

The most common evaluation measure is DER, i.e. Diarization Error Rate. This is computed by obtaining Miss, False Alarm, and Confusion. Here, miss means speech segment is labeled as non-speech. False alarm is reverse, that means, labeling non speech as speech region. And lastly the confusion, this is because the system gets confused between two speakers' speeches.

$$DER = \frac{Miss + False\_alarm + Confusion}{Total\_reference\_speech} \tag{2.1}$$

2.4.4 Datasets

Some widely used and available datasets for Speaker diarization system(dpending on the domain) evaluations are as follows:

a. Broadcast news database

   - Hub-4 1996
   - Hub-4 1997

- Mandarin Broadcast News Speech Corpus (Hub4-NE) 1997
- ESTER SD benchmark(French) GALE Mandarin dataset

b. Telephone Conversation

- Conversational Telephone Speech (CTS)

c. Meeting speech database

- ICSI Meetings Corpus
- CMU Meeting Corpus
- NIST Pilot meeting corpus
- CHIL meeting corpus(for MDM)
- AMI meeting speech corpus

Chapter 3

Implementation

3.1 Algorithm

As discussed earlier, the main phases to be implemented for this systems are,Acoustic beam-forming, feature extraction, voice activity detection, removal of non-speech regions, segmentation and clustering for obtaining homogeneous segments. For each phase, particular algorithmic techniques are implemented with some modifications, according to the intended system.

Dataset used is AMI meeting corpus. This is built for Speaker Diarization and Rich Transcription evaluations. Various scenario based and non-scenario based recorded meeting speech audio files are provided. Each meeting folder consists of total 16 files. Files are recorded with 2 microphone arrays and each array has 8 microphones. So, each file corresponds to each microphone.

Generally we need to do acoustic beamforming to eliminate the delay between microphones and reverberation present in the meeting rooms. But, for this dataset, one mixed and dominant file is already given on which the diarization can be performed and gives better results. So, the beamforming step is eliminated due to benefit provided by the dataset.

For feature extraction, MFCC's are used. 13 Dimentional-MFCC are used. MFCC feature extraction technique is the most recent and state-of-the-art technique. It takes

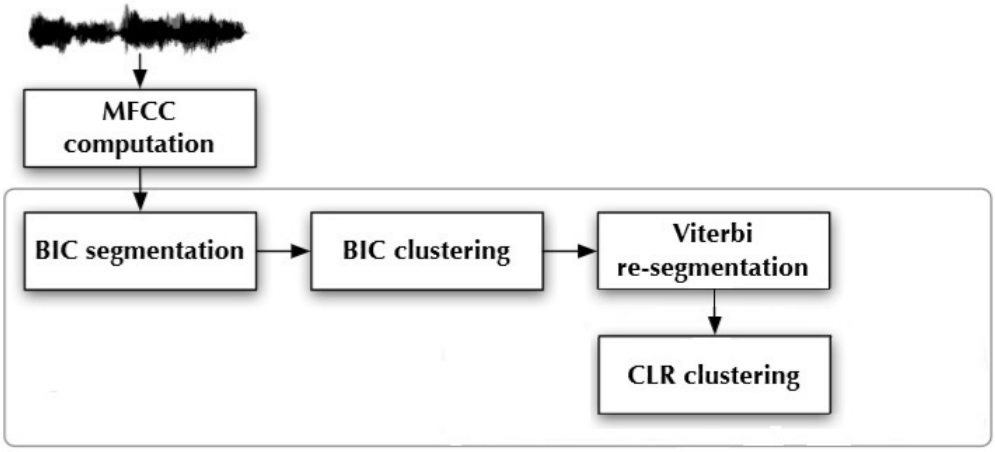

Figure 3.1: Algorithm used by the system

care of all the details related to our listening characteristics. The short frames are generated because sound signal is so much fluctuating over larger time period. Our human cochlea present in our ears, vibrates according to the frequencies in the signal. So, to match this, power spectrum is built. To cop up with two closely related higher frequencies, filterbanks are built. Once this is done, logarithm is applied to overcome the issue related to energy levels. Thes filterbanks are may be overlapping so lastly the DCT is done. MFCCs are computed starting from the beginning to the end of the recorded speech file. Audio features are stored in FeatureSet.

To remove silence, music and jingles, speech-non speech detection is done through Viterbi decoding with 8 one state HMMs.The eight models consist of 2 models of silence (wide and narrow band), 3 models of wide band speech (clean, over noise or over music), 1 model of narrow band speech, 1 model of jingles, and 1 model of music.

Initially only one gaussian is built. Then, the gaussians are getting split till number of hypothetical segments is reached. The gaussians are built using 1.8-5 sec. traversing window. Now, the GMMs are built using EM algorithm. Now, boundaries will be decided for segments. Then, the system will traverse the whole audio file and will detect the instantaneous change points. This will be done by

computing GLR(Generalized Likelihood Ratio). Which means, the change point is detected when we can see that the GLR distance within the window has touched the threshold. Then, the second traversal on the audio file will be done using BIC clustering to merge the subsequent segments those are found to be of the same speaker. A segment object is identified by name of the audio file and starting point and the length of the segment. BIC will behave as a threshold as well as stopping criterion for the following clustering step. Bottom-Up clustering will be applied, which will merge the closest clusters according to the BIC threshod. Cluster is made up of various segments and hence can be seen as container. The sorting of clusters are done according to speaker id.s given to them. And in each cluster the segments are sorted according to their beginning time. Once change points are finalized, viterbi decoder step will help to finalize the segmentation using the GMMs created for the speakers.

3.2 Toolkits available on the web

Various open-source toolkits have been developed and are available on the web for use.

- CMU Segmentation Toolkit, which was developed for diarization of ASR [14].
- AudioSeg, It includes an audio activity detector, BIC/GLR or KL2 segmentation and clustering tools, as well as a Viterbi decoder [15].
- ALIZE, E-HMM algorithm is used for diarization. In this toolkit, Segmentation & clustering are performed in iterartive manner [16].
- SHoUT, which contains a speech/non-speech detector and a diarization tool [17].
- DiarTK, this was published by IDIAP. Which considers the issue of information bottleneck principle [18].

3.3 LIUM Toolkit

LIUM toolkit is programmed in JAVA language. We know that JAVA is a platform independent language. So, This reduces the dependency problems we may face while using different operating systems. It is the extension of mClust, a segmentation tool. The diarization system consists of various elementary tools, like, model trainers, decoders, etc.

3.3.1 Installation

On the site [19] of LIUM Speaker DIarization Toolkit, they have uploaded the source for all the required classes.

The third-party packages used with this toolkit are:

- *gnu-getopt:* package to manage command lines with long options. As this toolkit is handled through the command-line only, one needs to write very long command to specify various options to to the desired task.
- **lapack:** package to manage matrices (blas, f2util and lapack). Only the inversion method based on Cholesky decomposition is used, for inversion of full covariance matrices of Gaussians.
- **Sphinx4 and jsapi:** packages to compute MFCCs.

3.3.2 Usage

To use the toolkit according to your need and the intended application, various changes needs to be made. For this, they provide the facility to write execution command with some more options. The command to run this jar file, made up of so many java files, is:

java -Xmx2024m -jar ./LIUM.jar –fInputMask=./showName.wav –sOutputMask=./showName.seg –doCEClustering showName

where,

- -Xmx2024m is used to specify the memory required to compute this file. This means 2024 MB. 2024 MB memory is enough to compute one hour audio file.
- LIUM.jar is the name of jar file to execute
- –fInputMask=./showName.wav specifies the name of input audi file(wave file).
- –sOutputMask=/showName.seg is used to specify to generate the output file. The ultimate diarization output will be written here.

Some execution snapshots are as follows:

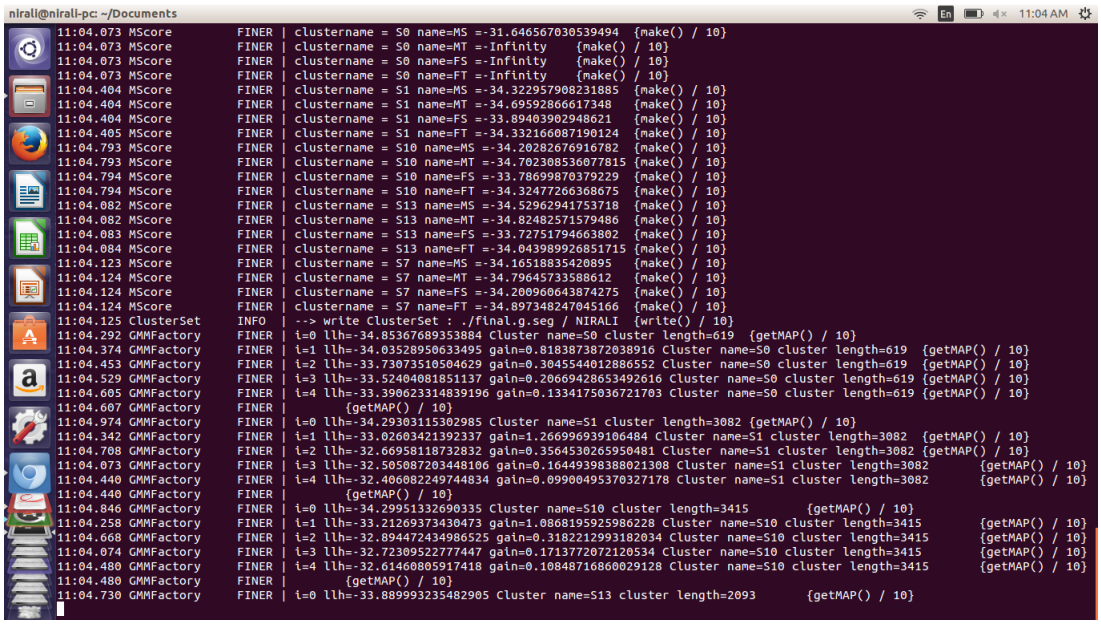

Figure 3.2: Iterations for EM algorithm

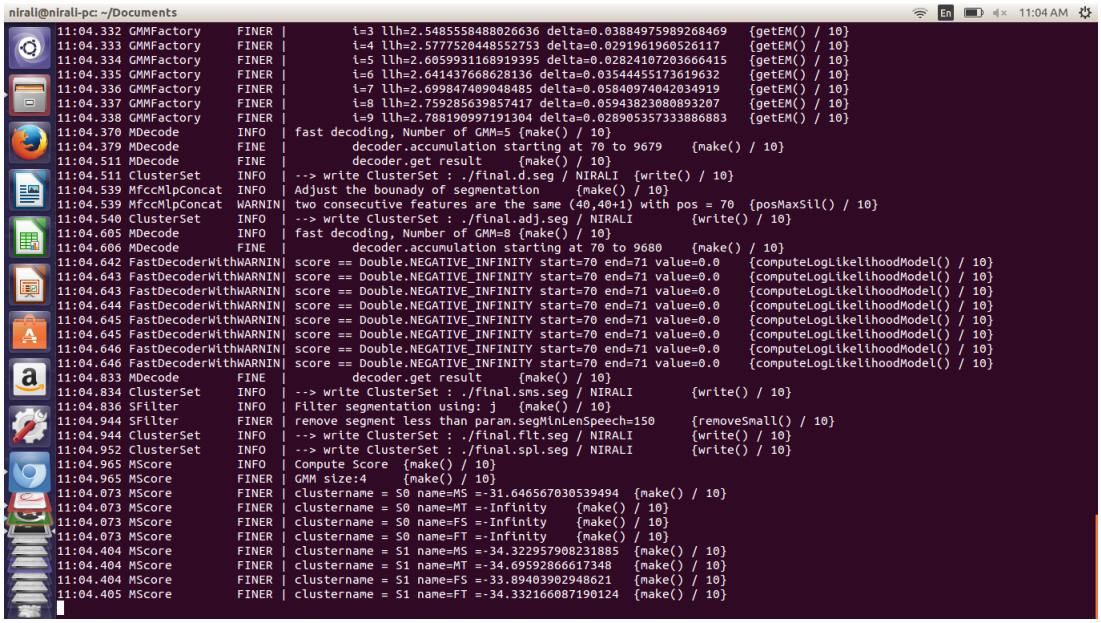

Figure 3.3: Cluster built according to the threshold

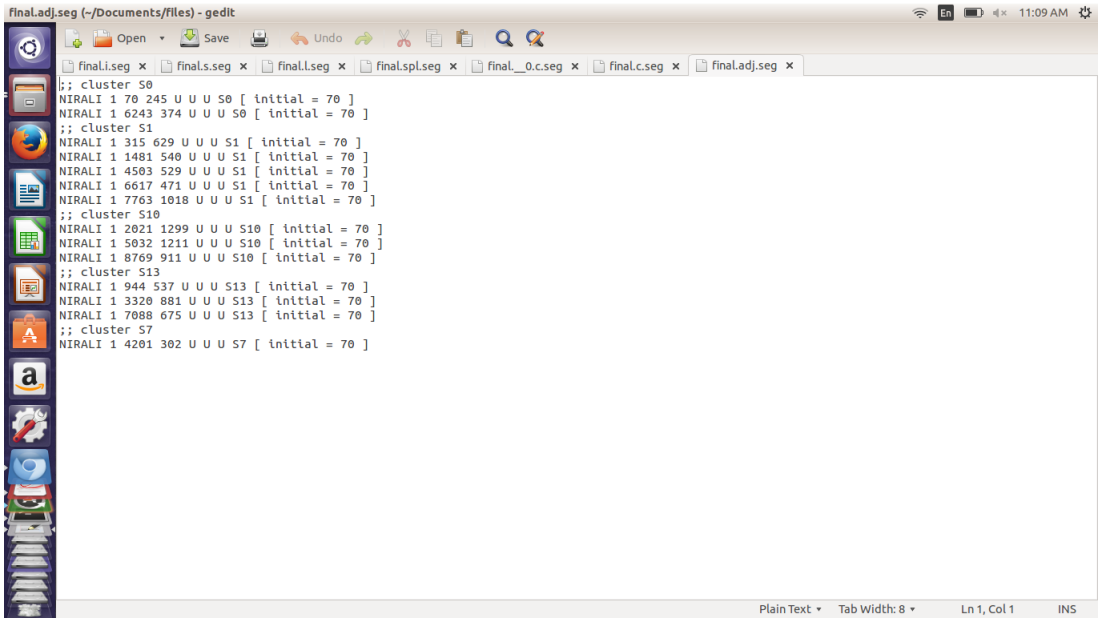

Figure 3.4: Initial Diarization output file

This toolkit is used and modified through the command line only. To modify so many parameters in the file,the original execution command will be followed by –[option1]=value1, –[option2]=value2,... etc. So, the command can get so longer. To

handle this, a third party package, gnu-getopt is included.

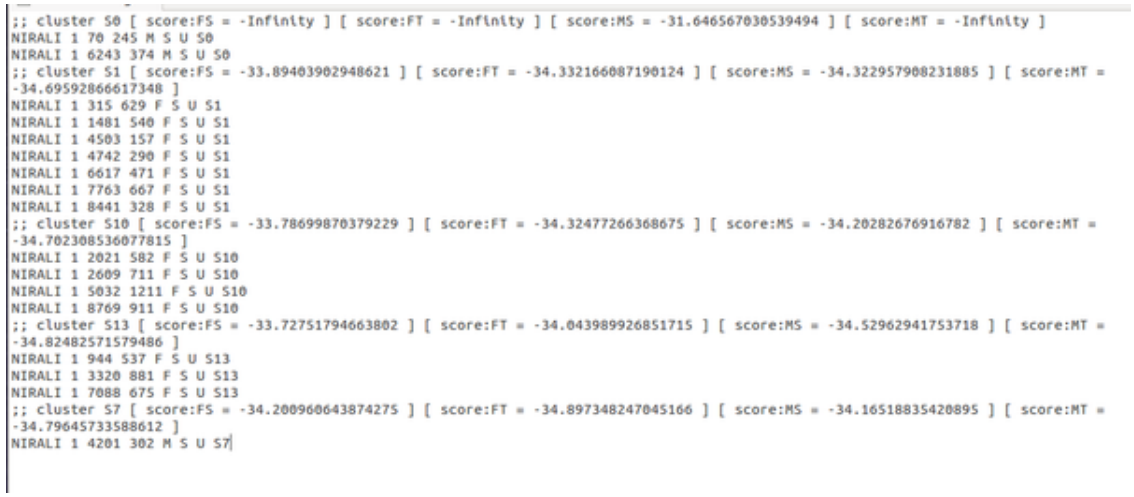

Figure 3.5: Final Output file with clusters made during execution

The output generated in output file can be explained like this :

- field 1: NIRALI = the show name
- field 2: 1 the channel number
- field 3: 1 the start of the segment (in features)
- field 4: 317 the length of the segment (in features)
- field 5: F the speaker gender (U=unknown, F=female, M=Male)
- field 6: S the type of band (T=telephone, S=studio)
- field 7: U the type of environment (music, speech only,)
- field 8: spk0 the speaker label

3.3.3 Modifications according to application

This toolkit is handled by command lines only. And hence, as mentioned above, to modify any options in the programs, we have to modify those options by specifying more options in command. It is little less obvious that one can remember all the

options related to one file and the value-structures. So, the best thing this toolkit provides is the "help" option. This toolkit gives the whole manual of the toolkit as well as individual programs with command as below:

java -cp LIUM_SpkDiarization.jar fr.lium.spkDiarization.programs.programme name -help.

To get help abot the whole JAR file, write:

java -cp LIUM_SpkDiarization.jar –help

```
java -cp "LIUM_SpkDiarization.jar"  fr.lium.spkDiarization.programs.MSegInit --
info[info]      ======================================================
info[program]      name = MSegInit
info[info]      ------------------------------------------------------
info[show]      [options] show
info[ParameterFeature-Input]      --fInputMask      Features input mask = %s.mf
info[ParameterFeature-Input]      --fInputDesc      Features info (type[,s:e:ds
info[ParameterFeature-Input]           type [sphinx,spro4,gztxt,audio2sphinx] =
info[ParameterFeature-Input]           static [0=not present,1=present ,3=to be
info[ParameterFeature-Input]           energy [0,1,3] = 1
info[ParameterFeature-Input]           delta [0,1,2=computed on the fly,3] = 0
info[ParameterFeature-Input]           delta energy [0,1,2=computed on the fly,
info[ParameterFeature-Input]           delta delta [0,1,2,3] = 0
info[ParameterFeature-Input]           delta delta energy [0,1,2,3] = 0
info[ParameterFeature-Input]           file dim = 13
info[ParameterFeature-Input]           normalization, center [0,1] = 0
info[ParameterFeature-Input]           normalization, reduce [0,1] = 0
info[ParameterFeature-Input]           normalization, window size = 0
info[ParameterFeature-Input]           normalization, method [0 (segment), 1 (c
info[info]      ------------------------------------------------------
info[ParameterSegmentationFile-Input]      --sInputMask      segmentation mask
info[ParameterSegmentationFile-Input ([seg,bck,ctl,saus.seg,seg.xml,media.xml],
info[ParameterSegmentationFile-Output]      --sOutputMask      segmentation mas
info[ParameterSegmentationFile-Output ([seg,bck,ctl,saus.seg,seg.xml,media.xml]
info[info]      ------------------------------------------------------
```

Figure 3.6: MSegInit program wih help option

This file shows all the options to be modified in MSegInit file.

3.3.4 Results:

Dataset used is AMI meeting corpus. This is built for Speaker Diarization and Rich Transcription evaluations. Various scenario based and non-scenario based recorded meeting speech audio files are provided. Each meeting folder consists of total 16 files. Files are recorded with 2 microphone arrays and each array has 8 microphones. So, each file corresponds to each microphone.

In this section, The tables contain the results observed during each diarization setup. Using the same algorithmic techniques for each step, it is noticeable that the output can be optimized by modifying the tuning parameters. Using above mentioned algorithmic approach, the best results achieved is 10.43% Diarization Error Rate. The errors occurred during the experiments are classified as Miss, Confusion and False alarm errors. And Diarization Errror Rate is computed using Eq. 2.1.

window size	segment length (in mSec)	Seg. BIC	Clust. BIC	Confusion(in mSec)	False Alarm (in mSec)	Miss (in mSec)	Total Error (in mSec)	DER (%)
350	7000	1	1.5	18009	12207	3840	34056	14.19
			2	17744	12219	3888	33851	14.1
			2.5	17855	12433	3864	34152	14.23
			3	17877	12281	3861	34019	14.17
			3.5	17700	12305	3854	33859	14.11
		1.5	2	17759	12406	3905	34070	14.2
			2.5	17866	12230	3784	33880	14.12
			3	18001	12126	3779	33906	14.13
			3.5	17971	11987	3646	33604	14
		2	2.5	17682	11673	3620	32975	13.74
			3	17668	11574	3538	32780	13.66
			3.5	17649	11479	3507	32635	13.6
		2.5	3	18491	12834	4077	35402	14.75
			3.5	18580	12787	4122	35489	14.79

Table I: DER for window size = 350 frames and 2nd pass Seg. and Clust. threshold values

window size	segment length (in mSec)	Seg. BIC	Clust. BIC	Confusion(in mSec)	False Alarm (in mSec)	Miss (in mSec)	Total Error (in mSec)	DER (%)
300	6000	1	1.5	17783	12737	3758	34278	14.28
			2	17521	12749	3804	34074	14.2
			2.5	17631	12973	3781	34385	14.33
			3	17652	12814	3778	34244	14.27
			3.5	17478	12839	3772	34089	14.2
		1.5	2	17536	12944	3821	34301	14.29
			2.5	17641	12761	3703	34105	14.21
			3	17775	12652	3698	34125	14.22
			3.5	17745	12507	3567	33819	14.09
		2	2.5	17459	12179	3543	33181	13.83
			3	17446	12076	3462	32984	13.74
			3.5	17427	11977	3432	32836	13.68
		2.5	3	18258	13391	3989	35638	14.85
			3.5	18346	13342	4033	35721	14.88

Table II: DER for window size = 300 frames and 2nd pass Seg. and Clust. threshold values

window size	segment length (in mSec)	Seg. BIC	Clust. BIC	Confusion(in mSec)	False Alarm (in mSec)	Miss (in mSec)	Total Error (in mSec)	DER (%)
250	5000	1	1.5	17434	13240	2743	33417	13.92
			2	17177	13253	2777	33207	13.84
			2.5	17285	13485	2760	33530	13.97
			3	17306	13320	2758	33384	13.91
			3.5	17135	13346	2753	33234	13.85
		1.5	2	17192	13455	2789	33436	13.93
			2.5	17295	13265	2703	33263	13.86
			3	17426	13152	2699	33277	13.87
			3.5	17397	13001	2604	33002	13.75
		2	2.5	17117	12660	2586	32363	13.48
			3	17104	12553	2527	32184	13.41
			3.5	17085	12450	2505	32040	13.35
		2.5	3	17900	13920	2912	34732	14.47
			3.5	17986	13869	2944	34799	14.5

Table III: DER for window size = 250 frames and 2nd pass Seg. and Clust. threshold values

window size	segment length (in mSec)	Seg. BIC	Clust. BIC	Confusion(in mSec)	False Alarm (in mSec)	Miss (in mSec)	Total Error (in mSec)	DER (%)
200	4000	1	1.5	16911	12856	2343	32110	13.38
			2	16662	12869	2372	31903	13.29
			2.5	16766	13094	2357	32217	13.42
			3	16787	12934	2355	32076	13.37
			3.5	16621	12959	2351	31931	13.3
		1.5	2	16676	13065	2382	32123	13.38
			2.5	16776	12880	2308	31964	13.32
			3	16903	12771	2305	31979	13.32
			3.5	16875	12624	2224	31723	13.22
		2	2.5	16603	12293	2208	31104	12.96
			3	16591	12189	2158	30938	12.89
			3.5	16572	12089	2139	30800	12.83
		2.5	3	17363	13516	2487	33366	13.9
			3.5	17446	13467	2514	33427	13.93

Table IV: DER for window size = 200 frames and 2nd pass Seg. and Clust. threshold values

window size	segment length (in mSec)	Seg. BIC	Clust. BIC	Confusion(in mSec)	False Alarm (in mSec)	Miss (in mSec)	Total Error (in mSec)	DER (%)
150	3000	1	1.5	14470	11903	2019	28392	11.83
			2	14257	11914	2044	28215	11.76
			2.5	14347	12123	2031	28501	11.88
			3	14364	11975	2030	28369	11.82
			3.5	14222	11998	2026	28246	11.77
		1.5	2	14269	12096	2053	28418	11.84
			2.5	14355	11925	1989	28269	11.78
			3	14464	11824	1986	28274	11.78
			3.5	14440	11688	1917	28045	11.69
		2	2.5	14207	11381	1903	27491	11.45
			3	14196	11285	1860	27341	11.39
			3.5	14181	11193	1844	27218	11.34
		2.5	3	14857	12514	2143	29514	12.3
			3.5	14928	12468	2167	29563	12.32

Table V: DER for window size = 150 frames and 2nd pass Seg. and Clust. threshold values

window size	segment length (in mSec)	Seg. BIC	Clust. BIC	Confusion(in mSec)	False Alarm (in mSec)	Miss (in mSec)	Total Error (in mSec)	DER (%)
100	2000	1	1.5	13076	11254	1786	26116	10.88
			2	12883	11265	1808	25956	10.82
			2.5	12964	11462	1797	26223	10.93
			3	12980	11322	1795	26097	10.87
			3.5	12851	11344	1792	25987	10.83
		1.5	2	12894	11437	1816	26147	10.89
			2.5	12971	11275	1760	26006	10.84
			3	13070	11179	1757	26006	10.84
			3.5	13048	11051	1695	25794	10.75
		2	2.5	12838	10761	1683	25282	10.53
			3	12828	10670	1645	25143	10.48
			3.5	12814	10583	1631	25028	**10.43**
		2.5	3	13425	11832	1896	27153	11.31
			3.5	13490	11789	1917	27196	11.33

Table VI: DER for window size = 100 frames and 2nd pass Seg. and Clust. threshold values

From the outputs obtained above, certain observations can be made.One major observation is made that the window size greatly affects the output even using the same algorithm. This can be justified as follows: Window size specifies the minimum length possible for one segment.

Case I : If window size is too large(suppose X msec), means the minimum segment length is large, then for that region the acoustic features will be used to model the speaker. Now, when we try to build the model based on these features, there might be the cases where there is very small part of speech (suppose x msec) is present in between. But while building the model because of the maximum part of the region is a non-speech portion, the entire region characteristics will be modeled as non-speech. And that will cause the *Miss* scenario. And, in turn will increase the DER(%) and the performance will be degraded.

Case II : If window size is too small then it may be possible that proper features are not extracted. And as we build the models according to features, model building will

window size	segment length (in mSec)	Seg. BIC	Clust. BIC	Confusion(in mSec)	False Alarm (in mSec)	Miss (in mSec)	Total Error (in mSec)	DER (%)
50	1000	1	1.5	22281	13359	3141	38781	16.16
			2	21952	13372	3180	38504	16.04
			2.5	22090	13606	3160	38856	16.19
			3	22117	13440	3158	38715	16.13
			3.5	21899	13466	3152	38517	16.05
		1.5	2	21971	13576	3193	38740	16.14
			2.5	22103	13384	3095	38582	16.08
			3	22270	13270	3090	38630	16.1
			3.5	22233	13118	2982	38333	15.97
		2	2.5	21876	12774	2961	37611	15.67
			3	21859	12666	2893	37418	15.59
			3.5	21835	12562	2868	37265	15.53
		2.5	3	22876	14045	3334	40255	16.77
			3.5	22986	13994	3371	40351	16.81

Table VII: DER for window size = 50 frames and 2nd pass Seg. and Clust. threshold values

be erroneous !!! This might increase the *Confusion* between speakers. And, in turn will increase the DER(%) and the performance will be degraded.

Both the cases depends on the intended application and the data used.

Chapter 4

Conclusion and Future work

The automatic speaker diarization will help storing the information efficiently. Each domain has its own characteristics and needs to be handled accordingly. In meeting domain, there may b the chances of simultaneous overlapping speech utterances. So, one should make sure such regions are not considered as non-speech, else the Miss error would be higher. Secondly,In meeting the number of participants will be higher compared to the Broadcast and telephones domain. So, the other thing needs to be taken care of is Confusion error should not be too large otherwise it will increase the DER drastically. With this experimental setup the optimal results achieved is 10.43% DER, which is considerable improvement for meeting domain.

Now, along with this, a speech detection module can be attached and rich transcription can be done. Moreover, scenario based meetings like, meeting for voting pole, the speech models can be built for each possible words that a speaker possibly speak(like yes, no, agree, etc.). Now, after performing speaker diarization and speech processing for these type of meeting recording, the speaker's answers can be easily extracted and we can easily find out what particular speaker thinks about the scenario. This can be the furthur extension for this approach.

One more thing can be done is the semi-supervised learning, for cross-show diarization, if the candidate speakers are already known.

References

[1] M. H. Moattar and M. M. Homayounpour, "A review on speaker diarization systems and approaches," *Speech Communication*, vol. 54, no. 10, pp. 1065–1103, 2012.

[2] M. T. Knox, "Speaker diarization: Current limitations and new directions," 2013.

[3] N. Evans, S. Bozonnet, D. Wang, C. Fredouille, and R. Troncy, "A comparative study of bottom-up and top-down approaches to speaker diarization," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 2, pp. 382–392, 2012.

[4] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series*, vol. 2. MIT press Cambridge, MA, 1949.

[5] C. Fredouille, D. Moraru, S. Meignier, L. Besacier, and J.-F. Bonastre, "The nist 2004 spring rich transcription evaluation: Two-axis merging strategy in the context of multiple distant microphone based meeting speaker segmentation," in *NIST 2004 Spring Rich Transcrition Evaluation Workshop*, 2004.

[6] Q. Jin and T. Schultz, "Speaker segmentation and clustering in meetings.," in *INTERSPEECH*, vol. 4, pp. 597–600, 2004.

[7] D. Istrate, C. Fredouille, S. Meignier, L. Besacier, and J. F. Bonastre, "Nist rt05s evaluation: pre-processing techniques and speaker diarization on multiple

microphone meetings," in *Machine Learning for Multimodal Interaction*, pp. 428–439, Springer, 2005.

[8] S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, vol. 8, Virginia, USA, 1998.

[9] X. Anguera and J. Hernando, "Evolutive speaker segmentation using a repository system," in *Proc. Interspeech*, vol. 21, Citeseer, 2004.

[10] A. Malegaonkar, A. Ariyaeeinia, P. Sivakumaran, and J. Fortuna, "Unsupervised speaker change detection using probabilistic pattern matching," *Signal Processing Letters, IEEE*, vol. 13, no. 8, pp. 509–512, 2006.

[11] H. Gish, M.-H. Siu, and R. Rohlicek, "Segregation of speakers for speech recognition and speaker identification," in *icassp*, pp. 873–876, IEEE, 1991.

[12] M. A. Siegler, U. Jain, B. Raj, and R. M. Stern, "Automatic segmentation, classification and clustering of broadcast news audio," in *Proc. DARPA speech recognition workshop*, vol. 1997, 1997.

[13] S. Meignier, D. Moraru, C. Fredouille, J.-F. Bonastre, and L. Besacier, "Step-by-step and integrated approaches in broadcast news speaker diarization," *Computer Speech & Language*, vol. 20, no. 2, pp. 303–330, 2006.

[14] M. A. Siegler, U. Jain, B. Raj, and R. M. Stern, "Automatic segmentation, classification and clustering of broadcast news audio," in *Proc. DARPA speech recognition workshop*, vol. 1997, 1997.

[15] G. Gravier, M. Betser, and M. Ben, "audioseg: Audio segmentation toolkit, release 1.2," *IRISA, january*, 2010.

[16] J.-F. Bonastre, N. Scheffer, D. Matrouf, C. Fredouille, A. Larcher, A. Preti, G. Pouchoulin, N. W. Evans, B. G. Fauve, and J. S. Mason, "Alize/spkdet: a state-of-the-art open source software for speaker recognition.," in *Odyssey*, p. 20, 2008.

[17] M. A. H. Huijbregts, "Segmentation, diarization and speech transcription: surprise data unraveled," 2008.

[18] D. Vijayasenan and F. Valente, "Diartk: An open source toolkit for research in multistream speaker diarization and its application to meetings recordings.," in *INTERSPEECH*, 2012.

[19] "Lium speaker diarization." `http://www-lium.univ-lemans.fr/diarization/doku.php/welcome`.