Speaker Diarization: Current Limitations and New Directions

by

Mary Tai Knox

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Nelson Morgan, Chair
Dr. Gerald Friedland
Dr. N. Nikki Mirghafori
Professor Kannan Ramchandran
Professor David Wessel

Spring 2013

Speaker Diarization: Current Limitations and New Directions

Abstract

Speaker Diarization: Current Limitations and New Directions

by

Mary Tai Knox

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Nelson Morgan, Chair

Speaker diarization is the problem of determining "who spoke when" in an audio recording when the number and identities of the speakers are unknown. Motivated by applications in automatic speech recognition and audio indexing, speaker diarization has been studied extensively over the past decade, and there are currently a wide variety of approaches – including both top-down and bottom-up unsupervised clustering methods. The contributions of this thesis are to provide a unified analysis of the current state-of-the-art, to understand where and why mistakes occur, and to identify directions for improvements.

In the first part of the thesis, we analyze the behavior of six state-of-the-art diarization systems, all evaluated on the National Institute of Standards and Technology (NIST) Rich Transcription 2009 evaluation dataset. While performance is typically assessed in terms of a single number – the diarization error rate (DER) – we further characterize the errors based on speech segment durations and their proximity to speaker change points. It is shown that for all of the systems, performance degrades both as the segment duration decreases and as the proximity to the speaker change point increases. Although short segments are problematic, their overall impact on the DER is small since the majority of scored time occurs in segments greater than 2.5 seconds. By contrast, the amount of time near speaker change points is relatively high, and thus poor performance near these change points contributes significantly to the DER. For example, for the single distant microphone (SDM) and multiple distant microphone (MDM) conditions, over 33% and 40% of the errors occur within 0.5 seconds of a change point for all evaluated systems, respectively.

In the next part of the thesis, we focus on the International Computer Science Institute (ICSI) speaker diarization system and explore the effects of various system modifications. This system contains many steps – including speech activity detection, initialization, speaker segmentation, and speaker clustering. Inspired by our previous analysis, we focus on modifications that improve performance near speaker change points. We first implement an alternative to the minimum duration constraint, which sets the shortest amount of speech time before a speaker change can occur. This modification results in a 12% relative improvement

of the speaker error rate for the MDM condition, with the largest improvement occurring closest to the speaker change point, and a 3% relative improvement for the SDM condition. Next, we show how the difference between the largest and second largest log-likelihood scores provides valuable information for unsupervised clustering, namely it indicates which regions of the output are likely correct.

Lastly, we explore the potential of applying speaker diarization methodologies to other applications. Specifically, we investigate the use of a diarization-based algorithm for the problem of duplication detection, where the goal is to determine whether a given query (e.g., a short audio clip) has been taken from a reference set (e.g., a large collection of copyrighted media). With minimal modifications of the ICSI diarization system, we are able to obtain moderate performance. However, our approach is not competitive with existing approaches designed specifically for the problem of duplication detection, and the extent to which diarization-based approaches are useful for this application remains an open question.

Contents

List of Figures

List of Tables

Acknowledgments

I have been surrounded many supportive people during my doctoral journey, to whom I owe a great deal of thanks. First, I would like to thank my three research advisors: my head advisor Nelson Morgan, and my research mentors Gerald Friedland and Nikki Mirghafori. Morgan has provided thoughtful and relevant advice throughout the years, for which I am extremely grateful. Gerald has been a wonderful mentor, with great enthusiasm for research and life in general. His energy is contagious and made for some rousing meetings. And finally, words cannot express what working with Nikki has meant to me; her experience and technical advice where invaluable to my research, but even more importantly, her faith and confidence in me helped me to believe in myself when I needed it most.

My time at ICSI has been enriched greatly by the many wonderful people who have helped me along the way; it seems impossible to list everyone... but I will try! Adam Janin, Andreas Stolcke, Steven Wegmann, Chuck Wooters, Michael Ellsworth, and Liz Shriberg have all provided useful feedback on my work. Also, the students, visitors, post-docs, and staff have made ICSI a great place to work (from research discussions to foosball, what more could once ask for!?!). These include Luke Gottlieb, Jacob Wolkenhauer, David Imseng, Korbinian Riedhammer, Marios Athineos, Hari Parthasarathi, Shuo-Yiin Chang, Howard Lei, Jaeyoung Choi, Arlo Faria, TJ Tsai, Oriol Vinyals, Ben Elizalde, Kofi Boakye, and Dan Gillick. I would also like to give special thanks to my officemates Suman Ravuri and Lara Stoll for all of the good memories in 531.

I also want to acknowledge my friends outside of ICSI for keeping me sane. In particular, I would like to give a shout out to all of the many residents at the Copa Colusa. Without you all, graduate school would not have been nearly as enjoyable or active (but maybe I would have finished sooner ;)).

I owe a great deal of gratitude to my family. In particular, to my parents and siblings who have always supported my goals, and my nieces and nephew who always remind me of what is most important in life.

Finally, I would like to thank Galen for his love and support. He has opened my eyes to a beautiful life full of research (and climbing).

Chapter 1

Introduction

1.1 Motivation

Imagine sitting in a research group meeting. As others are discussing their related work, you stop to write down a brilliant idea that will solve the current problem you are working on. After writing, you realize you have missed the last two minutes of your research advisor talking. Fortunately, you remember that the entire meeting is being recorded on your cell phone and therefore it is not necessary to interrupt the meeting. After the meeting, you want to quickly replay the portion of the meeting you missed. Instead of listening to the entire meeting, you recall you have recently installed a speaker diarization application on your phone which automatically annotates an audio recording by speaker. Now you can quickly sift through the utterances your advisor spoke and listen to the two minutes you missed. Problem solved. Speaker diarization for the win!

Speaker diarization is the problem of automatically partitioning an unprocessed audio recording into speaker homogenous regions, answering the question, "Who spoke when?" Speaker diarization is performed without prior knowledge of the speech/nonspeech regions, number of speakers, or speaker identities; therefore, there are no pre-trained models for the individual speakers in the recording. Speaker diarization is a long-standing problem within the speech community, along with automatic speech recognition and speaker verification. Although speaker diarization and speaker verification are closely related (both tasks involve distinguishing between speakers), the major difference is that in the speaker verification setting prior models are trained on the target speakers. Whereas for speaker diarization, there is no prior information regarding any of the speakers in the recording.

Speaker diarization applications include speaker adaption for automatic speech recognition [53], audio indexing [25, 43], and speaker localization [27]. Furthermore, the information provided by a speaker diarization system can be used to analyze behavior, including participant roles such as the dominant speaker [39].

Over the last decade, the National Institute of Standards and Technology (NIST) has

held eight Rich Transcription (RT) evaluations [48]. The RT evaluations encourage research in several automatic speech technologies, including automatic speech recognition and speaker diarization, and provide common datasets for evaluation of performance. The ultimate goal of the RT evaluation is to utilize these automatic systems to create more informative and useful transcriptions of recordings. A number of approaches to speaker diarization have been introduced and evaluated on RT datasets: from bottom-up and top-down Hidden Markov Model - Gaussian Mixture Model (HMM-GMM) approaches to Information Bottleneck approaches to Hierarchical Dirichlet Process - Hidden Markov Model (HDP-HMM) approaches.

Despite recent progress, performance of state-of-the-art diarization systems is still highly variable. For example, for the NIST RT 2009 evaluation, the median scores for the multiple distant microphone condition ranged from 9% Diarization Error Rate (DER) for the best scoring recording to 54% DER for the worst scoring recording. Thus, to proceed, there is a need to understand why systems work well and why systems work poorly.

1.2 Contributions

One of the main contributions of this thesis is an analysis of six state-of-the-art speaker diarization systems on the NIST RT 2009 evaluation dataset. Typically systems are compared based solely on the overall DER. However, in this work, with the collaboration of other speaker diarization researchers, we obtain the final speaker diarization outputs of six state-of-the-art speaker diarization systems. Using these final diarization outputs, we are able to determine where the systems work well, and more importantly where they struggle.

A second contribution of this thesis is an in-depth study of the performance of the speaker diarization system developed at the International Computer Science Institute (ICSI). In the analysis over multiple systems, we only study the final speaker diarization output. However, in the study of the ICSI speaker diarization system we further investigate the speaker models and change various parameters to determine how these settings affect speaker diarization results. Through this analysis we identify where improvements can be made. The ICSI speaker diarization algorithm is then modified to improve the results, particularly in areas that were found to be troublesome: near speaker change points.

A further contribution of this thesis is to investigate how methodologies developed in the context of speaker diarization can be used for other applications. In this work, we consider the problem of detecting duplicate audio clips.

1.3 Organization

This thesis is outlined as follows: Chapter 2 provides relevant background information on speaker diarization; Chapter 3 describes the analysis of multiple speaker diarization systems; Chapter 4 introduces new methodologies to improve speaker diarization and examines the

effect on the Diarization Error Rate (DER); Chapter 5 explores the use of speaker diarization methodologies for the problem of duplication detection; and Chapter 6 summarizes the thesis and outlines future directions.

Chapter 2

Speaker diarization background

As discussed in Chapter 1, this thesis seeks to analyze and improve upon current state-of-the-art speaker diarization systems, as well as explore other applications in which speaker diarization methods are useful. In this chapter, necessary background information regarding speaker diarization is described.

This chapter is organized as follows: Section 2.1 defines the speaker diarization problem; Section 2.2 outlines a number of speaker diarization systems; Section 2.3 describes the data used in this work; and Section 2.4 defines the Diarization Error Rate (DER), the scoring metric utilized throughout this thesis.

2.1 Speaker diarization problem

The goal of speaker diarization is to partition an input audio recording into speaker homogeneous speech regions, as shown in Figure 2.1. The number of speakers, speaker identities, and speech/nonspeech regions are not known a priori. Since the speaker identities are not known, each unique speaker in a given recording is given a generic label (e.g., A, B, C, or D as shown in Figure 2.1).

Speaker diarization is a problem that has been investigated within the speech community over the last decade, over which NIST has held eight Rich Transcription (RT) evaluations [48]. These evaluations give speaker diarization research sites the opportunity to evaluate and compare their systems on unseen datasets. The ultimate purpose of the RT evaluations is to automatically generate informative transcriptions [48]. By including speaker diarization information, the transcript annotates "who said what". Also, it is possible to determine the roles each participant plays in the recording, such as the dominant speaker [39].

There are three main tasks within speaker diarization: speech activity detection, speaker segmentation, and speaker clustering. These tasks are described in further detail in the following paragraphs.

As the name implies, speech activity detection identifies the regions of the recording which

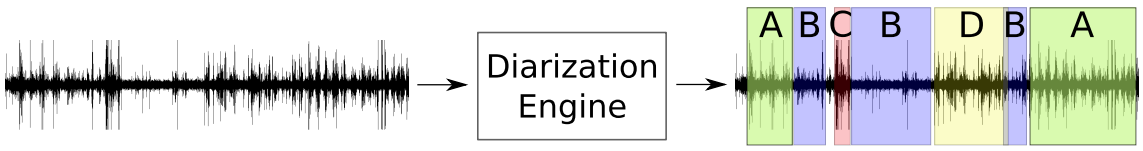

Figure 2.1: Overview of speaker diarization. From an input audio signal, segment the signal into nonspeech and speech segments, the latter labeled by speaker (e.g., A, B, C, D).

contain speech, thereby distinguishing the speech regions from the nonspeech regions (e.g., silence; transient sounds like door slams and mouse clicks; non-lexical speaker-generated noise including laughing, breathing, and coughing). In speaker diarization, speech activity detection is primarily used to ensure speaker models only include data from speech regions. Speech activity detection is often performed using Gaussian Mixture Models (GMMs) trained on speech and nonspeech regions [35, 47].

The speech regions are then split into speaker homogeneous segments either via speaker segmentation or speaker change point detection. Speaker segmentation is the more common method for identifying the beginning and end of each speaker homogeneous segment. Speaker segmentation is performed after creating models for the hypothesized speakers, using Viterbi decoding to label when each hypothesized speaker is speaking. A by-product of Viterbi decoding is the locations of the hypothesized speaker change points.

Another method of partitioning the speech regions into speaker homogeneous segments is to use speaker change point detection. This is typically done by performing hypothesis testing on a sliding window partitioned into two sections. The null hypothesis is there is no change point and the two sections are spoken by the same speaker, and thus only requires a single model. The alternative hypothesis is there is a speaker change at the boundary between the two sections, and therefore the two sections should be modeled separately, necessitating two models. Common metrics used to perform speaker change point detection include the Generalized Likelihood Ratio (GLR) [29] and Bayesian Information Criterion (BIC) [20].

Speaker clustering is the task of grouping speech regions containing the same speaker together. In other words, speech regions spoken by the same speaker are given the same unique label. Speaker clustering is often performed in conjunction with speaker segmentation. Typically, models are first trained based on the hypothesized clustering, where the first grouping (or speaker clustering) is performed according to some initialization procedure. Then based on these models, Viterbi decoding is performed to label when each hypothesized speaker spoke. Therefore, the Viterbi decoding both identifies when the speaker changes occur (or speaker segmentation) as well as determines when each hypothesized speaker speaks again (thereby grouping speech regions containing the same speaker together, or speaker clustering). This procedure is often repeated as follows. The latest segmentation is used to train the hypothesized speaker models. After which, Viterbi decoding is run to perform speaker

segmentation and speaker clustering.

In the bottom-up method of speaker diarization, described further in 2.2.1, a merging step is also used to perform speaker clustering. The bottom-up method initializes with a large number of clusters and iteratively merges clusters until a stopping criterion is met (at which point each cluster represents a hypothesized speaker). Two clusters are merged when the data in each cluster is hypothesized to be from the same speaker. A merging criterion is used to determine which clusters to merge. A number of merging criteria have been explored, including the GLR, BIC, and Kullback-Leibler (KL) divergence [55].

Another (less commonly used) method of speaker clustering is to group predetermined segments together if they contain the same speaker. This method of speaker clustering is often used in speaker diarization systems which perform change point detection. In these systems, change point detection is first used to segment the recording into speaker homogeneous segments and then these segments are grouped together according to a distance measure.

2.2 Speaker diarization systems

A variety of methods have been used to perform speaker diarization. In this section, the most common speaker diarization methods are described.

2.2.1 Hidden Markov Model - Gaussian Mixture Model (HMM-GMM)

The Hidden Markov Model - Gaussian Mixture Model (HMM-GMM) is the most popular approach to speaker diarization. In this approach, an ergodic HMM (where there is a non-zero probability of transitioning to any state from a given state) is used to segment the audio recording. Each HMM state (also referred to as a cluster) represents a hypothesized speaker; therefore, a transition from one state to the next represents a speaker change point. Each state is modeled using a GMM.

There are two methodologies within the HMM-GMM approach to speaker diarization: bottom-up and top-down. The bottom-up method initializes with many clusters and iteratively reduces the number of clusters until the stopping criterion is met, at which point the number of clusters represents the number of hypothesized speakers. The top-down method begins with very few clusters (typically only one) and iteratively increases the number of clusters until the stopping criterion is met. The bottom-up and top-down HMM-GMM speaker diarization algorithms are described in more detail in the following subsections.

Bottom-up

The bottom-up HMM-GMM algorithm is the most popular approach to speaker diarization. In fact, all but one of the systems submitted to the NIST Rich Transcription 2009 evaluation were bottom-up HMM-GMM systems. The other system was a top-down HMM-GMM system.

As previously stated, the bottom-up methodology is initialized with many clusters and iteratively merges clusters together. The process of iteratively reducing the number of clusters is also referred to as agglomerative hierarchical clustering (AHC).

An outline of the bottom-up HMM-GMM approach is shown in Figure 2.2. As a brief overview, the first step is to extract features. Next, speech activity detection is performed; after which the focus of the algorithm is on the speech regions. The speech regions are initially assigned to k clusters, where the number of initial clusters, k, is greater than the anticipated number of speakers. After the initial segmentation, a GMM is trained on the data from each cluster and the recording is re-segmented using the Viterbi form of the Expectation Maximization (EM) algorithm. In the E-step, segmentation is performed such that the likelihood of the features is maximized based on the current GMM parameters. In the M-step, the GMM parameters are updated according to the new segmentation. After updating the models for each of the clusters, the next step is to determine whether or not to merge two clusters. If the merging criterion is met, then the data from two clusters are merged and the re-training/re-segmenting step is performed again. Otherwise, the algorithm concludes and the final segmentation is returned.

There are many variants of bottom-up HMM-GMM speaker diarization systems. The algorithms typically utilize different initialization procedures, merging criterions, and stopping criterions. Some of the most common approaches are described below.

A number of initialization procedures have been examined. Uniform initialization is a popular approach in which the recording is split uniformly into k clusters. Another common initialization method is k-means clustering. Typically, each of the k clusters is modeled using a GMM and the initial means are randomly assigned. These approaches have been shown to yield similar results [5, 10].

A number of different metrics have been used as merging and stopping criterions. Bayesian Information Criterion (BIC) [20, 5] and Kullback-Leibler (KL) divergence [51] based metrics have been used for both the merging and stopping criteria. The ΔBIC calculates the difference between the log-likelihoods of two scenarios: modeling the combined data from two clusters with a single GMM and modeling two clusters separately with two GMMs. More explicitly, the ΔBIC is:

$$\Delta BIC(C_1, C_2) = \log \frac{p(x_{1,2}|\theta_{1,2})}{p(x_1|\theta_1)p(x_2|\theta_2)} - \frac{\lambda}{2} K \log(N) \tag{2.1}$$

where C_1 and C_2 are clusters 1 and 2; x_1 and x_2 are the multi-dimensional data from clusters 1 and 2; $x_{1,2}$ is the multi-dimensional data from x_1 and x_2; θ_1, θ_2, and $\theta_{1,2}$ are the parameters

of clusters 1, 2, and $1 \cup 2$; λ is the penalty weight (a tunable parameter); K is the difference in the number of parameters between $\theta_{1,2}$ and θ_1, θ_2; and N is the number of data points (or frames) in $x_{1,2}$. In order to avoid tuning λ, K is typically set to zero by setting the number of parameters in $\theta_{1,2}$ equal to the sum of the parameters in θ_1 and θ_2 [5]. The KL divergence measures the difference between two distributions (in this case the GMMs trained on data from two clusters) and is defined as:

$$\mathrm{KL}(C_1, C_2) = \int \log\left(\frac{p(x|\theta_1)}{p(x|\theta_2)}\right) p(x|\theta_1)dx, \tag{2.2}$$

where the integral is over the domain of the data x. Since the KL divergence is not symmetric, the symmetric KL divergence is often used to determine which clusters to merge and when to stop merging. The symmetric KL divergence is given by the sum of $\mathrm{KL}(C_1, C_2)$ and $\mathrm{KL}(C_2, C_1)$. Other merging criteria include the Generalized Likelihood Ratio (GLR) [29] and the Information Change Rate (ICR) [33]. The GLR is similar to the BIC and computes the ratio of the likelihood that the data in two clusters are generated by the same speaker to the likelihood that the data in two clusters are generated by two different speakers [29]. The ICR is a measure of the change in entropy (or information) after merging two clusters. If two clusters are from the same speaker (more homogeneous), ideally the change in entropy after merging the clusters is less than if two clusters are from two different speakers (more heterogeneous) [33].

Top-down

The top-down HMM-GMM procedure is similar to the bottom-up HMM-GMM procedure. Both systems first extract features and perform speech activity detection. Unlike the bottom-up system, the top-down system initializes the algorithm with one (or very few) cluster(s) (or state(s)) modeling all of the data in the speech regions of the recording. After initialization, based on the splitting criterion, speech data is designated as belonging to a new cluster (or state) and the clusters are re-trained and the data is re-segmented. The splitting and re-training/re-segmenting steps continue until the stopping criterion is met [24].

2.2.2 Sticky Hierarchical Dirichlet Process - Hidden Markov Model (HDP-HMM)

Speaker diarization is also performed using a Hierarchical Dirichlet Process - Hidden Markov Model (HDP-HMM) based approach. Unlike the HMM-GMM approach, the HDP-HMM approach is non-parametric. This eliminates the need to tune a number of parameters, such as the number of initial clusters k and the number of initial mixtures in the GMMs g. The DPs are used to both define a prior distribution on transition matrices over countably infinite state spaces as well as model the emission probabilities [23]. Like the HMM-GMM system, an HMM is utilized to segment the data. In order to avoid excessive transitioning

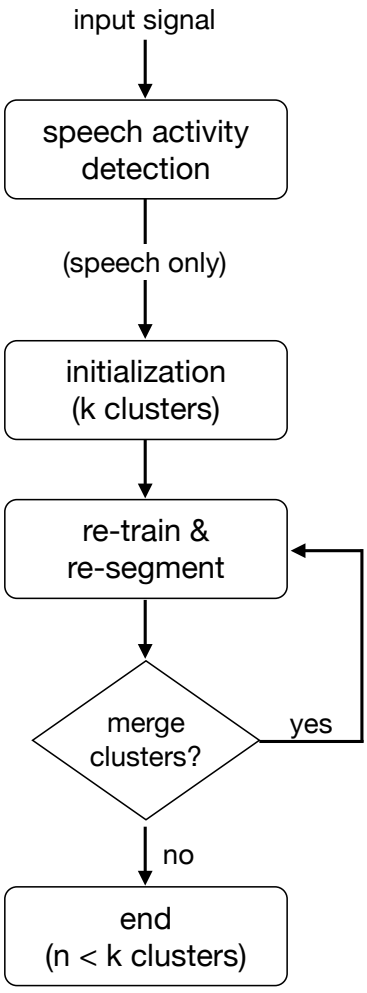

Figure 2.2: Bottom-up Hidden Markov Model-Gaussian Mixture Model (HMM-GMM) speaker diarization system diagram.

between states (or hypothesized speakers), a "sticky" component – which increases the self-transition bias, making it more probable to stay in a given state rather than transition to a new one – is introduced to the HDP-HMM system. The sticky HDP-HMM system performs similar to the bottom-up HMM-GMM system [23]. In practice, the HDP-HMM system is performed via sampling in order to avoid "complicated bookkeeping" [23]. Therefore, multiple runs of the HDP-HMM system on the same dataset often do not lead to the same results.

2.2.3 Agglomerative Information Bottleneck (aIB)

Agglomerative Information Bottleneck (aIB) is another bottom-up algorithm used to perform speaker diarization. The goal of the aIB system is to iteratively merge uniform short segments $x_1, x_2, \cdots, x_t$ into clusters $c_1, c_2, \cdots, c_n$ which simultaneously maximize the mutual information $I(Y, C)$ of a set of relevance variables Y and a set of clusters C, while minimizing the mutual information $I(C, X)$ of C and a set of segments X, as shown in Equation 2.3. The merging continues until the stopping criterion is met. After which, Viterbi decoding is performed in order to determine the segment boundaries.

$$\max[I(Y, C) - \frac{1}{\beta} I(C, X)] \tag{2.3}$$

In this setting X is the set of uniform short segments, Y is a set of components of a background GMM trained on the entire audio recording, and β is a Lagrange multiplier. Thus, Equation 2.3 is used to determine a cluster representation C which is useful for describing the relevance variables Y (maximize $I(Y, C)$) and simple (minimize $I(C, X)$). The aIB is more computationally efficient than the HMM-GMM speaker diarization system since new models are not trained for each potential merging of two clusters. Instead, for the aIB framework subsequent statistics are taken to be averages of previously defined statistics. For example, $p(y|C_{1,2}) \propto p(y|C_1)p(C_1) + p(y|C_2)p(C_2)$, where $C_{1,2}$ represents merging clusters C_1 and C_2 [56].

2.2.4 Factor analysis

Factor analysis based speaker diarization systems are also used to perform speaker diarization [49, 50]. In this approach, the goal is to separate the speaker and channel variabilities to determine the low dimensional identity vector, referred to as the i-vector [21]. This is performed using GMM supervectors, a vector containing the mixture means of a GMM which are stacked to create a column vector. More specifically, the speaker– and channel–dependent supervector M is defined as:

$$M = m + Tw + \epsilon \tag{2.4}$$

where m is the speaker– and channel– independent supervector often computed from a Universal Background Model (UBM); T is a rectangular, low rank matrix of the Total Variability subspace, w is a low-dimensional vector, often referred to as an i-vector; and ϵ is residual noise. The cosine distance is then used to compute the distance between two i-vectors w_1 and w_2 as follows:

$$\cos(w_1, w_2) = \frac{(w_1)^T(w_2)}{||w_1|| \cdot ||w_2||} \tag{2.5}$$

where each i-vector is computed over short segments. In [50], i-vectors are first extracted over short ($\approx$ 1 second) windows and the number of speakers is determined via spectral clustering. Then k-means clustering based on the cosine distance is used to cluster the i-vectors (and their corresponding segments). After clustering, a number of post-processing steps are performed before obtaining the final segmentation.

The factor analysis approach to speaker diarization has been evaluated on Speaker Recognition Evaluation datasets as opposed to the more commonly used Rich Transcription (RT) datasets [49, 50]. Therefore, it is difficult to compare performance between the factor analysis system and state-of-the-art HMM-GMM based speaker diarization systems.

2.3 Data

Speaker diarization is performed on a number of domains, including telephone, broadcast news, lecture, and the meeting domain. Recordings from the meeting domain are often studied within the speaker diarization community. Since 2002, there have been a number of meeting related projects, including the ICSI Meeting project, European Union Multimodal Meeting Manager (M4) project, Swiss Interactive Multimodal Information Management (IM2) project, European Union Augmented Multi-party Interaction (AMI) project, European Union Augmented Multi-party Interaction with Distant Access (AMIDA) project, and European Union Computers in the Human Interaction Loop (CHIL) project [8]. Also, beginning in 2002, NIST has conducted a number of Rich Transcription (RT) Evaluations, which focus on speech-to-text transcription as well as speaker diarization. Initially, the RT evaluations contained recordings from a large number of sources, including meetings, broadcast news, lectures, and conversational telephone speech. However, since 2006, the RT evaluations have only included meeting recordings.

The meeting domain is interesting for a number of reasons. First, the meeting domain contains spontaneous speech, which is representative of "real-world" interactions and is challenging due to disfluencies. Also, meetings are typically informal and include many interruptions, overlapped speech (or multiple people speaking at the same time), and backchannels. Furthermore, there are interesting socials interactions that occur between the meeting participants [45].

Given the prevalence of speaker diarization work in the meetings domain as well as the usefulness of meeting analysis, the focus of this speaker diarization work is on the meeting domain. More specifically, we focus on the NIST RT evaluation datasets.

A number of different microphone conditions are examined within the meeting domain. In this work, we focus on the multiple distant microphone (MDM) and single distant microphone (SDM) conditions. In the MDM condition, recordings are taken from an array of microphones. However, in the SDM condition only a single microphone recording is utilized. These audio conditions are commonly used in the NIST RT evaluations [48]. As demonstrated throughout this thesis, MDM results are significantly better than SDM results.

Table 2.1: Names of meetings studied in this thesis.

Corpus	Meeting Names	
RT-02	LDC_20011116-1400	LDC_20011116-1500
	ICSI_20010208-1430	
RT-04S	NIST_20030623-1409	NIST_20030925-1517
	ICSI_20000807-1000	
RT-05S	AMI_20041210-1052	AMI_20050204-1206
	CMU_20050228-1615	CMU_20050301-1415
	VT_20050304-1300	VT_20050318-1430
RT-06	CMU_20050912-0900	CMU_20050914-0900
	EDI_20050216-1051	EDI_20050218-0900
	NIST_20051024-0930	NIST_20051102-1323
	VT_20050623-1400	VT_20051027-1400
RT-07	CMU_20061115-1030	CMU_20061115-1530
	EDI_20051113-1500	EDI_20051114-1500
	NIST_20051104-1515	NIST_20060216-1347
	VT_20050408-1500	VT_20050425-1000
RT-09	EDI_20071128-1000	EDI_20071128-1500
	IDI_20090128-1600	IDI_20090129-1000
	NIST_20080201-1405	NIST_20080227-1501
	NIST_20080307-0955	

2.4 Scoring metric

There exist a number of metrics used to evaluate speaker diarization systems. In this section we describe the Diarization Error Rate (DER), the metric used throughout this thesis.

The Diarization Error Rate (DER) defined by NIST [48] is the most commonly used metric in speaker diarization. The DER represents a time weighted accuracy of a speaker

diarization system. In order to compute the DER, first an optimal one-to-one mapping of reference speakers to system output speakers is determined. The DER is then the sum of the per speaker false alarm time (the amount of time the system overestimates the number of speakers, T_{FA}), miss time (the amount of time the system underestimates the number of speaker, T_{MISS}), and speaker error time (the amount of time the hypothesized speaker(s) is (are) not matched to the reference speaker(s), T_{SPKR}) divided by the total speech time in an audio file (T_{SPEECH}), as shown in Equation (2.6). A 0.25 second no-score "collar" is placed at the beginning and end of each segment boundary (as defined in the reference) in order to not penalize slight discrepancies in the start and end times of the speech segments.

$$\text{DER} = \frac{T_{FA} + T_{MISS} + T_{SPKR}}{T_{SPEECH}} \tag{2.6}$$

Overlapped speech errors have been a long-standing, known source of speaker diarization error [12, 36, 60]. Therefore, we further examine each of the three types of errors (T_{FA}, T_{MISS}, and T_{SPKR}) in terms of times during overlapped speech and during single speaker speech. Note that the DER metric, counts overlapped speech time multiple times. For instance, if three people are speaking simultaneously for 1.5 seconds then the total speech time (T_{SPEECH}) is 4.5 seconds.

Chapter 3

Analysis of multiple speaker diarization systems

The focus of this chapter is to analyze state-of-the-art speaker diarization systems. In doing so, it is possible to pinpoint their weaknesses and determine trends between the speaker diarization methods. Furthermore, determining where speaker diarization systems perform poorly helps focus attention to areas in need of improvement.

More specifically, the outputs from six state-of-the-art speaker diarization systems are investigated[1]. These systems have performed well on NIST Rich Transcription (RT) evaluation datasets and use a wide variety of speaker diarization techniques. We examine performance beyond the typical breakdown of speaker diarization error rate: false alarm, miss, and speaker errors. Specifically, we analyze the performance for a number of segment types, including short/long segments and segments surrounding speaker change points.

This chapter is outlined as follows: Section 3.1 describes previous work in speaker diarization error analysis; Section 3.2 defines the types of segments investigated; Section 3.3 provides the experimental setup used in this analysis; and Sections 3.4 and 3.5 discuss the results and conclusions of the analysis.

3.1 Previous work in speaker diarization error analysis

Thus far, there have been two main methods of performing speaker diarization analysis. The first compares performance between systems using characteristics of the recording, such as the number of speakers and average speaker turn duration (or the amount of time between speaker change points) [44, 14]. The second method involves replacing components of a given system with oracle components and calculating the effect on the DER [38, 37].

For instance, in [44], Mirghafori and Wooters studied how characteristics of entire broadcast news recordings correlated with Diarization Error Rate (DER) derived statistics (e.g.,

[1]We are grateful to all six sites for sharing their full system outputs.

mean and standard deviation). The recording characteristics fell into four main categories: speaker count features (e.g., total number of speakers, number of male speakers, number of female speakers), conversation turn features (e.g., number of conversation turn changes per minute, total number of turns, mean turn duration), speaker duration features (e.g., normalized entropy of total speaker duration, "do-nothing" score), and show duration features (total show duration, duration of non-scored regions, etc.). These characteristics were computed for two types of audio files: "nuts" – recordings which have high DER and are "hard to crack" – and "flakes" – recordings which are sensitive to tuning parameters. For "nutty" recordings, the number of speakers and number of turns had the highest correlation with the DER. For "flaky" recordings, the do-nothing DER – a measure of the percentage of total speech time assigned to one speaker – had the highest correlation with the standard deviation of the DER scores for a number of parameter settings. In [14], Bozonnet et al. combined a top-down and bottom-up diarization system. Before combining the two systems, Bozonnet et al. investigated the strengths and weaknesses of the two systems, concluding that the top-down system better estimated the true number of speakers while the bottom-up system output better matched the reference transcription in terms of the number of segments and average segment duration.

The second method of error analysis involves replacing components of a given system with oracle components and calculating the effect on the DER [38, 37]. For example, Huijbregts et al. began with an oracle diarization system, where each component (e.g., speech activity detection, initialization, merging criterion, stopping criterion) was an oracle component which utilized the reference transcription. In both a top-down and bottom-up fashion, each oracle component was replaced with its speaker diarization system component and the change in DER before and after the replacement reflected the effect that component (and potentially subsequent components) had on the DER. The areas which contributed most to the DER were speech activity detection, robustness of the merging criterion with respect to maintaining cluster purity, and the inability to address the overlapped speech problem.

3.2 Investigated segment types

The previous analysis work focused on correlating diarization performance with attributes of the recording and computing the change in DER associated with each component of the system. By contrast, the goal in this chapter is to characterize which types of segments are difficult for six state-of-the-art speaker diarization systems. These results provide insight into where speaker diarization researchers should focus their attention in order to further improve speaker diarization.

Speaker diarization performance is evaluated for two types of segments: segments categorized based on the segment duration and segments categorized based on their proximity to speaker change points. In this work, a *segment* is defined according to the reference speaker diarization segmentation. The reference segmentation is created by first force aligning the

individual headset microphone audio recordings to the reference transcripts using LIMSI tools. Then the word boundaries obtained from the forced alignment are smoothed using a 0.3 second window, thereby grouping multiple words together into a segment [6].

3.2.1 Segment Duration

Speaker diarization system performance is evaluated based on the duration of each segment. More specifically, the DER is computed for ten bins of segment durations, where each bin contains segments of similar duration. For illustration, Figure 3.1 shows an example reference segmentation which is split into three bins of segment durations (short, intermediate, and long segments).

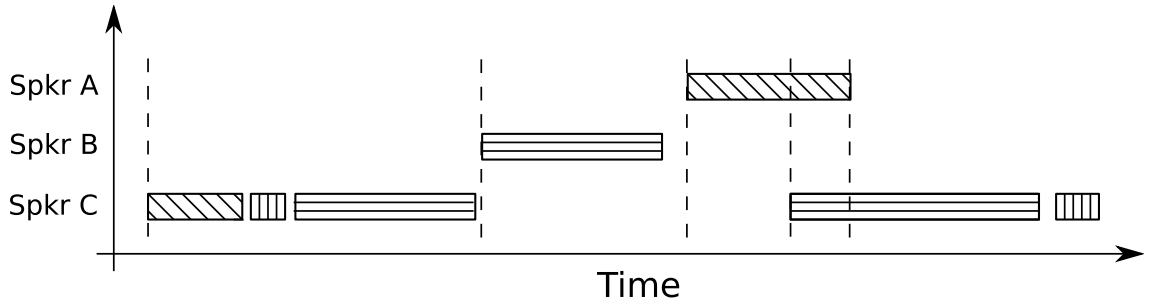

Figure 3.1: Example reference segmentation containing three speakers (A, B, and C). Speaker change points are shown using vertical dashed lines. The short, intermediate, and long segments are filled with vertical, diagonal, and horizontal lines, respectively.

3.2.2 Speaker Change Points

Segments surrounding speaker change points are also examined. In this work, a *speaker change point* is defined as an instance in which the current speaker(s) differs from the previous speaker(s). Nonspeech segments are ignored since most speaker diarization systems similarly discard these segments [35, 26, 57, 47, 15]. Thus, if a speaker talks for some time, pauses, and then resumes talking there is no speaker change point when the speaker resumes talking. In the case of overlapped speech, as shown in Figure 3.2, a speaker change point occurs both when the number of speakers increases from one to two and when the number of speakers decreases from two to one. This follows from the definition of a speaker change point. A segment is labeled a first segment after a speaker change point (*FirstAfter*) if any portion of the segment immediately follows a speaker change point. Similarly, a segment is labeled a last segment before a speaker change point (*LastBefore*) if any portion of the segment immediately precedes a speaker change point. Examples of FirstAfter and LastBefore segments are shown in Figure 3.2.

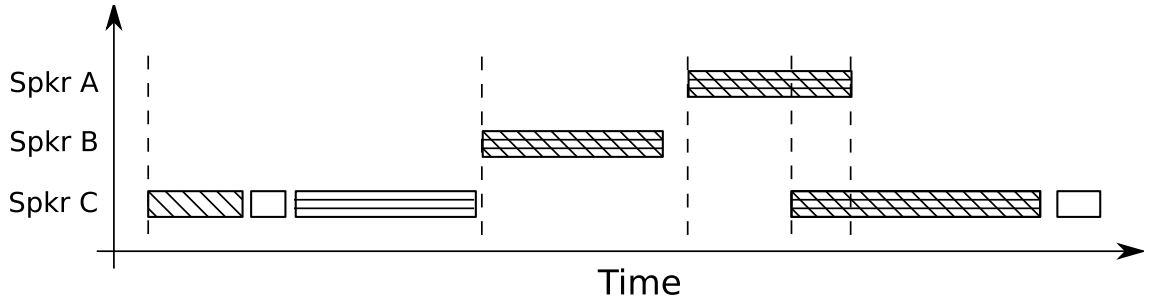

Figure 3.2: Visualization of first segments after speaker change points (FirstAfter) and last segments before speaker change points (LastBefore). Segments filled with diagonal lines are FirstAfter segments. Segments filled with horizontal lines are LastBefore segments.

3.3 Analysis setup

3.3.1 Speaker Diarization Systems

The final segmentation from six speaker diarization systems (AMI [35], ICSI [26], Idiap [57], IIR-NTU [47], LIA-Eurecom [15], and MIT [23]) are analyzed in this work. These systems represent the state-of-the-art in speaker diarization and have consistently performed well in the NIST Rich Transcription evaluations. The following paragraphs provide brief descriptions of the algorithms.

The AMI speaker diarization system [35] uses a bottom-up HMM-GMM framework, as described in Section 2.2.1. A uniform initialization approach is utilized, where the number of initial clusters is determined based on the duration of the audio recording. The ΔBIC metric described in Equation (2.1) is used to determine which clusters to merge as well as when to stop merging clusters. The total number of parameters is kept constant after merging two clusters (i.e. the number of parameters after merging is equal to the sum of the number of parameters in each of the clusters to be merged), thereby eliminating the last term in Equation (2.1). This simplification will be referred to as the *modified* ΔBIC criterion. Therefore, the modified ΔBIC value for two clusters C_1 and C_2 is:

$$\text{modified } \Delta BIC(C_1, C_2) = \log \frac{p(x_{1,2}|\theta_{1,2})}{p(x_1|\theta_1)p(x_2|\theta_2)}. \tag{3.1}$$

For the single distant microphone (SDM) condition, Mel-Frequency Cepstral Coefficients (MFCCs) are used to perform speaker diarization. For the the multiple distant microphone (MDM) condition, both MFCCs and time delay of arrival (TDOA) features obtained from *BeamformIt* [11] are used.

The ICSI speaker diarization system also utilizes a bottom-up HMM-GMM approach [26]. It is very similar to the AMI system. The differences between the two systems include

the initialization procedure and some of the parameter selection. The ICSI system performs initialization based on long-term features extracted from the audio and uses an automatic method to determine both the number of initial clusters as well as the number of mixtures in each initial cluster.

The Idiap speaker diarization system [57] uses an agglomerative information bottleneck (aIB) approach, as described in Section 2.2.3. The Idiap system only participated in the MDM condition. It uses four features streams: MFCC, TDOA, modulation spectrum, and frequency domain linear prediction features.

The IIR-NTU [47] speaker diarization system is yet another bottom-up HMM-GMM based algorithm. The IIR-NTU system uses Linear Prediction Cepstral Coefficients (LPCCs) instead of MFCCs. The IIR-NTU system utilizes a different initialization technique. For the MDM condition, the TDOA features for each microphone pair are first quantized. Then based on the quantized TDOA values, the nine most frequent occurring quantized vectors are chosen to be the centroids. Finally, the quantized feature vectors are assigned to the nearest centroid and this is the initial clustering for the MDM condition. For the SDM condition, first uniform clustering is performed and GMM models are trained for each of the initial clusters. Each cluster is then split into 0.5 second segments. Then the GMMs are re-trained on the top 25% of the segments in each cluster. In other words, the models are re-trained on the "pure" segments. The rest of the segments are iteratively classified and the GMMs are re-trained, after which Viterbi is used to re-segment the data. After which, the final initial segmentation is obtained. After the initialization procedure, the IIR-NTU system follows the standard bottom-up HMM-GMM approach of merging clusters and re-training/re-segmenting until the stopping criterion is met. However, unlike the AMI and ICSI systems, which use the modified ΔBIC criterion for the merging and stopping criterion, the IIR-NTU system uses a modified version of the cross likelihood ratio to determine which clusters to merge [47] and a variant of the student's T-test to determine when to stop the algorithm [46].

The LIA-Eurecom system [15] is a top-down HMM-GMM system. For both the SDM and MDM conditions only normalized Linear Frequency Cepstral Coefficients (LFCCs) are used to perform speaker diarization. The LIA-Eurecom algorithm also incorporates a "purification" step, where the GMMs are trained on the top 55% of the data in each cluster. Unlike the IIR-NTU system which performs "purification" during the initialization procedure, the LIA-Eurecom system performs "purification" at the end of the algorithm.

Finally, the MIT speaker diarization system is an HDP-HMM system, as described in Section 2.2.2. The MIT system only participated in the SDM condition. It uses MFCC features, which are averaged over non-overlapping 0.25 second windows.

3.3.2 Data

This analysis is performed on the NIST Rich Transcription 2009 (RT-09) evaluation dataset. The RT-09 evaluation is the most recent NIST RT evaluation. The scores for this

evaluation are much worse than scores for previous NIST RT evaluations, which is likely due to the increased amount of overlapped speech. The RT-09 dataset consists of seven meetings recorded at three sites: Idiap, Edinburgh, and NIST. There is approximately one hour of meeting data from each of the three sites. Both the multiple distant microphone (MDM) and single distant microphone (SDM) conditions are analyzed in this chapter.

3.3.3 Scoring metric

The Diarization Error Rate (DER) defined by NIST [48] and described in Chapter 2.4 is used to evaluate each system's performance. Since a number of the errors occur during overlapped speech, each of the three types of errors (false alarm (FA), miss ($MISS$), and speaker ($SPKR$)) are further split into times during overlapped speech and during single speaker speech. As a reminder: a miss error occurs if the system underestimates the number of speakers; a false alarm error occurs if the system overestimates the number of speakers; and a speaker error occurs if the hypothesized speaker(s) does not match the reference speaker(s). Thus if a speaker error occurs during overlapped speech, this means that the hypothesized speaker matches *none* of the reference speakers.

Note that each of the segment types studied in this work are labeled using the reference transcription. Therefore, nonspeech time (as transcribed in the reference) is not scored in this study. Thus, the only way to have a false alarm error would be if a system is able to hypothesize overlapped speech, or more than one speaker speaking at a given instance. Only the AMI system hypothesizes overlapped speech. The other five systems annotate at most one speaker at a given time. In order to retain the anonymity of the systems, the false alarm errors are not shown. This does not have an impact on the results of work since the false alarm error rate during speech time is negligible.

3.4 Analysis results

In this section, the results are presented for the MDM and SDM conditions. In order to maintain anonymity, the results are shown in terms of Systems A, B, C, D, and E. Although there are six systems in total, only five systems participated in both the MDM condition (all systems except MIT) and SDM condition (all systems except Idiap).

Regarding the figures shown in this section, the DER is color coded according to the type of error it is (miss and speaker error). The miss and speaker error rates are further split into times containing overlapped and single speaker speech. The miss rates during overlapped and single speaker speech are annotated as light red and red, respectively. Similarly, the speaker error rates during overlapped and single speaker speech are annotated as light blue and blue, respectively. Thus, the total height each bar (the sum of the miss rates and speaker error rates for overlapped and single speaker speech) reflects the total DER, less the false alarm errors. As explained in Section 3.3.3, the false alarm errors are not included since

we are examining results in terms of the reference segmentation. In Figures 3.3 and 3.4, we show the total DER results for all five systems, A-E, in descending order (i.e. from worst performing system, in terms of overall DER, to best performing system). This ordering is maintained throughout the chapter.

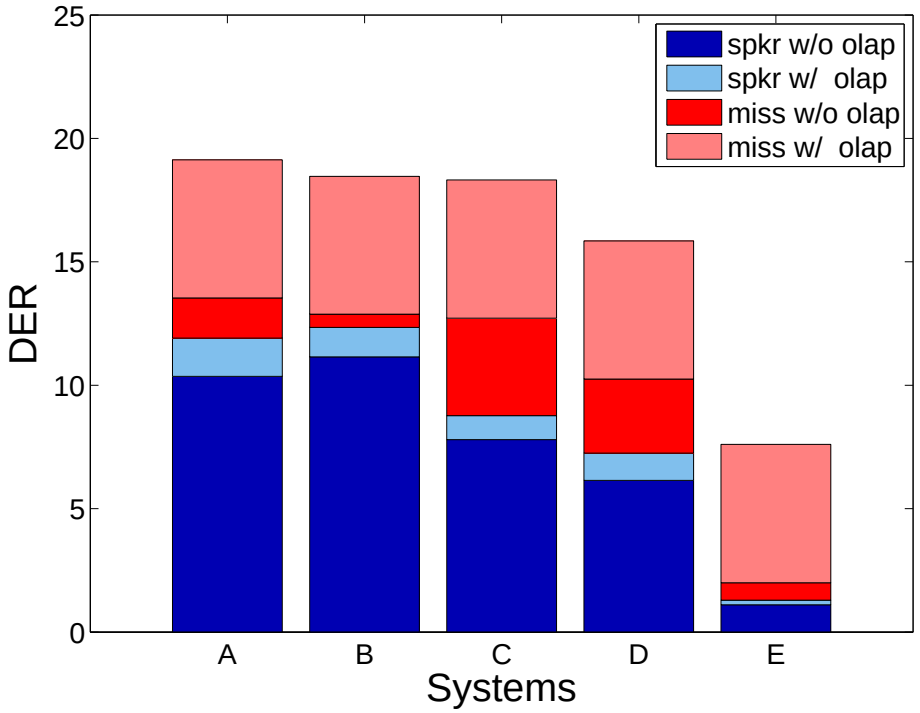

Figure 3.3: MDM condition: A breakdown of the total DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for each of the five systems. Note that the false alarm rate does not have a major impact on the total DER and is therefore not shown (see Section 3.3.3). The systems are arranged in descending order according to the overall DER.

3.4.1 Segment duration

The DERs for each of the systems are computed over segments of similar duration. The scored segments are split into ten bins with approximately the same number of segments in each bin. For each bin, the DER is calculated for all of the systems as shown in Figures 3.5 and 3.6 for the MDM and SDM conditions, respectively. For the MDM and SDM conditions, all five systems display the same trend: the overall diarization error rate decreases as the duration of the segments increase. The miss rate (particularly due to overlapped speech) plays a large role in the decreasing DER. The speaker error rate tends to decrease as the

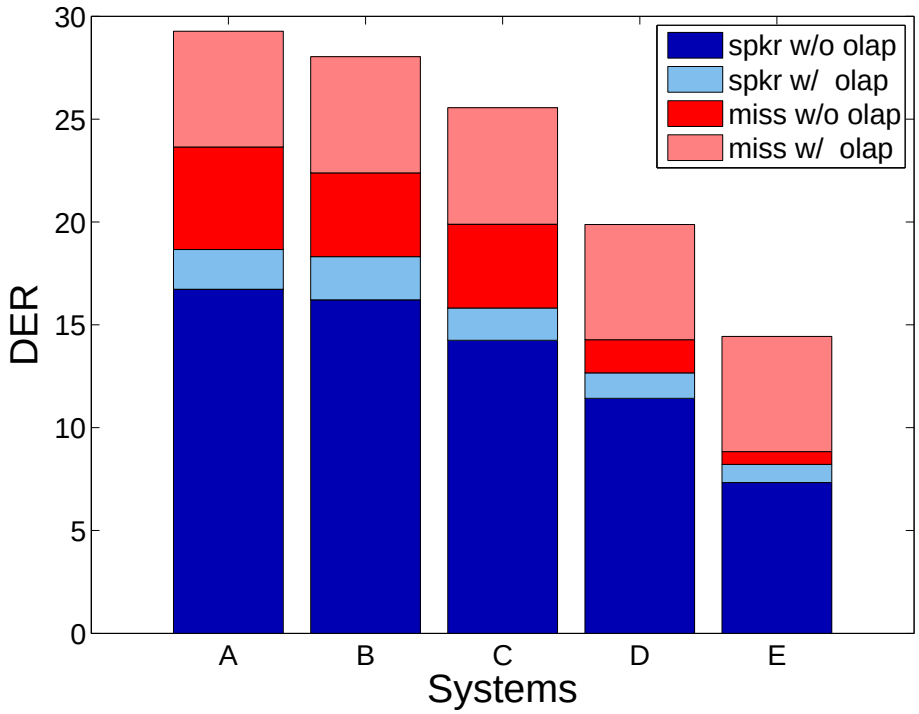

Figure 3.4: SDM condition: A breakdown of the total DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for each of the five systems. Note that the false alarm rate does not have a major impact on the total DER and is therefore not shown (see Section 3.3.3). The systems are arranged in descending order according to the overall DER.

segment duration increases; however, this trend is not very consistent and often oscillates as opposed to solely decreasing. The speaker error rate trends decrease more consistently for the better performing systems for both the MDM and SDM conditions. The other systems typically see a decrease in the speaker error rate between the shortest and longest segments. However, there is not a consistent downward trend. For instance, for the worst performing MDM system the speaker error rate is worst for segments of duration 1.00-1.25 seconds.

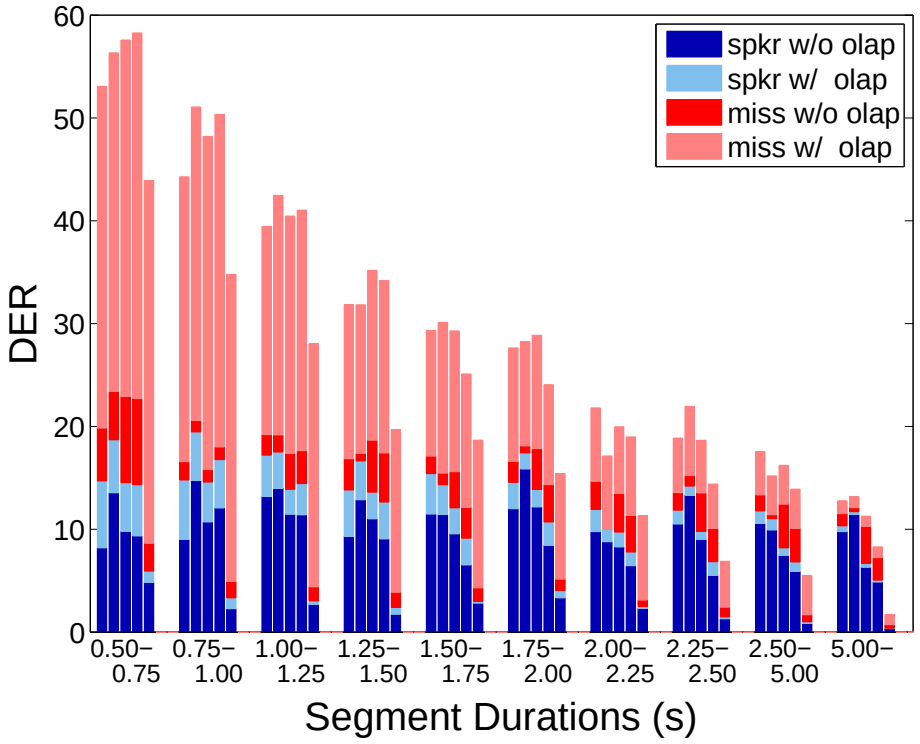

Figure 3.5: MDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for various segment duration lengths. Each segment durations bin (e.g., from 0.50-0.75 seconds) contains five bars, where each bar represents one of the five analyzed systems. The systems are arranged in descending order according to the overall DER. Note that the shortest segment is greater than 0.5 seconds due to the ± 0.25 no-score "collar" placed at the start and end of each segment boundary.

Although the DER is very bad for short segments, recall that each bin contains roughly the same number of segments. This means that bins containing short segments will contain less of the total scored speech time than bins containing long segments, as indicated by the horizontal red line in Figures 3.7 and 3.8. Since the DER is a function of the total scored speech time, as described in Section 2.4, short segments do not have much of an impact on

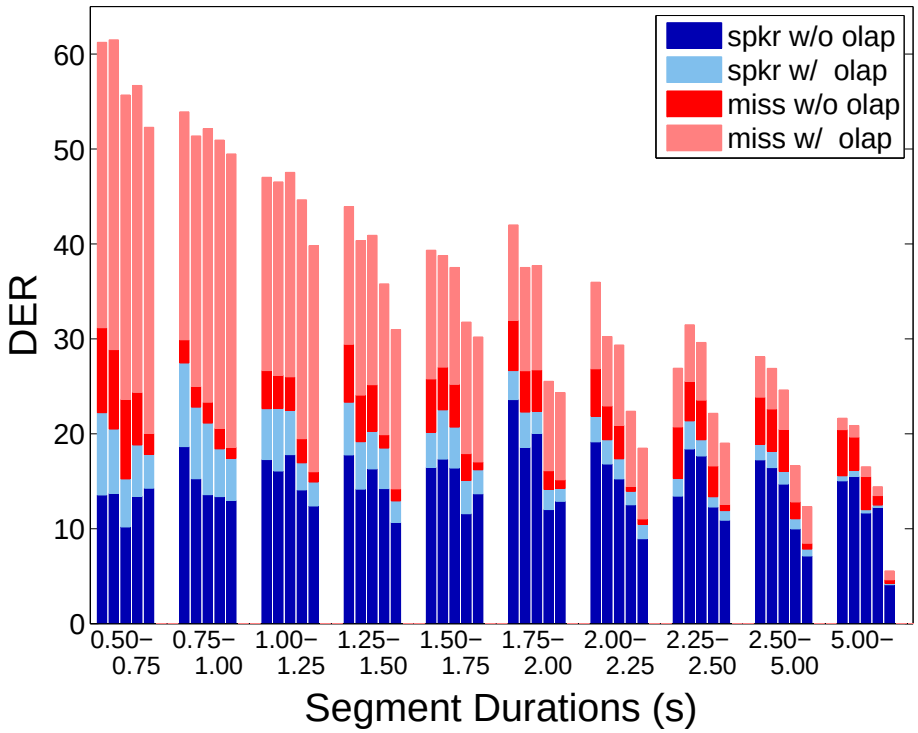

Figure 3.6: SDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for various segment duration lengths. Each segment durations bin (e.g., from 0.50-0.75 seconds) contains five bars, where each bar represents one of the five analyzed systems. The systems are arranged in descending order according to the overall DER.

the total DER. This is shown in detail in Figures 3.7 and 3.8, which display the percent of each system's DER contained in each of the bins shown in blue as well as the percent of scored speech time in each of the bins shown in red. In fact from Figures 3.7 and 3.8, we see that less than 5% of each system's DER occurs during segments between 0.50 and 0.75 seconds long despite the DERs greater than 40% and over 70% of scored time occurs during segments greater than 2.5 seconds long.

Note that due to the ±0.25 second collar, the minimum scored segment duration time is greater than 0.50 seconds. Also, for each segment duration range (e.g., 0.51-0.75, 0.75-1.00, 1.00-1.25 seconds), the plots show the results for all five systems, A-E, in descending order (i.e. from the worst performing system to best performing system, in terms of overall DER).

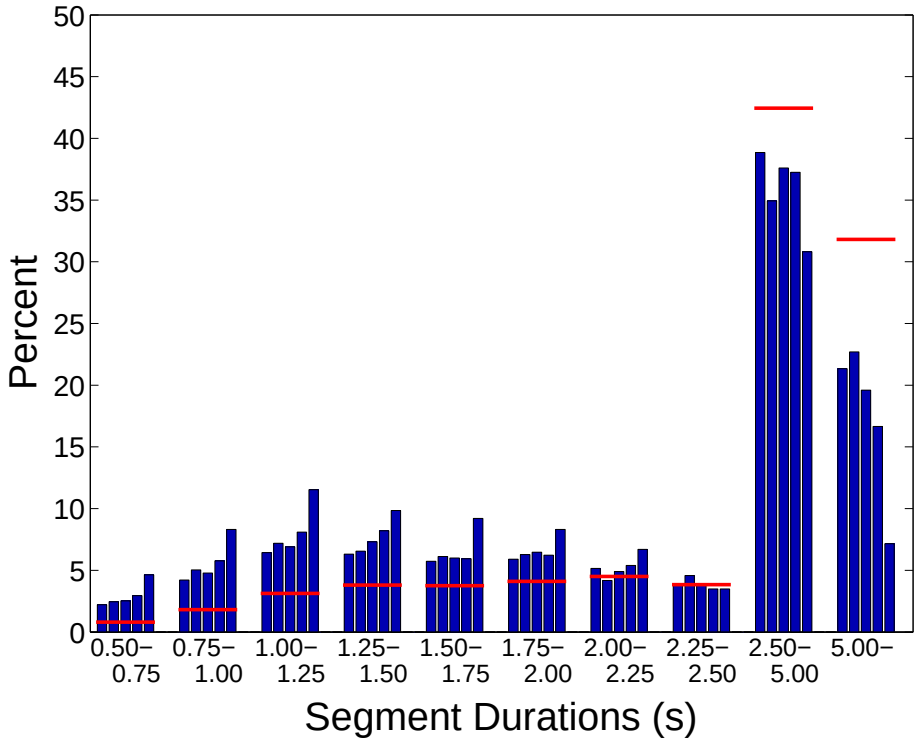

Figure 3.7: MDM condition: Percent of total system DER contained in each segment duration bin (shown in blue). The horizontal red line in each segment duration bin represents the percent of scored speech time in each bin. Recall that DER is a function of the scored speech time, so bins with more scored speech time have a larger impact on the total DER. Each segment duration bin (e.g., from 0.50-0.75 seconds), contains five blue bars representing the five systems that are analyzed.

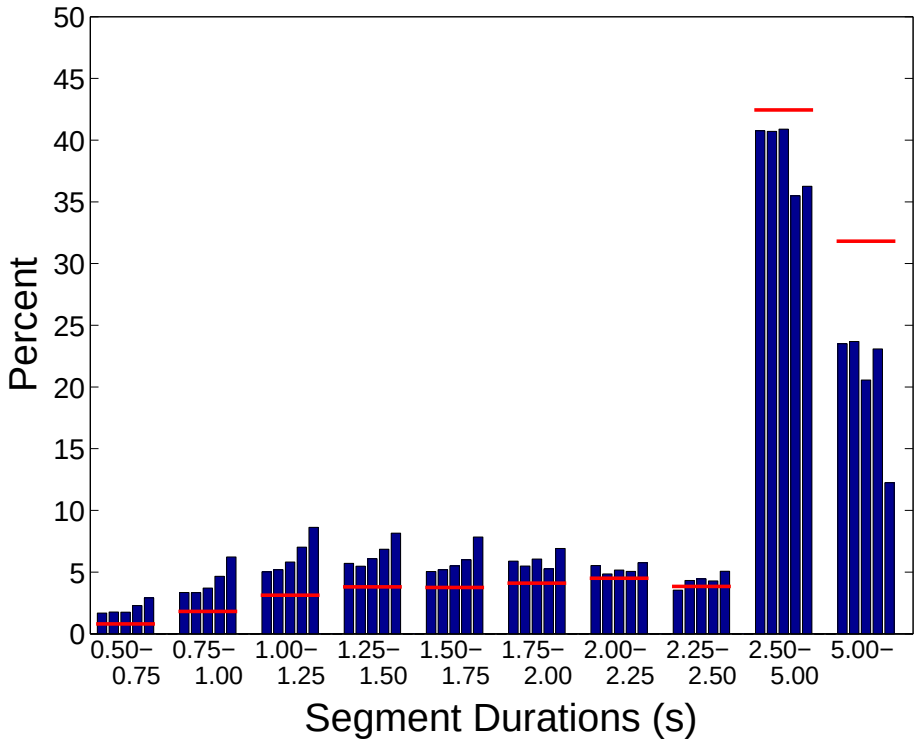

Figure 3.8: SDM condition: Percent of total system DER contained in each segment duration bin (shown in blue). The horizontal red line in each segment duration bin represents the percent of scored speech time in each bin. Recall that DER is a function of the scored speech time, so bins with more scored speech time have a larger impact on the total DER. Each segment duration bin (e.g., from 0.50-0.75 seconds), contains five blue bars representing the five systems that are analyzed.

3.4.2 Speaker change points

The errors surrounding speaker change points are also examined. In Figures 3.9 and 3.10, the DERs for each of the systems are shown for segments immediately following a speaker change point (FirstAfter) and not (not FirstAfter). For both the MDM and SDM conditions, FirstAfter segments perform significantly worse than not FirstAfter segments both in terms of the miss rate and speaker error rate. Though the decrease in the miss rate is largely due to misses during overlapped speech, the speaker error rate decreases significantly for single speaker speech in addition to overlapped speech. Similar results are obtained for the segments immediately before a speaker change point (LastBefore) as shown in Figures 3.11 and 3.12. Since a segment is classified as FirstAfter if *any* portion of the segment immediately follows a change point (and similarly for LastBefore segments), not FirstAfter and not LastBefore segments did not contain any overlapped speech.

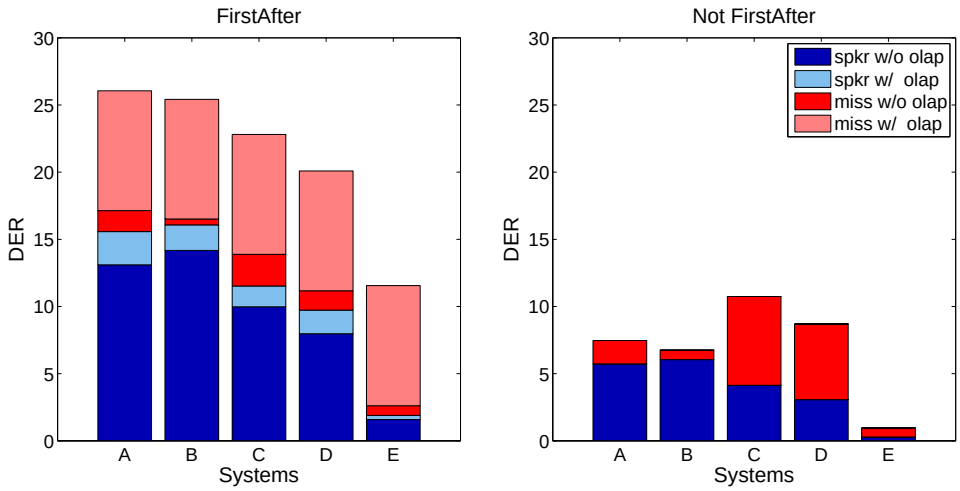

Figure 3.9: MDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for segments immediately following (FirstAfter) and not following (not FirstAfter) a speaker change point.

Both short segments and segments preceding or following speaker change points performed worse than their counterparts. In order to verify that these are in fact two separate types of errors (and it is not the case that segments preceding and following speaker change points are dominated by short segments) we computed the cumulative distribution functions (CDFs) of the segment durations for segments immediately after and before speaker change points as well as their respective complements (i.e. not FirstAfter and not LastBefore). The distributions are shown in Figure 3.13. The CDFs of the segment durations for FirstAfter and LastBefore segments lie on top of one another, as do the CDFs for not FirstAfter and

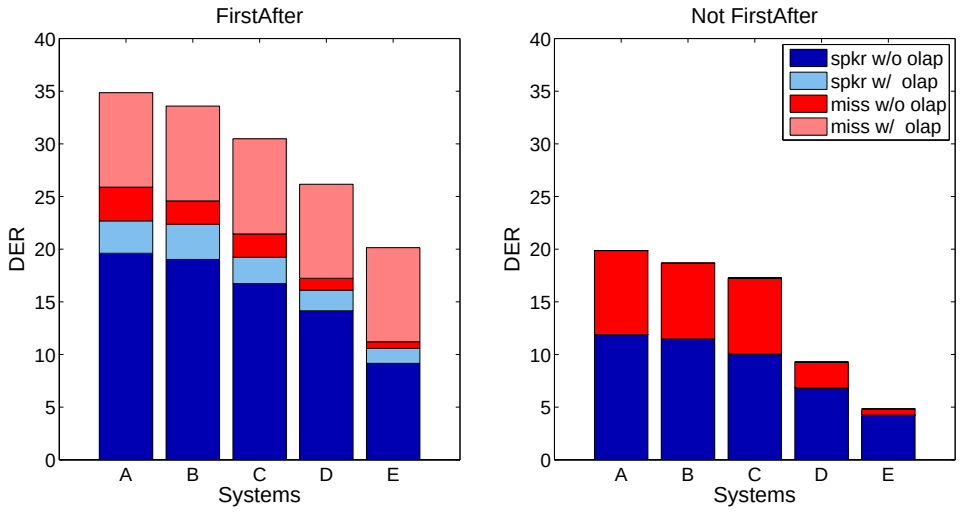

Figure 3.10: SDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for segments immediately following (FirstAfter) and not following (not FirstAfter) a speaker change point.

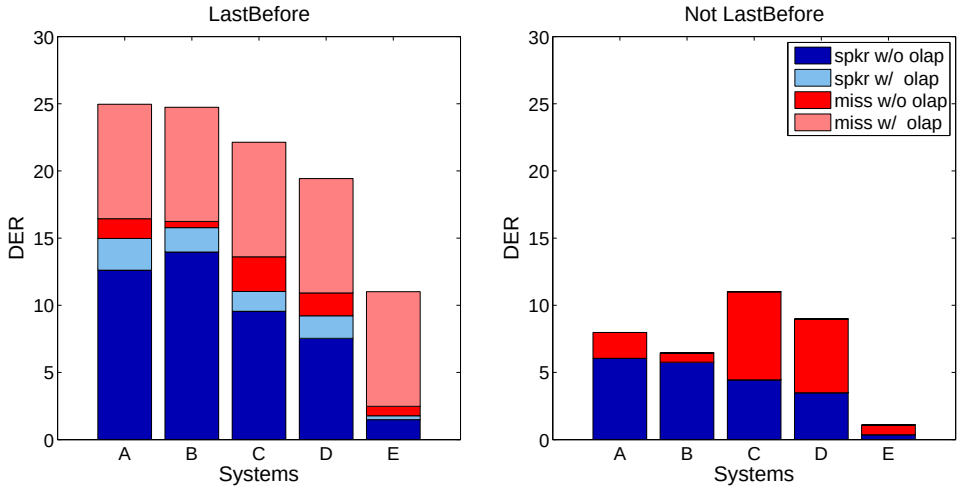

Figure 3.11: MDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for segments immediately preceding (LastBefore) and not preceding (not LastBefore) a speaker change point.

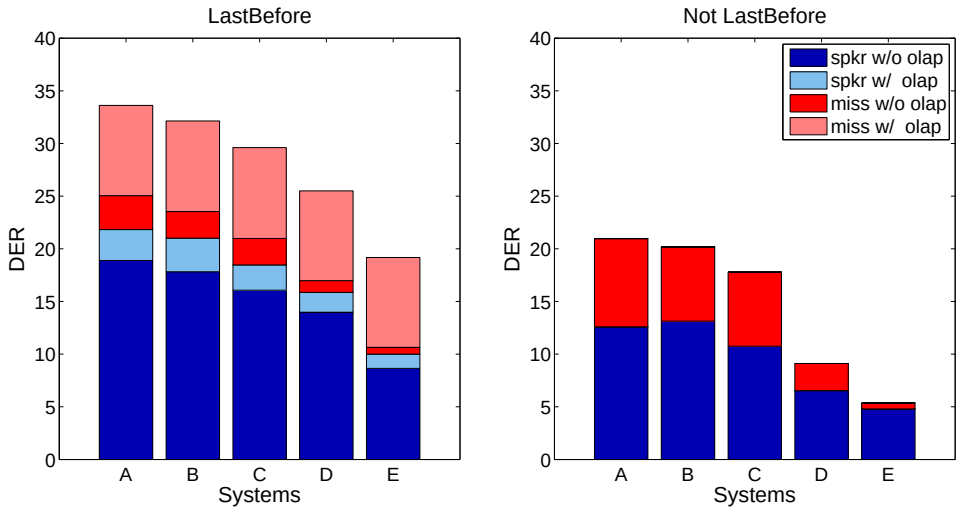

Figure 3.12: SDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for segments immediately preceding (LastBefore) and not preceding (not LastBefore) a speaker change point.

not LastBefore segments. Although FirstAfter and LastBefore segments have slightly more short segments, all four CDFs are quite similar.

Next, we investigate the time surrounding speaker change points in more detail. Instead of grouping an entire segment together, we split each segment into 0.25 second intervals. We then plot the DER as a function of the time after/until the previous/next change point as shown in Figures 3.14 and 3.15 for the MDM and SDM conditions, respectively. These figures demonstrate that the systems have a more difficult time closer to change points than farther away from change points.

The overall DER decreases considerably as the time to the last/next change point increases. This decrease is largely due to the miss rate during overlapped speech. Once again, for better systems as the time from the last change point or the time until the next change point increases, the speaker error rate for single speaker speech improves. However, for the other systems, the speaker error rate for single speaker speech trend arcs more. In other words, as the time from the last change point or until the next change point increases the speaker error rate first gets worse before getting better. When considering the total speaker error rate, the trend of getting better performance as the time from the change point increases is more evident. Surprisingly, the miss rates are worse immediately preceding a change point than immediately following a change point.

Next, we analyze the results in terms of the time to the closest change point, regardless of whether the change point is before or after the given instance. The results are shown in Figures 3.16 and 3.17. In this setting, the DER initially decreases dramatically and then

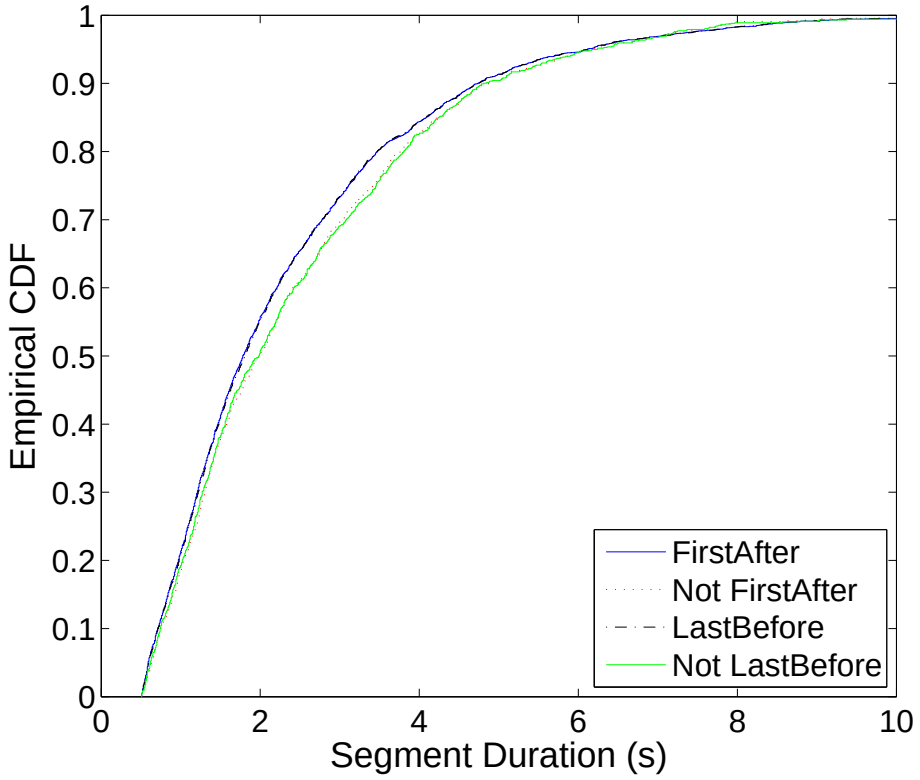

Figure 3.13: CDFs of segment durations for FirstAfter, not FirstAfter, LastBefore, and not LastBefore segments. Note that the CDFs are close to one another.

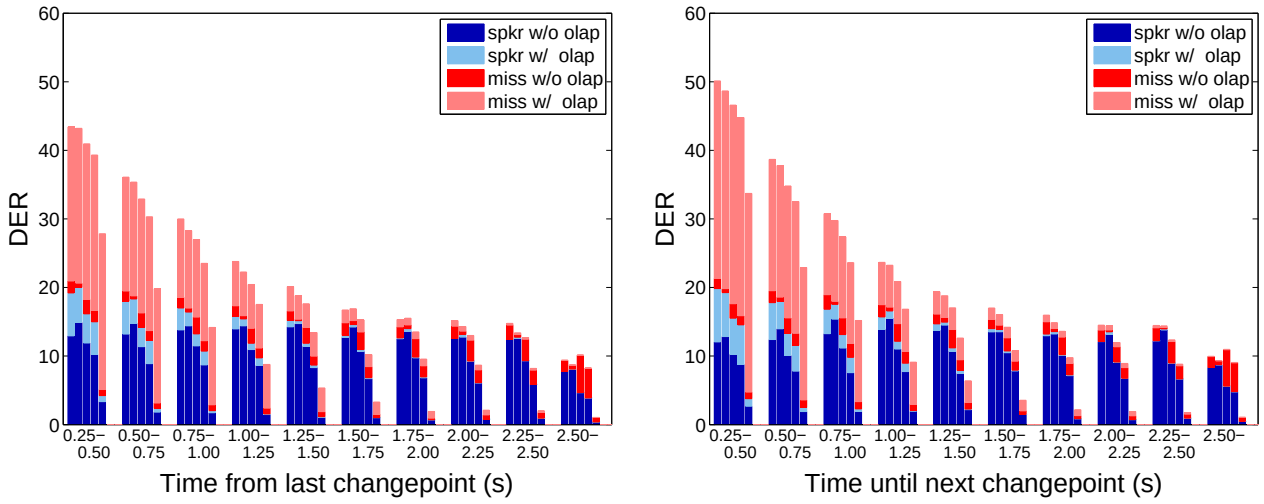

Figure 3.14: MDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) as a function of the time since the last speaker change point (on the left) and until the next speaker change point (on the right). Each distance from the change point bin (e.g., 0.25-0.50 seconds) contains five bars, one for each of the five analyzed systems. Note that the shortest distance to a speaker change point is 0.25 seconds due to the ±0.25 no-score "collar" placed at the start and end of each segment boundary.

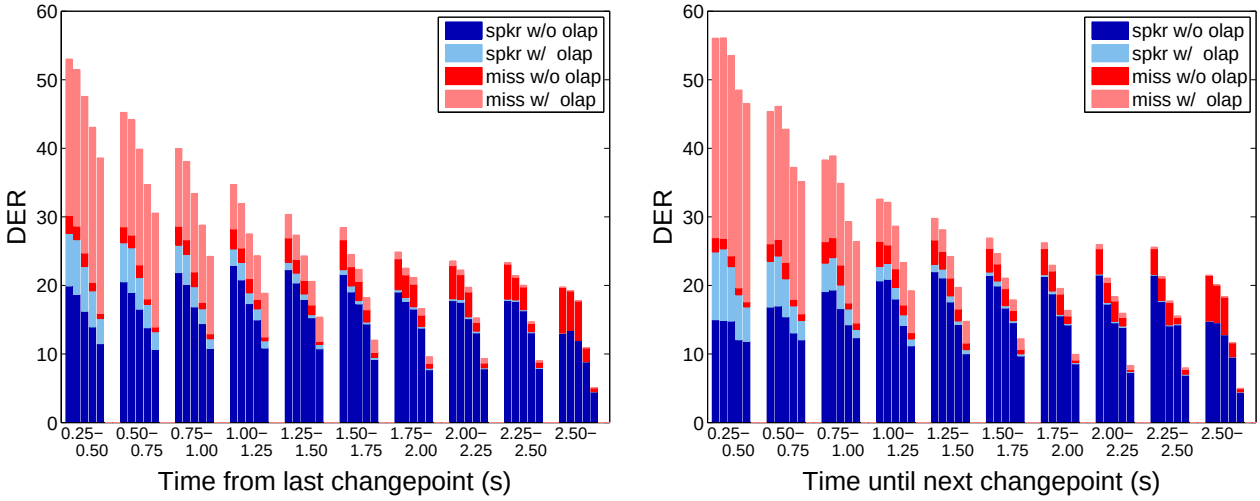

Figure 3.15: SDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) as a function of the time since the last speaker change point (on the left) and until the next speaker change point (on the right). Each distance from the change point bin (e.g., 0.25-0.50 seconds) contains five bars, one for each of the five analyzed systems.

remains steady. In Figures 3.18 and 3.19, we show the percent of scored speech time (shown in red) in terms of the proximity to a speaker change point. About 20% of the scored speech time is within 0.5 seconds of the change point. As the distance increases, it contains a smaller percentage of the scored time until the last bin, which contains all portions of the recording greater than 2.5 seconds from a speaker change point which accounts for approximately 35% of scored speech time. Then in Figure 3.18, we see that for all five MDM systems at least 40% of the DER occurred between 0.25 and 0.50 seconds from the speaker change point. For the SDM condition, at least 33% of the DER occurred within 0.5 seconds of the speaker change point, as demonstrated in Figure 3.19. Though again, a significant source of the decrease in DER as the time from a speaker change point increases is due to overlapped speech (mostly in terms of the miss rate and also partially due to the speaker error rate). In this thesis, we do not address the overlapped speech problem. Although a considerable amount of the error is due to overlapped speech, this is partially due to the definition of DER which multiply counts speech time during overlapped speech as described in Section 2.4. Instead, we focus on single speaker speech time in this thesis. Therefore in Figures 3.20 – 3.23 the results are shown again, this time without including any of the overlapped speech errors. Once again, the better systems show a more clear trend of better performance as the time from the change point increases while worse systems show more of an arc in performance (first getting worse and then better). After ignoring errors due to overlapped speech, the amount of error within 0.5 seconds of a change point is reduced to at least 22% and 18% for

the MDM and SDM conditions, respectively. Considering that 12% of scored time occurs within 0.5 seconds of a change point, this amount is still significant. Furthermore, at least 35% of the MDM DER and 31% of the SDM DER for single speaker speech occurs within 0.75 seconds of a change point (which accounts for 22% of the scored speech time). These percentages are even greater for the best performing system at 49% and 47% respectively.

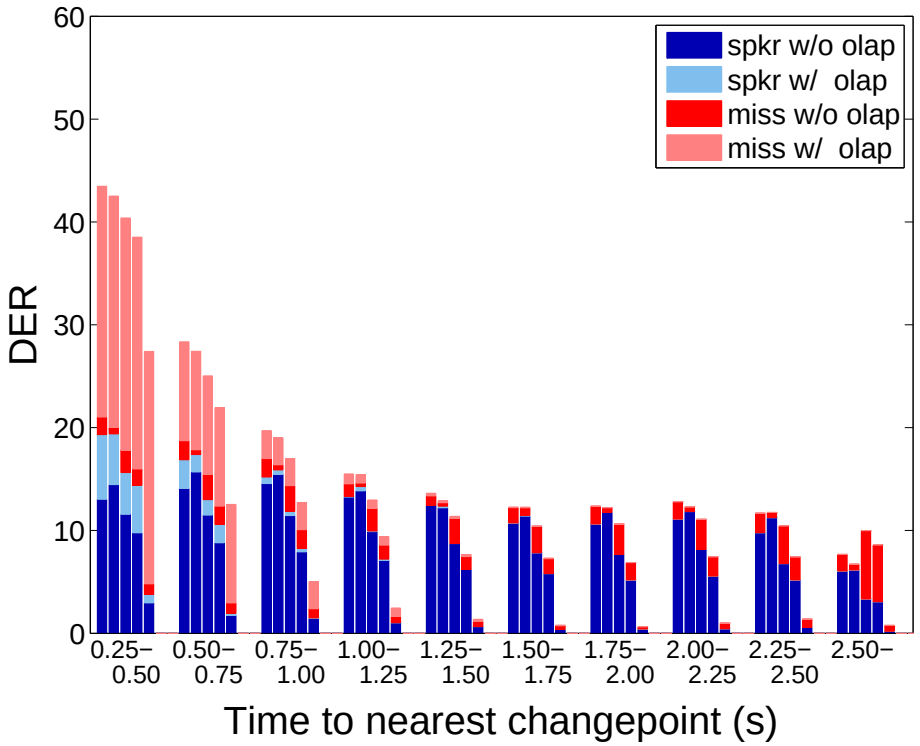

Figure 3.16: MDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) as a function of the time to the nearest speaker change point. Each distance from the change point bin (e.g., 0.25-0.50 seconds) contains five bars, one for each of the five analyzed systems. Note that the shortest distance to a speaker change point is 0.25 seconds due to the ±0.25 no-score "collar" placed at the start and end of each segment boundary.

3.5 Discussion

In conclusion, we have demonstrated two problematic types of segments for speaker diarization systems. For the MDM and SDM conditions, both short segments and segments surrounding speaker change points have significantly worse DERs than their counterparts for all five state-of-the-art speaker diarization systems.

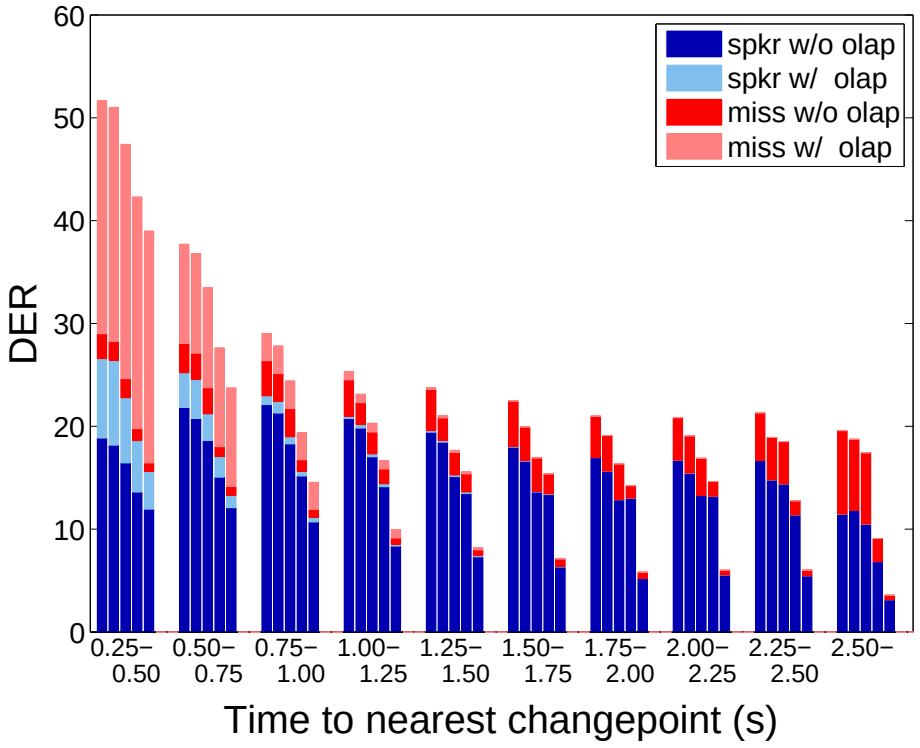

Figure 3.17: SDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) as a function of the time to the nearest speaker change point. Each distance from the change point bin (e.g., 0.25-0.50 seconds) contains five bars, one for each of the five analyzed systems. Note that the shortest distance to a speaker change point is 0.25 seconds due to the ± 0.25 no-score "collar" placed at the start and end of each segment boundary.

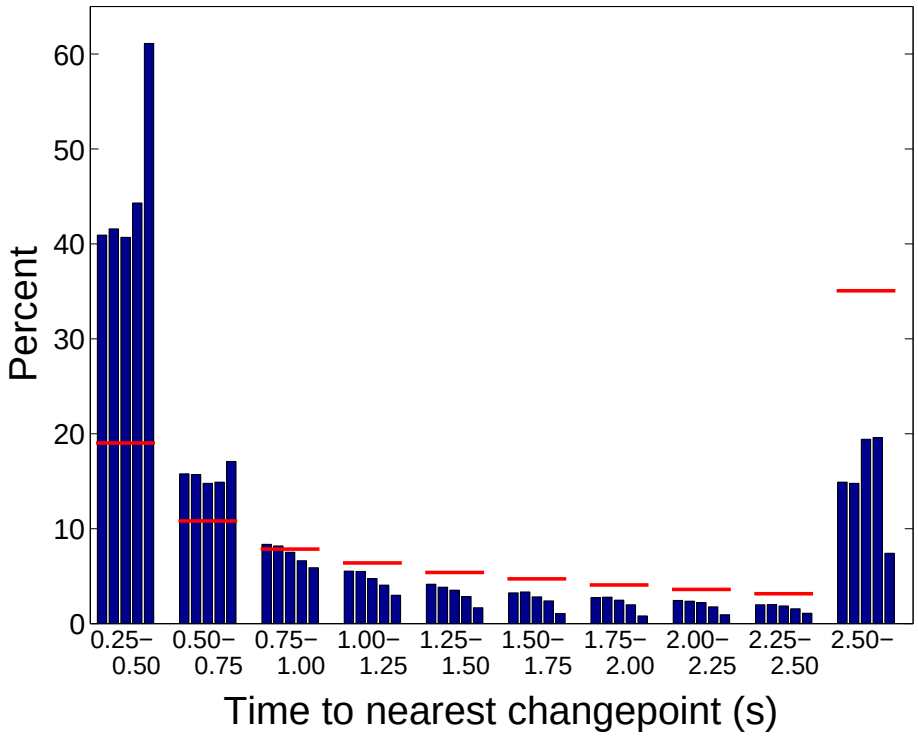

Figure 3.18: MDM condition: Percent of total DER contained in each distance to the speaker change point bin (shown in blue). The horizontal red line in bin represents the percent of scored speech time in each bin. Recall that DER is a function of the scored speech time, so bins with more scored speech time have a larger impact on the total DER. Each distance to the closest change point bin (e.g., from 0.25-0.50 seconds), contains five blue bars representing the five systems that are analyzed.

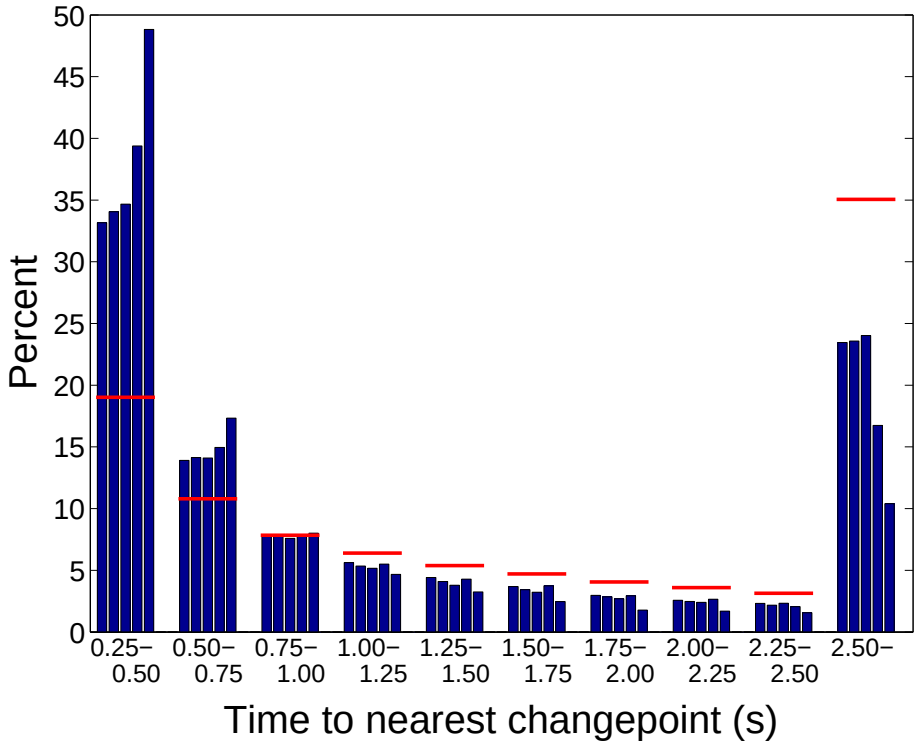

Figure 3.19: SDM condition: Percent of total DER contained in each distance to the speaker change point bin (shown in blue). The horizontal red line in bin represents the percent of scored speech time in each bin. Recall that DER is a function of the scored speech time, so bins with more scored speech time have a larger impact on the total DER. Each distance to the closest change point bin (e.g., from 0.25-0.50 seconds), contains five blue bars representing the five systems that are analyzed.

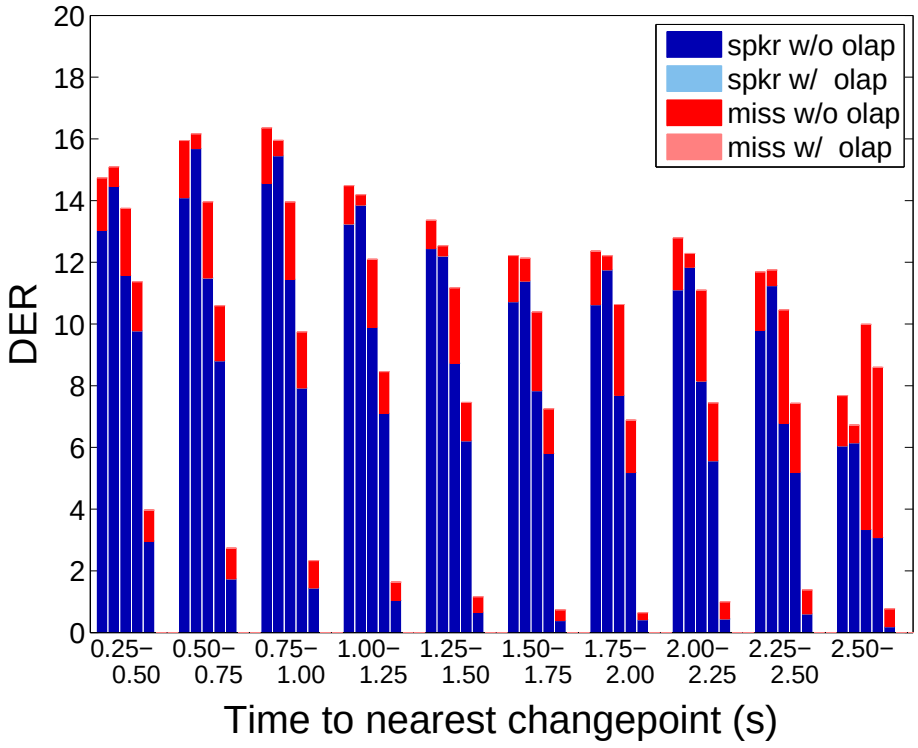

Figure 3.20: MDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for *only* single speaker speech (w/o olap)) as a function of the time to the nearest speaker change point. Overlapped speech time is not shown in this figure.

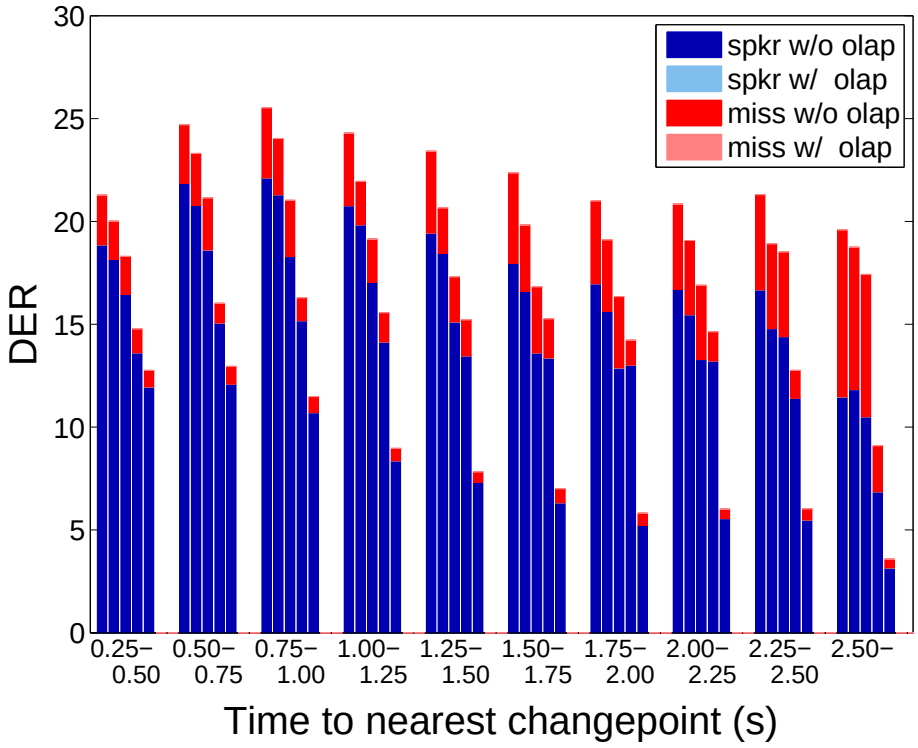

Figure 3.21: SDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for *only* single speaker speech (w/o olap)) as a function of the time to the nearest speaker change point. Overlapped speech time is not shown in this figure.

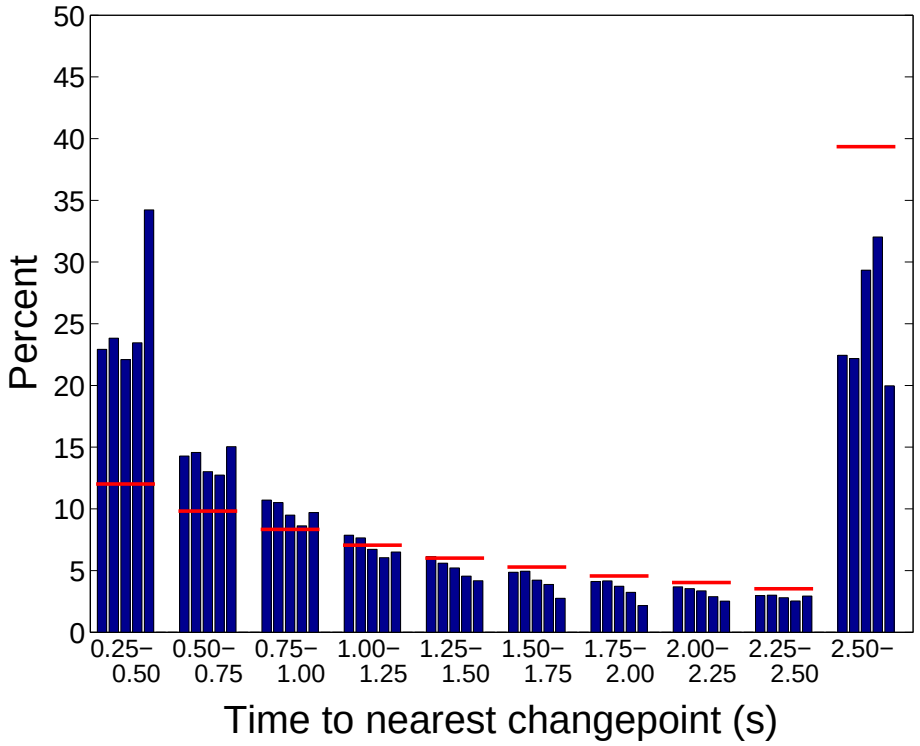

Figure 3.22: MDM condition: Percent of *single speaker speech* DER contained in each distance to the speaker change point bin (shown in blue). The horizontal red line in bin represents the percent of scored single speaker speech time in each bin. Overlapped speech is ignored in this figure in order to get a better visualization of the results for single speaker speech time.

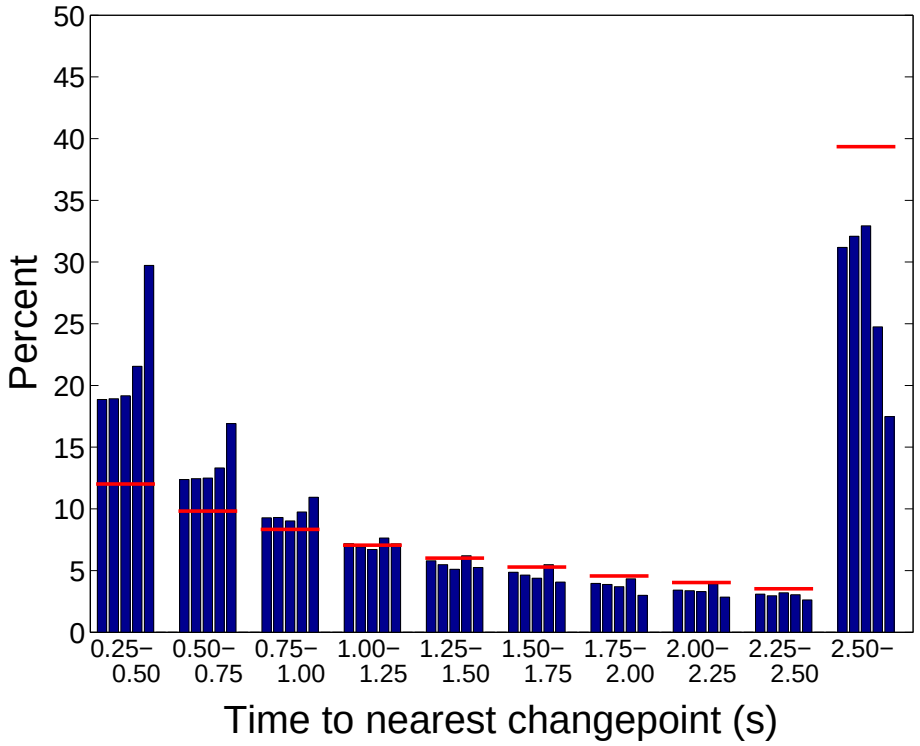

Figure 3.23: SDM condition: Percent of *single speaker speech* DER contained in each distance to the speaker change point bin (shown in blue). The horizontal red line in bin represents the percent of scored single speaker speech time in each bin. Overlapped speech is ignored in this figure in order to get a better visualization of the results for single speaker speech time.

We have shown that the DER increases as the time to the nearest change point decreases, with at least 40% and 33% of each system's DER occurring within 0.50 seconds of the speaker change point for the MDM and SDM conditions, respectively. After excluding errors due to overlapped speech, a considerable amount of the error still occurs within 0.75 seconds of the speaker change point. More specifically, for the MDM and SDM conditions at least 35% and 31% of the DER occurs within 0.75 seconds of a speaker change point for all systems. For the best speaker diarization systems, these amounts are even higher at 49% and 47%.

Although DERs are much worse for shorter segments than longer segments, this trend did not have a large effect on the total DER. This is because less than 3% of the scored speech time occurred during segments less than 1.00 seconds long. Therefore, although the DER for short segments is considerably worse than for long segments, segments less than 1.00 seconds long account for at most 13% and 10% of the total DER for the MDM and SDM conditions, respectively.

Chapter 4

Applying findings to the ICSI speaker diarization system

In the previous chapter, it is shown that state-of-the-art speaker diarization systems perform poorly on short segments and near speaker change points. Now that we have gained insight on *where* systems fail, we dig deeper into one system, the ICSI speaker diarization system, to determine *why* the performance degrades and *how* we can improve results.

Again, the goals of this chapter are to analyze the ICSI speaker diarization system to determine why performance degrades for short segments and near speaker change points; then we apply these findings to improve speaker diarization performance, particularly the speaker error rate. The speaker error rate is the percent of speech time that the hypothesized speaker does not match the reference speaker. Although there is a large amount of error – particularly miss error – due to overlapped speech, overlapped speech will not be investigated in this thesis. Overlapped speech is an area of active research [12, 36, 60]. It plays a large role in the speaker diarization rate, especially since time during overlapped speech is included multiple times in terms of scoring (i.e. if there are two speakers speaking simultaneously for one second, this counts as two seconds of total speech time). Although the speaker error rate contributes to a smaller portion of the total DER, determining the correct speaker is one of the main tasks in speaker diarization. Therefore, there is value in obtaining a deeper understanding of what can be done to improve speaker error rates.

This chapter is arranged as follows: Section 4.1 describes the relevant background information; Sections 4.2, 4.3, and 4.4 analyze preliminary findings on the development set; Section 4.5 provides the test set results; and Section 4.6 discusses the results.

4.1 Background

4.1.1 Baseline diarization system

The baseline system in this work is based on the ICSI speaker diarization system used in the NIST Rich Transcription 2009 (RT-09) evaluation [26]. The ICSI speaker diarization system uses a Hidden Markov Model (HMM) - Gaussian Mixture Model (GMM) agglomerative hierarchical clustering approach [59, 40]. This is a bottom-up approach, which starts with many initial clusters. The clusters are iteratively merged until each cluster represents a hypothesized speaker in the meeting. The algorithm utilizes an HMM where each state (or cluster) is modeled with a GMM.

More specifically, speech activity detection is performed first and the speech regions are initially assigned to k clusters where the number of initial clusters k is greater than the anticipated number of speakers. The speech activity detection is performed using the open source SHoUT toolkit [2]. Using twelve Mel-frequency cepstral coefficients (MFCCs) and the zero crossing rate, along with their derivatives and second derivatives, the audio is parsed into speech and nonspeech segments using an HMM-GMM algorithm.

After speech activity detection is performed, the initial segmentation is created [40]. This segmentation is achieved by first, splitting the speech regions into one second segments. Twelve long-term acoustic features, shown in Table 4.1, are extracted over these segments using Praat [13]. The initialization is performed using GMMs to model the initial clusters, where the number of GMMs is determined using an iterative procedure. The algorithm starts by training one GMM via the EM algorithm on the speech segments. Using 10-fold cross validation (which is, to split the data into 10 subsets; then train the GMM(s) on one subset and compute the log-likelihoods of the other nine; this is repeated ten times such that each of the ten subsets is used for training one time), the log-likelihood when using one GMM is computed. Then the number of GMMs is increased by one and the procedure repeats (again computing the log-likelihood when using a certain number of GMMs). The final number of GMMs used for initialization is determined according to the setup which produces the highest log-likelihood. Finally, each initial segment is assigned to the most probable GMM, which models a particular initial cluster. The number of mixtures g to use for each GMM is determined according to Equations 4.1 and 4.2, where k is the number of initial clusters. More information regarding how these parameters are determined can be found in [40].

$$sec\_per\_gauss = 0.01 \cdot \text{speech time in seconds} + 2.6 \tag{4.1}$$

$$g = \frac{\text{speech time in seconds}}{sec\_per\_gauss \cdot k} \tag{4.2}$$

After the initial segmentation, the GMM parameters are trained and the input stream is re-segmented using the Viterbi (or "hard") Expectation Maximization (EM) algorithm. Note that for segmentation, a minimum duration constraint t_{mindur} of 2.5 seconds of speech

Table 4.1: Twelve long-term acoustic features used in initialization procedure.

Category	Short description
pitch	median pitch
pitch	minimum pitch
pitch	mean pitch tier
pitch	mean pointprocess of the periodicity contour
formants	standard deviation of the 4th formant
formants	minimum 4th formant
formants	mean 4th formant
formants	standard deviation of the 5th formant
formants	minimum 5th formant
formants	mean 5th formant
formants	mean formant dispersion
harmonics	mean harmonics-to-noise ratio

is used to prevent rampant speaker changes [59]. More specifically, each state has a number of substates, which span t_{mindur} seconds and have the same probability density function.

After updating the models of each of the clusters, the next step is to determine which two clusters to merge. This is done using the modified delta Bayesian Information Criterion (ΔBIC) [5], shown in Equation 3.1, which is computed for each pair of clusters. The cluster pair with the largest modified ΔBIC value greater than zero is merged.

Once two clusters are merged, the GMM parameters are re-trained and Viterbi decoding is performed to output the most probable segmentation (with a 2.5 second minimum duration constraint). The merging, re-training, and re-segmentation is repeated iteratively until the stopping criterion is met. After which, a final re-segmentation/re-training step is performed where the minimum duration is reduced to 1.5 seconds. As a final smoothing step, if the speaker immediately preceding and following a short nonspeech segment (less than 0.5 seconds) is the same, the nonspeech segment is relabeled to be a speech segment spoken by the same speaker. This step is later referred to as *gapsmoothing*.

For the SDM condition, 19 Mel-frequency cepstral coefficients (MFCCs) (MFCC-19) are used for modeling the GMMs. For the MDM condition, both MFCC-19 and time delay of arrival (TDOA) features obtained from BeamformIt [9] are used for classification. The MFCC and TDOA feature streams are combined at the log-likelihood level according to Equation 4.3, where x_{MFCC} and x_{TDOA} are MFCC and TDOA feature vectors; $\theta_{i,MFCC}$ and $\theta_{i,TDOA}$ are the GMM parameters for cluster i using MFCC and TDOA features, respectively; and α and β are the mixing weights of the two feature streams. The mixing parameters α and β are 3.0 and 1.6, respectively. These values were determined empirically.

$$p(x_{MFCC}, x_{TDOA}|\theta_i) = \alpha \log p(x_{MFCC}|\theta_{i,MFCC}) + \beta \log p(x_{TDOA}|\theta_{i,TDOA}) \qquad (4.3)$$

4.1.2 Data

The results in this chapter are shown for the NIST Rich Transcription (RT) datasets. The data is split into two partitions: a development set and a test set. The development set consists of 28 recordings from RT evaluations prior to RT-09. The test set consists of 7 recordings from the latest evaluation set, RT-09. The specific meetings are given in Table 2.1. Both the multiple distant microphone (MDM) and single distant microphone (SDM) conditions are investigated.

4.2 Temporal modeling

In the previous chapter, it was shown that a significant amount of errors occur near speaker change points. Two hypotheses for why this occurs are investigated in this section. The first hypothesis is that speakers modify their speech patterns to take the floor or allow the floor to be taken. The second hypothesis is that the minimum duration constraint does not allow the speaker change to happen when it should.

4.2.1 Rearranging the features

In order to determine if speakers modify their speech patterns near speaker change points, the diarization results are re-examined when rearranging the feature vectors. By reordering the feature vectors, we change the speech patterns at the beginning, middle, and end of each segment.

Three configurations of rearranging the feature vectors are investigated. The first method is to resample the feature vectors. Using the final speaker diarization system speaker models and segmentation, we replace the feature vector during hypothesized speech with a sample from the appropriate speaker's distribution. For example, if hypothesized speaker 2 is speaking from 1.0 seconds to 1.5 seconds, replace the feature vectors corresponding to 1.0 to 1.5 seconds with feature vectors obtained by sampling from the speaker 2's GMM.

The second and third methods of rearranging the features involve flipping each speaker-homogeneous speech partition inside out. This is done by first splitting the segments into speaker-homogeneous speech partitions. If there is only one speaker at a time, the speaker-homogeneous speech partitions correspond to the original segments as defined in Section 3.2. However, if there is overlapped speech the segments are broken up into speaker-homogenous partitions. For example, if speaker 1 speaks from 1.0 to 2.0 seconds and speaker 2 speaks from 1.5 to 2.2 seconds, then there are three speaker-homogenous speech partitions: 1.0 to 1.5 seconds, 1.5 to 2.0 seconds, and 2.0 to 2.2 seconds. The second method of rearranging the feature vectors is done by splitting the feature vectors from each speaker-homogenous speech partition into quarters. Then rearrange the feature vectors by first swapping the first and second quarters and then swapping the third and fourth quarters. This method of rearranging the features is referred to as *flipQ*. In Section 3.4.2, it is shown that a significant

source of error occurs within 0.5 seconds of change points. Therefore, the final method of rearranging the features is to swap the feature vectors from the first 0.5 seconds with the rest of the first half feature vectors and similarly swap the last 0.5 seconds with the rest of the feature vectors from the second half of the partition. This is referred to as *flip050*.

The ICSI speaker diarization system described in Section 4.1.1 is then run using the new feature vectors. As shown in Figure 4.1, for the MDM condition, the original diarization system performs best with a DER of 9.58%. The other systems perform worse with DERs of 10.14%, 11.29%, and 11.81% for the resampled, flipQ, and flip050 conditions, respectively. The differences between the reordered feature condition and the original system are all statistically significant. For the SDM condition, the original system has a 19.59% DER. The resampled condition outperformed the original setup with a DER of 18.63%. The difference between the resampled condition and the original is statistically significant. However, the difference between the original setup and the flipQ (19.75%) and flip050 (20.11) conditions are not statistically significant. As shown in Figures 4.2 – 4.7, for the MDM and SDM conditions, the trends of the DER as a function of segment duration and proximity to the speaker change point are strikingly similar. The fact that the trends for both the MDM and SDM conditions did not change after reordering the feature vectors, leads us to believe that speakers do not change their speech patterns at the beginning and end of the speech segments.

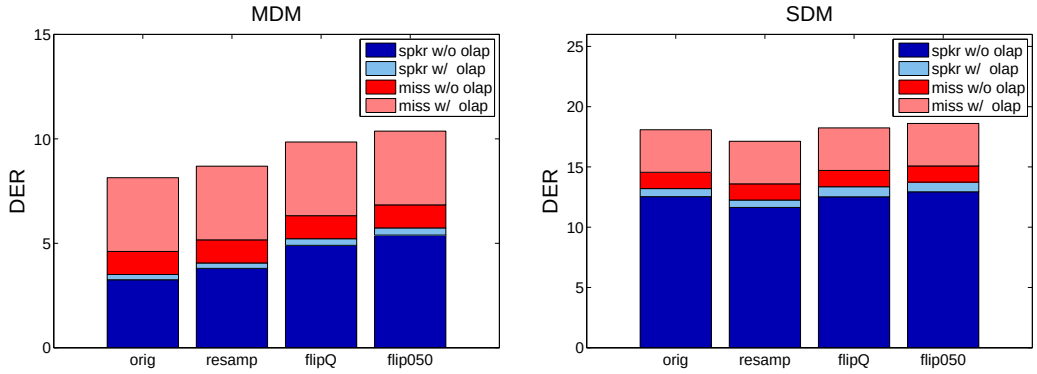

Figure 4.1: A breakdown of the total DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for each of the four feature vector setups: original, resample, flipQ, and flip050. The results are shown for the MDM condition (on the left) and the SDM condition (on the right). Note that the false alarm rate does not have a major impact on the total DER and is therefore not shown (see Section 3.3.3).

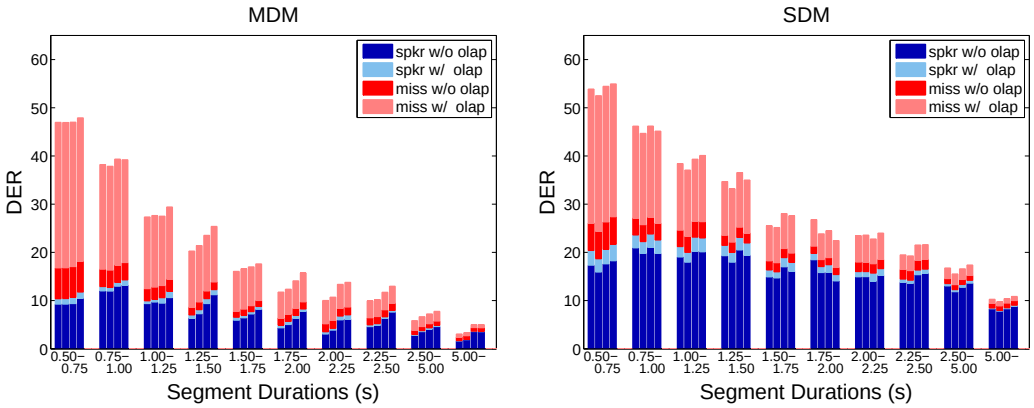

Figure 4.2: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for various sizes of segment durations. Each segment durations bin (e.g., from 0.50-0.75 seconds) contains four bars, representing the original, resampled, joinQ, and join050 conditions from left to right. The results are shown for the MDM condition (on the left) and the SDM condition (on the right).

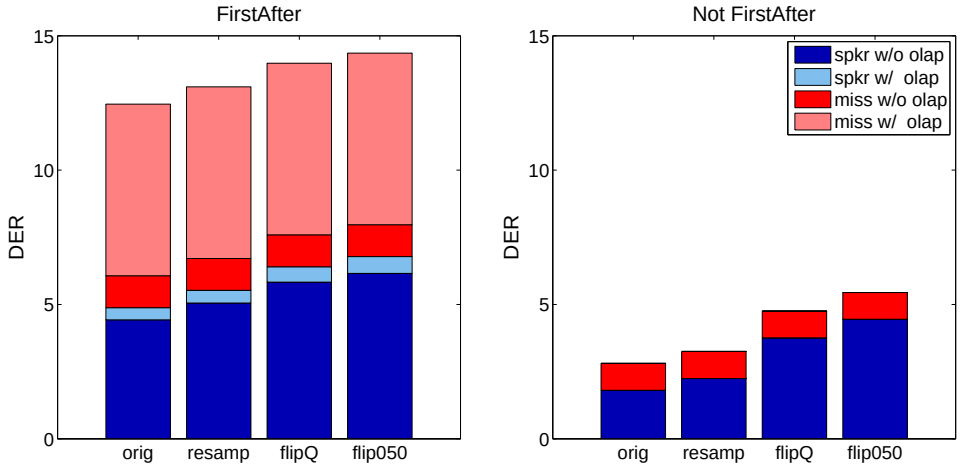

Figure 4.3: MDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for segments immediately following (FirstAfter) and not following (not FirstAfter) a speaker change point.

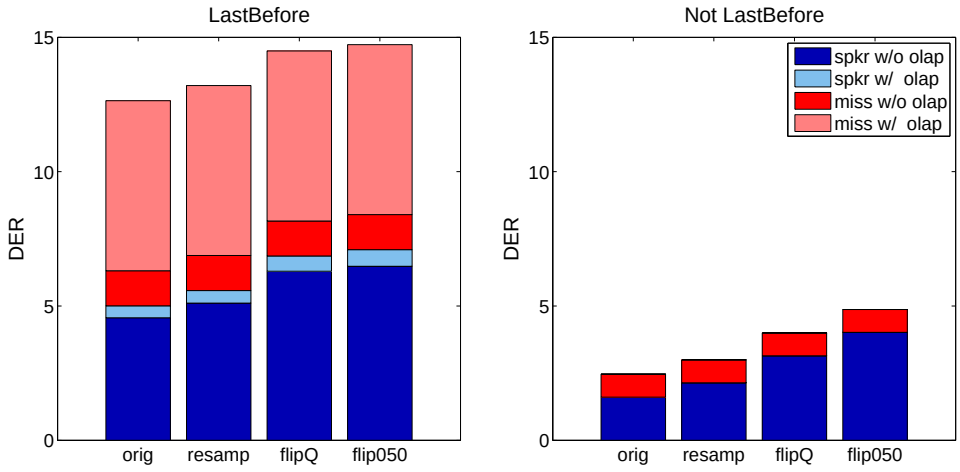

Figure 4.4: MDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for segments immediately preceding (LastBefore) and not preceding (not LastBefore) a speaker change point.

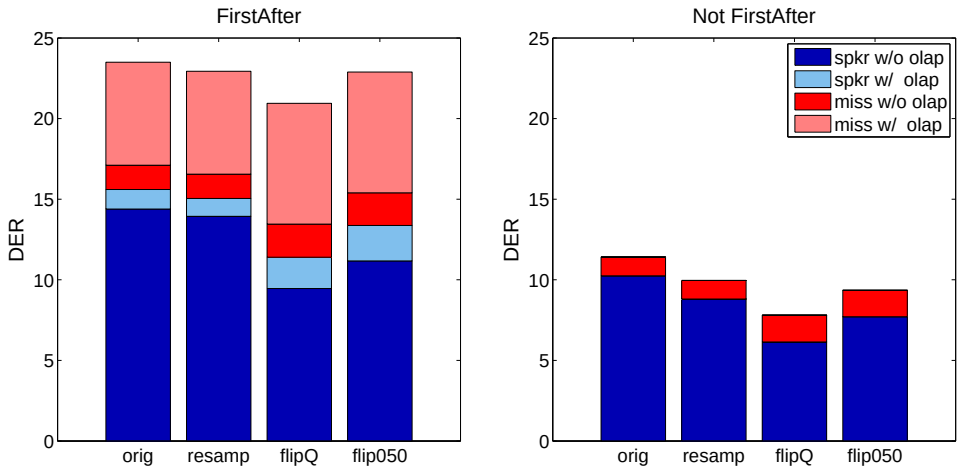

Figure 4.5: SDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for segments immediately following (FirstAfter) and not following (not FirstAfter) a speaker change point.

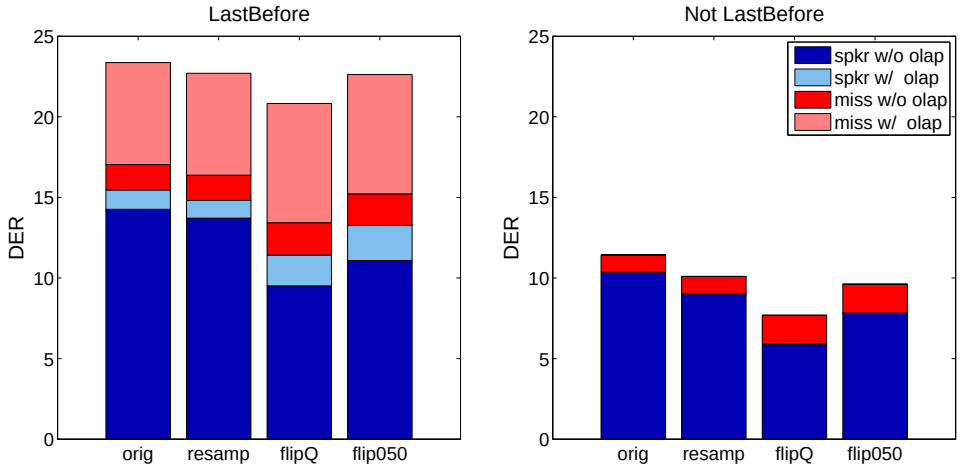

Figure 4.6: SDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for segments immediately preceding (LastBefore) and not preceding (not LastBefore) a speaker change point.

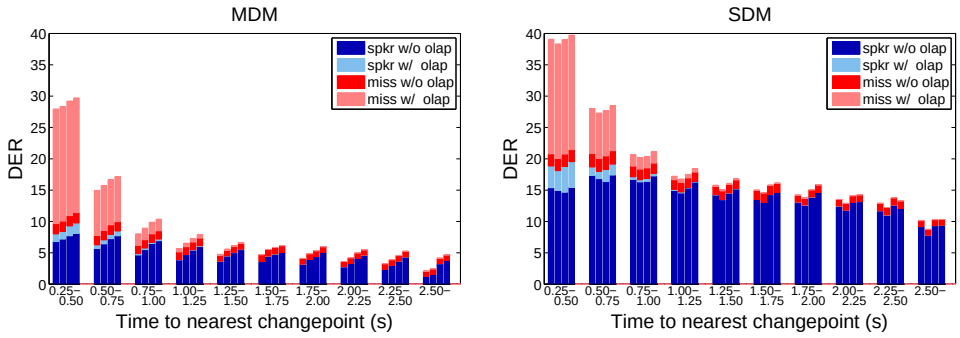

Figure 4.7: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) as a function of the time to the nearest speaker change point. Each distance from the change point bin (e.g., 0.25-0.50 seconds) contains four bars, representing the original, resampled, joinQ, and join050 conditions from left to right. These results are shown for the MDM condition (on the left) and the SDM condition (on the right).

4.2.2 Minimum duration constraint

It is hypothesized that poor performance near speaker change points is a result of the minimum duration constraint. The minimum duration constraint does not allow speaker changes to occur within t_{mindur} seconds of speech. Note that the minimum duration constraint is placed on the speech time (ignoring all nonspeech time).

In this section, the speaker diarization algorithm is rerun using a number of values for the minimum duration constraint. Furthermore, an alternative to the minimum duration constraint is examined, namely median and mean smoothing over the log-likelihoods for each hypothesized speaker. While the minimum duration constraint is useful for eliminating rapid speaker changes, it puts a sharp threshold on the smallest duration a segment can be (often 1.5 to 2.5 seconds of speech [26]). Utilizing a smoothing approach lessens this restriction while still reducing the ability to have rapid speaker changes. Unlike the minimum duration constraint which is enforced over speech time, the smoothing is done over all time (including nonspeech).

The results presented in this section are on the development set and are shown in terms of the speaker error time (T_{SPKR}). Four different approaches are considered to reduce rapid speaker changes. The first approach is to modify the minimum duration constraint used within the algorithm (the original system uses 2.5 seconds) while keeping the minimum duration constraint used in the last iteration at the default 1.5 seconds. The second approach changes the minimum duration constraint used in the last iteration of the algorithm (the original system uses 1.5 seconds) while using the default 2.5 second minimum duration constraint within the algorithm. The third and fourth approaches apply mean and median smoothing (over a varied number of frames) to the final log-likelihoods for each hypothesized speaker. The new segmentation is performed on a per frame basis, where the hypothesized speaker has the highest mean or median smoothed log-likelihood. The experiments are performed for both the MDM and SDM conditions and the results are shown in Tables 4.2 and 4.3, respectively.

As shown in Tables 4.2 and 4.3, determining the speaker via mean smoothing the log-likelihood scores is the best and second best method for the MDM and SDM conditions, respectively. This method results in a 18.5% and 3.2% relative decrease in speaker error rate over the baselines (526.79 seconds and 1985.49 seconds) for the MDM and SDM conditions, respectively. Though the results for the SDM condition are not as dramatic as the MDM condition, the smoothing results are consistently better for shorter smoothing values. The total DERs decreased from 9.58% to 9.01% for the MDM condition and from 19.59% to 19.27% for the SDM condition. Also, for the SDM condition, since results are best when the minimum duration constraint used throughout the algorithm is 3.5 seconds, we examine results when the minimum duration constraint is increased to 4.0 seconds. We find that when the minimum duration constraint is 4.0 seconds, the speaker error time increases to 1979.14 seconds. As an aside, the values annotated with an asterisk contain a higher combined miss and false alarm error rate than the other values in the table. This is due to gapsmoothing

Table 4.2: MDM – Speaker error time (in seconds). Values denoted with * have combined miss and false alarm rates greater than 6.0%.

Min dur or smooth time	Min dur	Min dur last iter	Mean smoothing	Median smoothing
0.0		4598*	1083	1083
0.5	1611	544*	480	529
1.0	914	505	**429**	446
1.5	880	527	456	447
2.0	960	589	503	478
2.5	527	666	568	519
3.0	727	730	625	566
3.5	1102	817*	691	623

Table 4.3: SDM – Speaker error time (in seconds). Values denoted with * have combined miss and false alarm rates greater than 6.4%.

Min dur or smooth time	Min dur	Min dur last iter	Mean smoothing	Median smoothing
0.0		5355*	6303	6303
0.5	7457	2312*	2288	2671
1.0	3949	2069	1945	2089
1.5	2839	1985	**1923**	1991
2.0	2218	2017	1955	2002
2.5	1985	2133	1994	2025
3.0	1939	2109	2039	2043
3.5	**1875**	2118	2095	2081

as described in Chapter 4.1.1.

These results are further analyzed in terms of performance for various segment durations and according to the proximity to speaker change points. We have plotted the results for the MDM condition after mean smoothing over 1.0 seconds in Figures 4.8 – 4.12. These results focus on the single speaker speech errors since the overlapped speech errors are essentially the same, as illustrated in Figure 4.8. As shown in Figure 4.9, regardless of the segment duration the speaker error rates for single speaker speech are always better when using mean log-likelihood smoothing, with the exception of segments between 2.00 and 2.25 seconds long which are marginally worse. Figures 4.10 and 4.11 demonstrate that a far greater amount of the speaker error rate improvement is during FirstAfter and LastBefore segments. More specifically, from Figure 4.12, we see that the speaker error rate is always better for the

mean log-likelihood smoothed MDM system when considering the distance to the nearest change point. The difference is greatest when the distance is closest to the speaker change point. The trend of getting better results as the time to the speaker change point increases still holds after mean log-likelihood smoothing. However, the trend is slightly less dramatic since performance has improved, particularly for those values closest to the speaker change point. The SDM results showed no visible difference between the results when using the standard minimum duration constraint and mean log-likelihood smoothing, and therefore are not shown. This is understandable since the overall DER decreased from 19.59%DER for the original minimum duration constraint setup to 19.27% DER after mean smoothing over 1.5 seconds, which is only a 1.6% relative improvement.

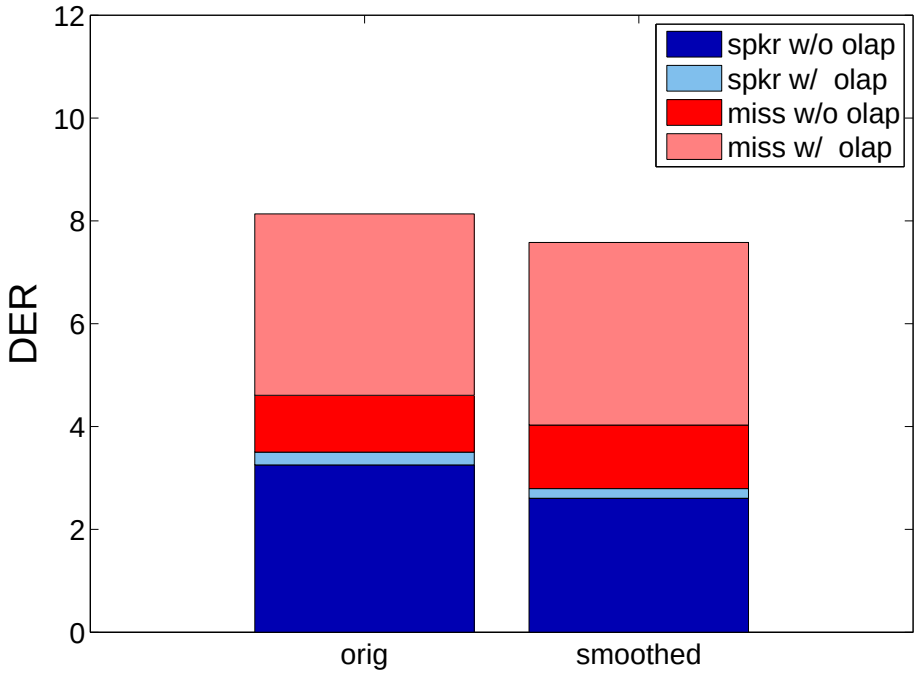

Figure 4.8: MDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for two setups: the original system and after mean smoothing the log-likelihoods over 1.0 seconds.

4.3 Identifying "pure" frames

Cluster purification methods are also investigated in this thesis, where cluster models are trained only the "pure" data, to improve speaker error rates. Since the goal of cluster

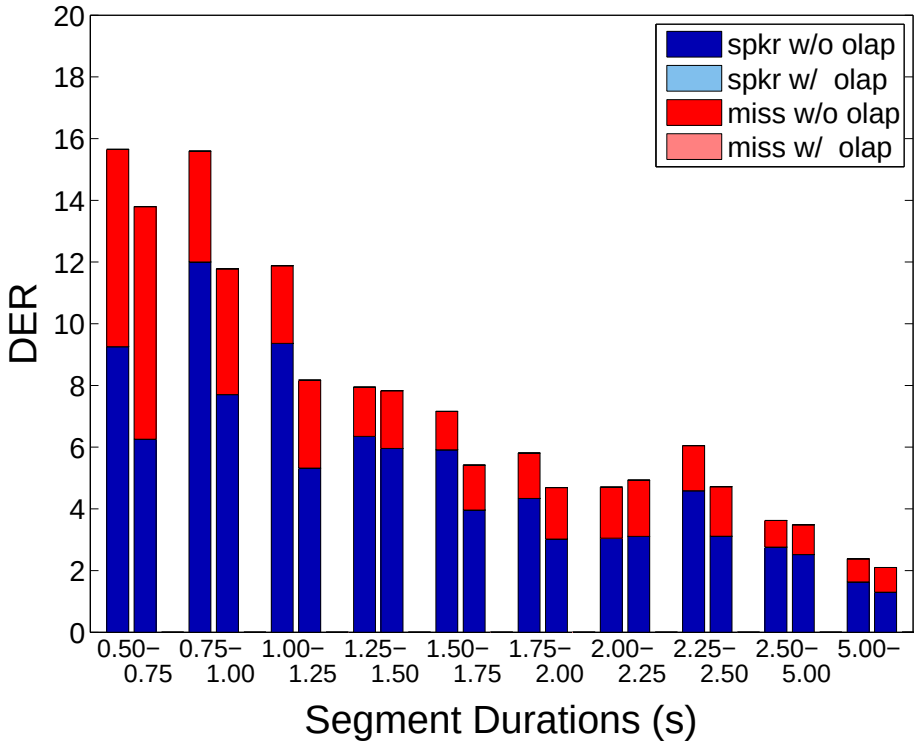

Figure 4.9: MDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for *only* single speaker speech (w/o olap)) for various sizes of segment durations. Each segment durations bin (e.g., from 0.50-0.75 seconds) contains two bars representing the original system and after mean smoothing the log-likelihoods over 1.0 seconds.

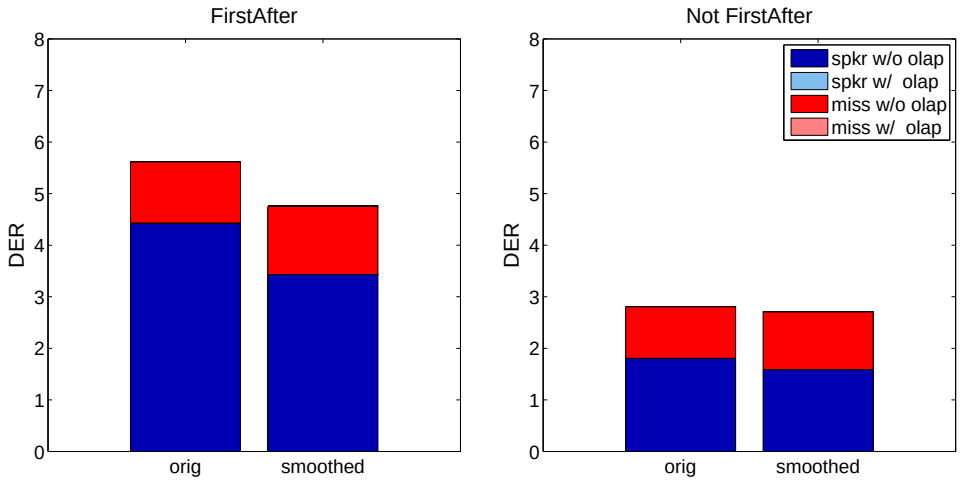

Figure 4.10: MDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for *only* single speaker speech (w/o olap)) for segments immediately following (FirstAfter) and not following (not FirstAfter) a speaker change point.

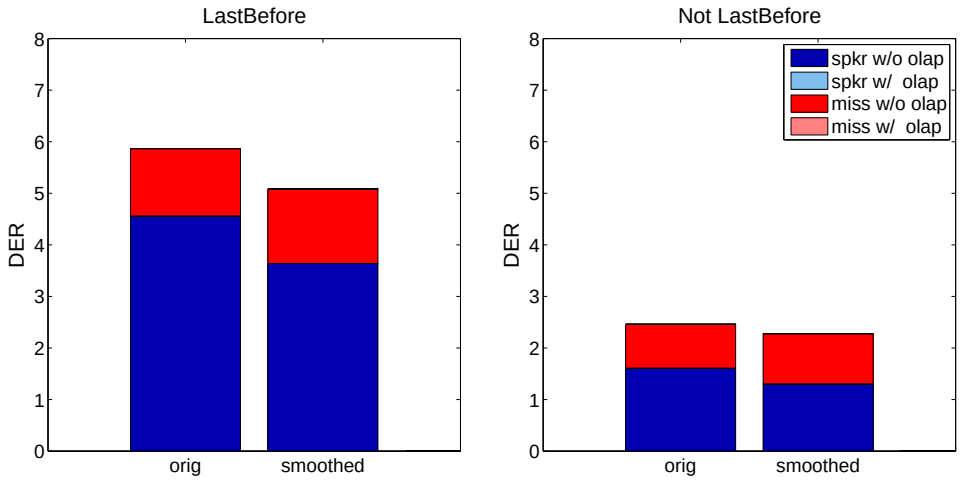

Figure 4.11: MDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for *only* single speaker speech (w/o olap)) for segments immediately preceding (LastBefore) and not preceding (not LastBefore) a speaker change point.

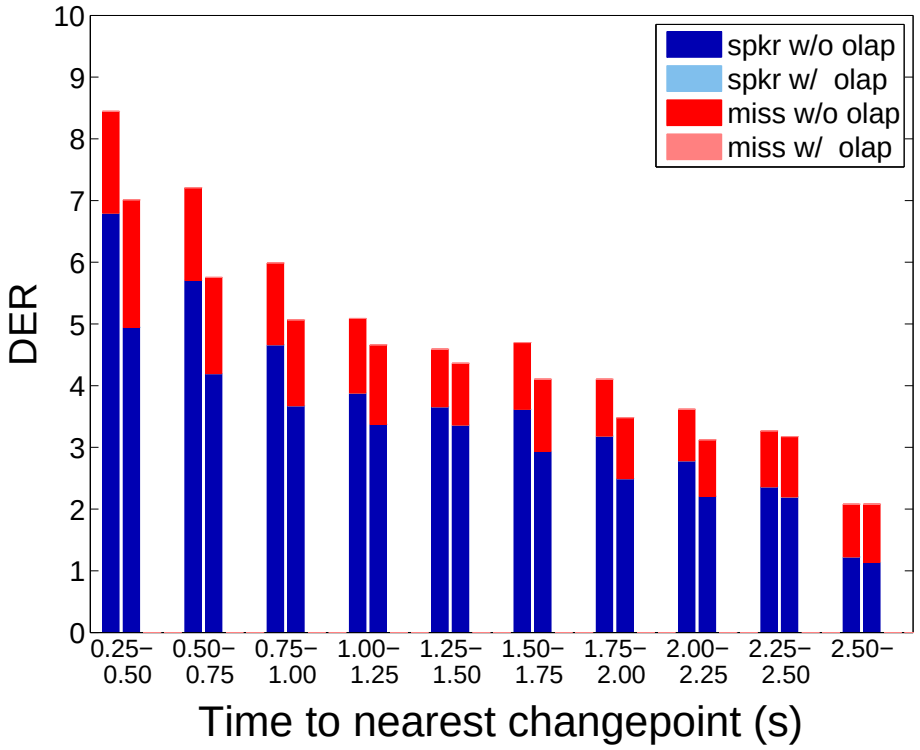

Figure 4.12: MDM condition: A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for *only* single speaker speech (w/o olap)) as a function of the time to the nearest speaker change point. Each distance from the change point bin (e.g., 0.25-0.50 seconds) contains two bars, representing the original system and after mean smoothing the log-likelihoods over 1.0 seconds.

purification is to identify "pure" frames, first five attributes of the log-likelihood scores are evaluated in terms of their effectiveness at separating the correct frames from the incorrect frames. The five attributes of the log-likelihood scores are: maximum, mean, variance, "entropy", and difference between the largest and second largest log-likelihood scores. The log-likelihoods for each of the final clusters are computed and mean smoothed over a number of durations. Then the five attributes summarized below are computed for each frame.

- **Maximum:** The maximum smoothed log-likelihood score.
- **Mean:** The average of the smoothed log-likelihood scores for all of the final clusters.
- **Variance:** The variance of the smoothed log-likelihood scores for all of the final clusters.
- **Entropy:** The "entropy" of the smoothed log-likelihood scores for all of the final clusters. More specifically, let $p(x_t|\theta_k)$ be the probability of the feature vector x at time t given θ_k (the GMM parameters of cluster k). Then the "entropy" of the log-likelihoods is defined as,

$$H(p(x_t|\theta)) = -\sum_{k=1}^{n} p(x_t|\theta_k) \log p(x_t|\theta_k), \tag{4.4}$$

  where n is the number of final clusters.
- **Difference:** The difference between the largest and second largest smoothed log-likelihood scores.

In order to measure the strength of each attribute, the five attributes are thresholded to determine the percent of the correct frames and incorrect frames (in terms of speaker error) that are greater than the threshold. Then for each threshold value, the difference between the percent of correct frames and the percent of incorrect frames greater than the threshold is computed. The attributes which better separate the correct and incorrect classes have a bigger difference between the correct and incorrect percentages. Tables 4.4 and 4.5 show the maximum difference in the percentage of correct and incorrect frames (according to speaker error) which exceed a given threshold of the log-likelihood attribute. Note that the correct and incorrect labels are based on the baseline system results.

From Tables 4.4 and 4.5, we see that for both the MDM and SDM conditions the difference and maximum log-likelihood attributes perform consistently well. In fact, for both MDM and SDM the difference attribute outperforms the other log-likelihood attributes. The maximum attribute performs second best for the SDM condition and third best for the MDM condition. Although, the variance log-likelihood attribute performs well for the MDM condition, it

Table 4.4: MDM condition: Maximum difference between the percentage of correct and incorrect frames which exceed a given threshold for the various mean smoothed log-likelihood attributes. In this table a *larger* value means the attribute is better at separating correct and incorrect frames.

Smooth time (s)	Diff	Var	Max	Entr	Mean
0.0	40.35	21.92	17.46	17.38	1.46
0.5	45.63	23.92	19.53	19.24	2.64
1.0	44.67	23.40	19.41	19.05	2.52
1.5	42.96	22.27	19.30	19.02	2.83
2.0	41.68	21.13	18.10	17.83	3.20
2.5	40.19	20.12	17.12	16.83	3.11
3.0	38.76	19.05	16.72	16.55	3.39
3.5	37.85	18.29	16.00	15.75	3.51

performs worst for the SDM condition. Therefore, the difference and maximum log-likelihood attributes will be investigated further.

Next, the difference and maximum log-likelihood attributes are further analyzed to determine their strength in separating correct frames from incorrect frames, which is useful for performing cluster purification. Note that previous work [47, 15] relies on using the maximum log-likelihood scores to determine which frames should be used for cluster purification. In this experiment, we compute the speaker accuracy for the frames which had the per cluster highest difference or maximum scores. Note that now the hypothesized speaker is the speaker with the greatest smoothed log-likelihood score. Figures 4.13 and 4.14 show the results for all scored time (i.e. not including the no-score "collar" time) for the MDM and SDM conditions. The figures show that for both the MDM and SDM conditions, the speaker accuracy for the best difference scores is better than the speaker accuracy for the best maximum log-likelihood scores (particularly so when looking at the very best scores for each of the two attributes).

4.4 Cluster purification

Cluster purification methods have shown to improve diarization results [47, 15]. In [47], models are first trained according to uniform initialization. Then the data in each cluster is split into 0.5 second segments. The top 25% of the segments in each cluster are labeled and the models for each cluster are re-trained. More segments are iteratively labeled and the models are re-trained until all of the data is labeled and included in the models. Another method of purification is used in [15], where the authors use the top 55% of segments to re-train speaker models. The latter work employs the purification step at the end of the

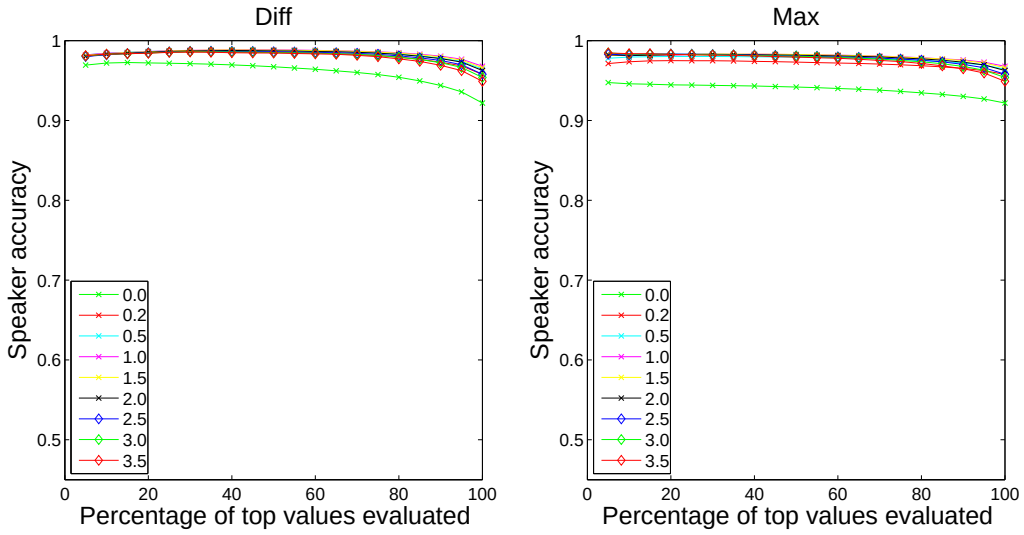

Figure 4.13: MDM condition: Speaker accuracy for the per cluster top x % of difference between the largest and second largest mean smoothed log-likelihood values (on the left) and maximum log-likelihood values (on the right) shown for a variety of smoothing durations as denoted in the legend. When evaluating a very small percentage of the top values, the difference log-likelihood attribute outperforms the maximum.

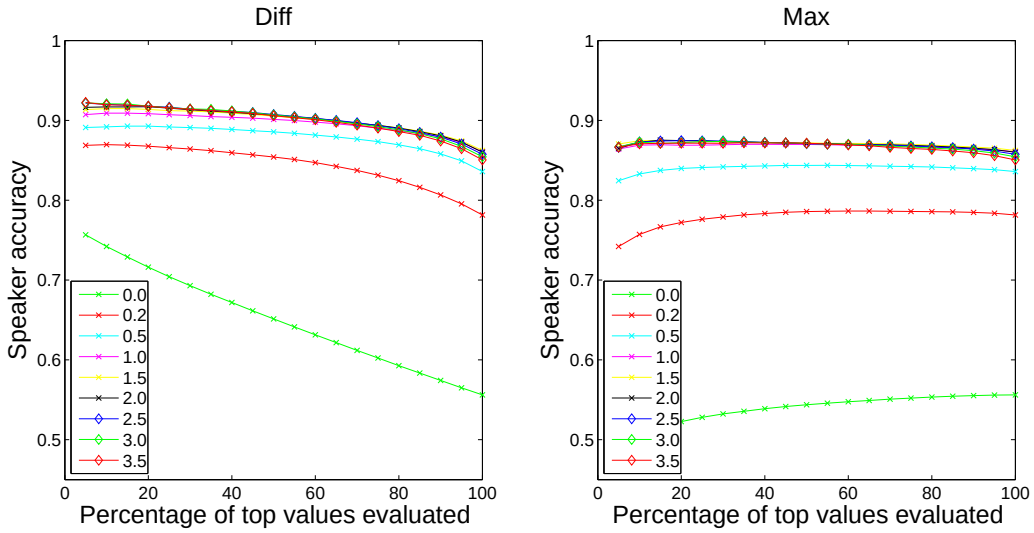

Figure 4.14: SDM condition: Speaker accuracy for the per cluster top x % of difference between the largest and second largest mean smoothed log-likelihood values (on the left) and maximum log-likelihood values (on the right) shown for a variety of smoothing durations as denoted in the legend.

Table 4.5: SDM condition: Maximum difference between the percentage of correct and incorrect frames which exceed a given threshold for the various mean smoothed log-likelihood attributes.

Smooth time (s)	Diff	Var	Max	Entr	Mean
0.0	11.25	0.21	4.23	2.95	1.61
0.5	28.05	3.47	8.17	5.89	4.95
1.0	30.92	4.25	9.75	7.27	5.97
1.5	31.04	4.68	9.69	7.14	6.95
2.0	31.61	4.57	9.73	6.84	7.25
2.5	31.72	4.73	9.63	6.80	7.83
3.0	31.48	4.78	9.68	6.76	8.38
3.5	31.54	4.89	9.84	6.90	8.75

algorithm, as opposed to the former purification method which is performed in the initialization step. Note that the system described in [15] is a top-down speaker diarization system while the system in [47] is a bottom-up system.

In this work, a novel method is utilized to determine which data to use to re-train the models. As opposed to previous work which uses the data that best fits the Gaussian Mixture Model (GMM) (the data associated with the maximum log-likelihoods for each cluster), in this work the models are re-trained on the data with the largest difference in log-likelihoods for the best matched cluster and the second best matched cluster. In other words, it uses data which better matches one cluster over all other clusters.

Based on results from the previous section, the difference between the largest and second largest log-likelihood values is used to determine which frames to use to re-train the cluster models. Log-likelihood values are smoothed over 1.0 seconds and 1.5 seconds for MDM and SDM, respectively. The smoothing values are determined according to the results found in Tables 4.5 and 4.5. Figures 4.15 and 4.16 show the final speaker error rates when using a variable amount of data (according to the difference log-likelihood attribute) to re-train the speaker models at the end and beginning of the speaker diarization algorithm, respectively. For comparison, the results when using the maximum log-likelihood to determine which frames to train the "purified" models on are also shown. Similar to previous purification work [47, 15], each cluster is split into 0.5 second segments. More specifically, the scores are averaged over 0.5 second non-overlapping windows. Also, for the MDM condition only the Mel-Frequency Cepstral Coefficients (MFCCs) are "purified". The GMMs trained on delay features are kept the same. This is because there is not much diversity in the delay feature values. Previous work also does not purify GMMs trained on delay features. Based on the results shown in Figure 4.15, we see that when applying "purification" at the end of the speaker diarization system, re-training on the best per cluster difference between the largest and second largest log-likelihood values results in a better speaker error rate than

re-training on the best per cluster maximum log-likelihood scores. However, decreasing the amount of training data used in the final models and then mean filtering the results performs worse than mean filtering alone. In Figure 4.16, we see that when applying "purification" at the beginning of the speaker diarization system, results are variable and show no real trend. Figures 4.15 and 4.16 show that purification methods do not consistently improve results and therefore, purification methods will not be used.

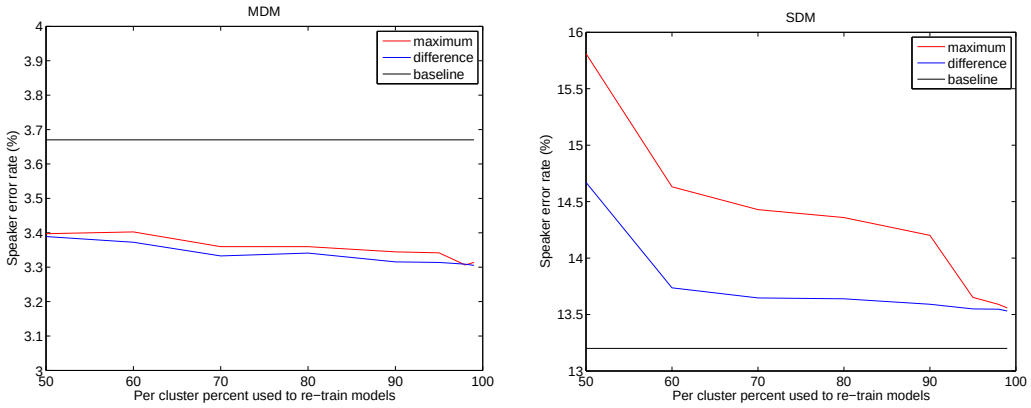

Figure 4.15: Speaker accuracy after re-training models during the final iteration on the top x % of difference between the largest and second largest mean smoothed log-likelihood values (blue) and maximum log-likelihood values (red) for the MDM (left) and SDM (right) conditions.

4.5 Test set results

Since cluster purification is not found to consistently improve results, we simply perform mean filtering at the last iteration. For the MDM condition, the log-likelihoods are mean filtered over a 1.0 second window and for SDM this window is increased to 1.5 seconds. For MDM, the amount of speaker error is reduced from 430.7 seconds to 379.4 seconds, which is an 11.9% relative improvement. This results in a DER of 17.3% and 16.5%, respectively. For the SDM condition, the result is not as dramatic. The speaker error time is reduced from 1086.5 seconds to 1055.7 seconds, or a 2.8% relative improvement, and the DER decreased from 29.2% to 28.6%. In Figures 4.17, 4.18, and 4.19, we show the total DER, DER as a function of the segment duration, and DER as a function of the nearest speaker change point. Similar to the results previously displayed for the development set (shown in Figures 4.9 and 4.12), results for the test set improved most for the shortest segments and the regions in closest proximity to speaker change points.

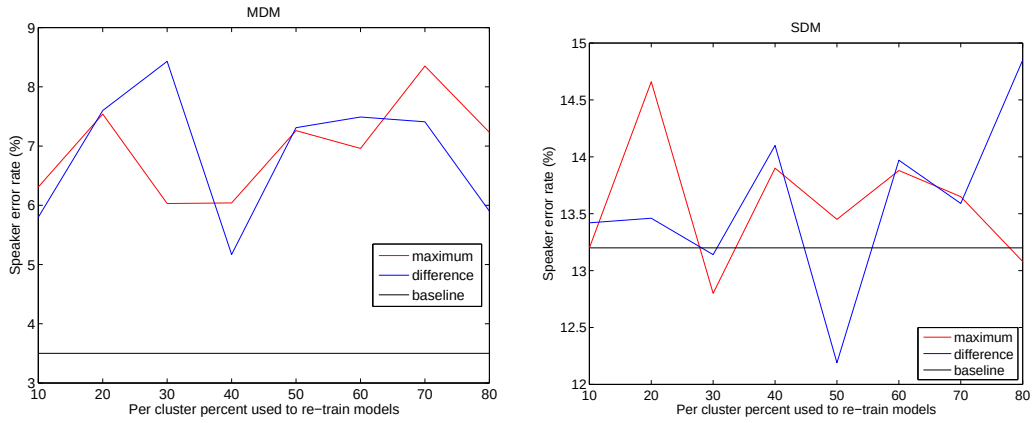

Figure 4.16: Speaker accuracy after re-training models during initialization on the top x % of difference between the largest and second largest mean smoothed log-likelihood values (blue) and maximum log-likelihood values (red) for the MDM (left) and SDM (right) conditions.

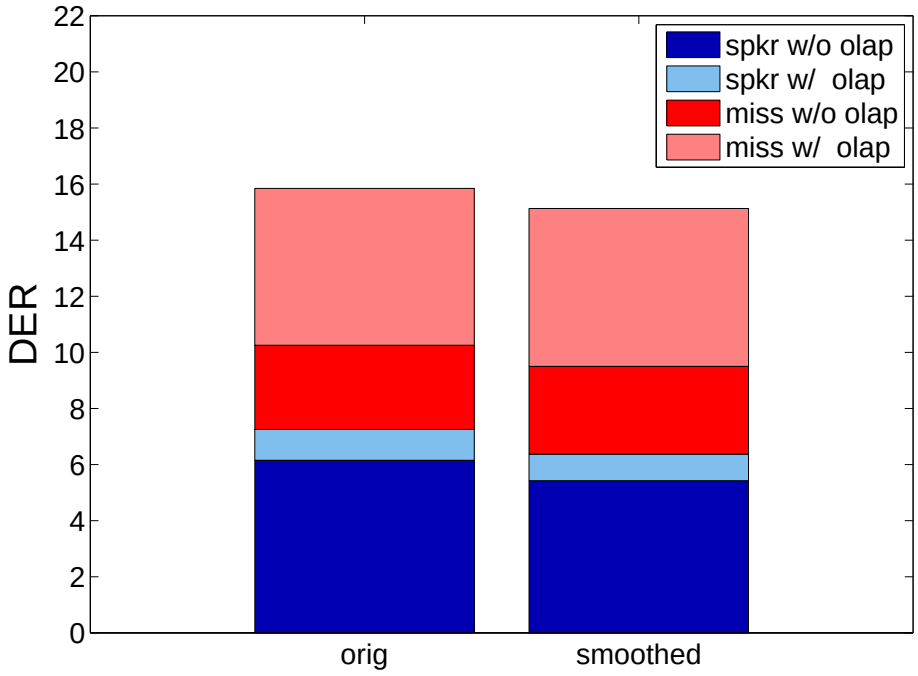

Figure 4.17: MDM condition: Test set results. A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for two setups: the original system and after mean smoothing the log-likelihoods over 1.0 seconds.

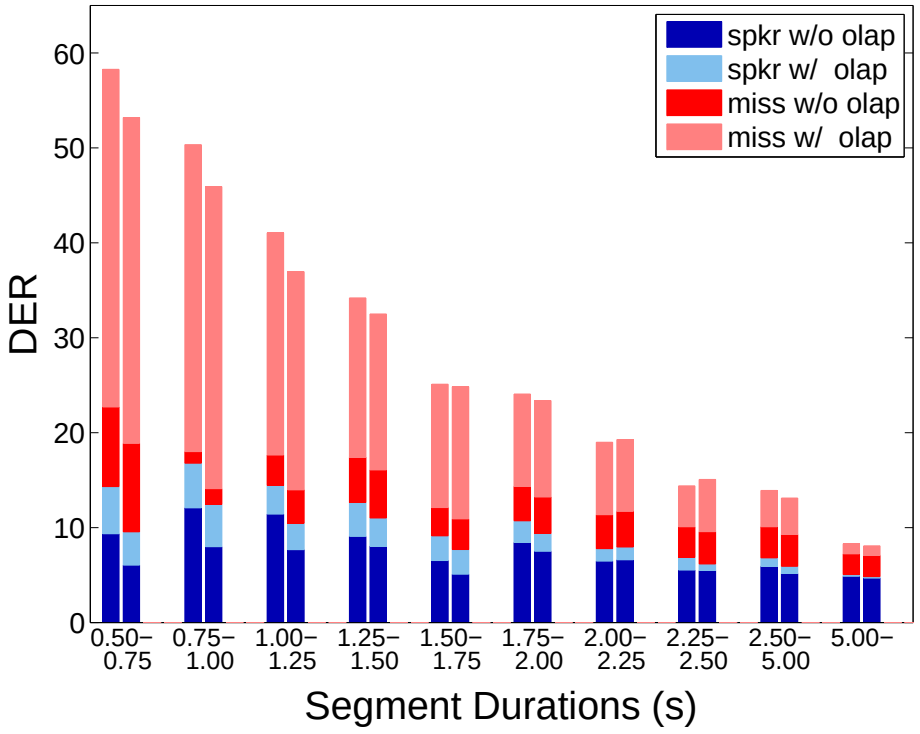

Figure 4.18: MDM condition: Test set results. A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) for various sizes of segment durations. Each segment durations bin (e.g., from 0.50-0.75 seconds) contains two bars representing the original system and after mean smoothing the log-likelihoods over 1.0 seconds.

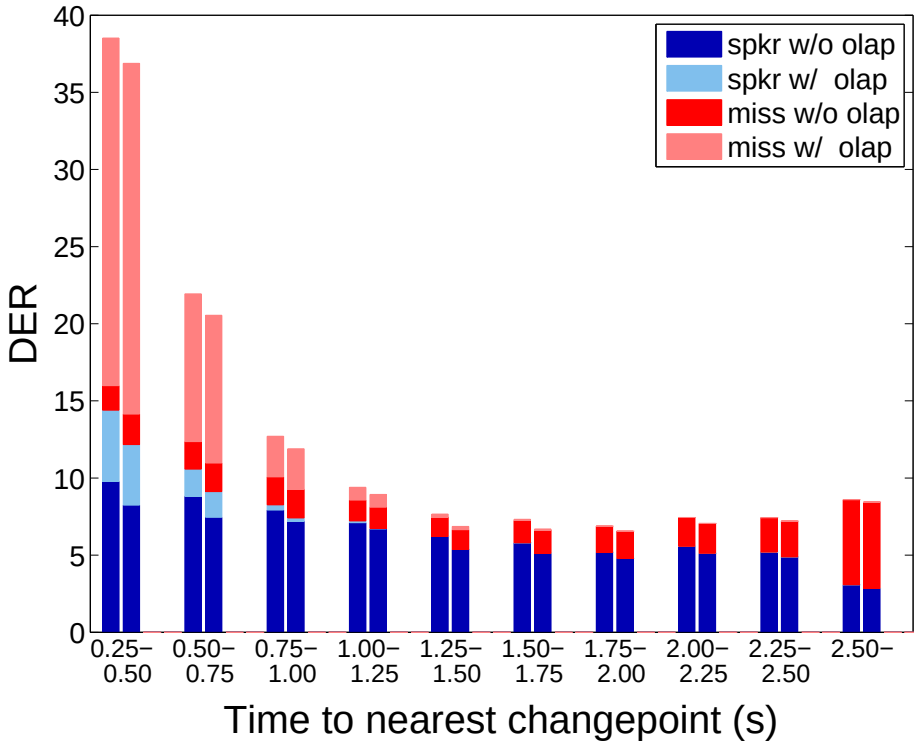

Figure 4.19: MDM condition: Test set results. A breakdown of the DER (in terms of the miss rate (miss) and speaker error rate (spkr), for single speaker speech (w/o olap) and overlapped speech (w/ olap)) as a function of the time to the nearest speaker change point. Each distance from the change point bin (e.g., 0.25-0.50 seconds) contains two bars, representing the original system and after mean smoothing the log-likelihoods over 1.0 seconds.

4.6 Discussion

In summary, we perform a further investigation of the ICSI speaker diarization system. We study a number of scenarios of rearranging the input feature vectors to determine if it has an effect on performance near speaker change points. For all of the experiments the trends all stayed the same (worse performance for short segments and near speaker change points). This leads us to believe that poor performance near change points is not due to speakers changing their speech but that the system is unable to handle the transitions.

We then find that replacing the final minimum duration constraint with mean smoothing the log-likelihood scores for each hypothesized speaker results in an 11.9% relative improvement of the speaker error rate for the MDM condition. More importantly, a significant amount of the improvement occurs near the speaker change point. The results for the SDM condition are less significant, resulting in a 3% relative improvement of the speaker error rate.

We also investigate the usefulness of the difference between the largest and second largest log-likelihood in separating correct and incorrect frames. We find that while the difference log-likelihood attribute performed better than the maximum in terms of identifying correct frames, neither method was useful for performing cluster purification on the first or last iteration. Although purification has been shown to improve results for other systems, purification may be less effective for the ICSI system due to the non-uniform long-term feature based initialization procedure.

Chapter 5

Exploring new domains

Methodologies from the ICSI speaker diarization system have been used previously for other research problems, including speaker localization [27] and audio concept detection [43]. In this chapter, we investigate the use of speaker diarization methodologies for yet another domain of research: duplicate detection.

In this chapter, we investigate a novel method of determining whether a short query is a "duplicate" from a full length reference recording based on acoustic diarization methods. In this setting a duplicate is a recording that has the same content as another recording, though the two files do not necessarily have identical digital representations (due to editing or filtering). Though the goal of speaker diarization differs from that of duplicate detection, we find that diarization is a useful method to segment the data and cluster similar data together. Initially, we test our algorithm on a broadcast news recording dataset under four audio conditions: unmodified, with reverberation, resampled, and lowpass filtered. The algorithm performs well under both the unmodified and reverberation conditions achieving areas under the receiver operating characteristic curve (ROC AUC) values of 0.9. Performance degrades in the resampled and lowpass filtered conditions, achieving ROC AUC values of 0.6. The algorithm is also evaluated on the more widely available TRECVID dataset [52]. The results are better than random leading us to believe the diarization-based system has potential. However, the diarization-based system performs poorly in comparison to a state-of-the-art system.

This chapter is outlined as follows: Section 5.1 provides background information; Section 5.2 describes the duplicate detection system; Section 5.3 provides the experiments and results; and Section 5.4 discusses the results.

5.1 Background

The problem of duplicate detection has a variety of applications, including data deduplication, copyright infringement, and social networking. For example, the problem of data

deduplication (which eliminates duplicate copies of repeated data) has received increasing attention in recent years due to the increasing amount of data taken, stored, and shared. A key aspect of data deduplication is comparing chunks of data to previously stored data and identifying whether there is an appropriate match, or "duplicate". Copyright infringement is another area in which it is important to be able to identify a duplication of copyrighted material.

Searching and identifying similar content is a long standing problem in areas of multimedia research. Similarity detection has been used for recommendation systems (e.g., songs suggestions), searching, and copyright infringement. These tasks have different goals but all measure similarities between items.

A great deal of previous work has focused on searching for perceptually similar content [41, 42]. A review of audio fingerprinting is given in [17]. There exist many techniques in the computer vision community on video copy detection [22] and the NIST TRECVID evaluation [52] has a copy detection evaluation track.

Similarity work has also been done in the music community. In [58], the authors describe the algorithm behind Shazam, a popular commercial application used on mobile devices to recognize music. In [19], the authors aim to identify remixed audio tracks using audio shingles with locality sensitive hashing. Their method identifies remixes based on whether the shingles are similar, thus the remix does not need to have similar spectral content for the entire song. In [16], the authors also investigate duplicate detection for the music setting.

5.2 System description

In this section we describe the system we use to perform data duplication. The system is first given a large number of full length audio recordings. These recordings are referred to as the reference recordings. Then, given a short audio query taken from one of the recordings, the system determines which part of the reference recording the query came from. Typically, the algorithm provides a range of times from which the query likely came.

In order to do this, first, diarization is performed on each of the reference recordings. The diarization algorithm is used to segment each recording and group similar segments together into clusters, where a GMM is trained on each of the clusters. Then diarization is run for each of the queries. The queries are then evaluated to determine if they are in fact a duplicate. In order to determine if a query is a duplicate, the symmetric KL divergence is computed between each cluster from the query and all of the clusters from all of the reference recordings. A small symmetric KL divergence value means the two clusters are very similar, which suggests that at least a portion of the query is likely a duplicate of the reference.

5.2.1 Features

Similar to the speaker diarization system we use Mel-Frequency Cepstral Coefficients (MFCCs) in our duplication detection system. More specifically, we use the first 19 MFCCs, which are computed over a 30 ms window with a 10 ms forward shift. The MFCC features are extracted using the Hidden Markov Model Toolkit (HTK) [1].

5.2.2 Diarization system

The diarization algorithm used in our duplication detection system is based on the ICSI speaker diarization system described in Chapter 4.1.1. As described earlier, the system performs both segmentation and clustering, which are performed iteratively using an agglomerative clustering approach. Segmentation entails identifying the boundaries where audio changes occur (e.g. speaker changes). Clustering is grouping segments which contain similar audio together. Usually, the speaker diarization system first separates the speech and non-speech regions and then subsequently deals only with the speech regions. However, since the goal of this work is to detect duplicates and not speakers, all portions of the recording are used in order to not eliminate any potentially important data. Also, a uniform initialization procedure is used instead of extracting long-term features. The number of mixtures per initial GMM is kept constant at five and the number of initial clusters is determined based on the duration of the query.

5.2.3 Symmetric Kullback-Leibler (KL) divergence

Once the final segmentation of the audio is obtained, the symmetric Kullback-Leibler (KL) divergence is used to quantify the difference between the probability distributions defined by the two clusters (one cluster from the query and one cluster from the original broadcast news recording). The KL divergence is defined as

$$D_{KL}(f(x)\|g(x)) = \int f(x) \log \frac{f(x)}{g(x)} dx, \tag{5.1}$$

where $f(x)$ is the probability distribution of the first cluster and $g(x)$ is the probability distribution of the second cluster. Similarly, we define the symmetric KL divergence as

$$D_{KL,sym}(f(x), g(x)) = D_{KL}(f(x)\|g(x)) + D_{KL}(g(x)\|f(x)). \tag{5.2}$$

The unscented transform based approximation of the KL divergence [30] is utilized. This approximation, used specifically for the case of GMM probability distributions, has been shown to work well for speaker recognition [30] as well as speaker diarization [34]. The unscented transform based approximation is deterministic and subsequently efficient to compute [30].

Let X be a D-dimensional GMM with distribution $f(x) = \sum_{i=1}^{M} w_i N(\mu_i, \sigma_i)$, where M is the number of mixture components, w_i is the mixture weight, μ_i is the mean vector of the ith component, and σ_i is the covariance matrix of the ith component. Then the unscented transform can be used to approximate the expectation of $\log g(x)$ by evaluating Equation (5.3) at a number of sigma points $x_{i,k}$.

$$\begin{aligned} \mathbb{E}[\log g(x)] &= \int f(x) \log g(x) dx \\ &\approx \frac{1}{2D} \sum_{i=1}^{M} w_i \sum_{k=1}^{2D} \log g(x_{i,k}), \end{aligned} \tag{5.3}$$

where

$$\begin{aligned} x_{i,k} &= \mu_i + (\sqrt{D\sigma_i})_k \quad k = 1, ..., D \\ x_{i,D+k} &= \mu_i + (\sqrt{D\sigma_i})_k \quad k = 1, ..., D \end{aligned} \tag{5.4}$$

and $(\sqrt{\sigma})_k$ is the kth column of the matrix square root of σ. In our work, a diagonal covariance matrix is used so Equation 5.4 is further simplified to

$$\begin{aligned} x_{i,k} &= \mu_i + \sqrt{D}\sigma_{i,k} \mathbb{1}_{index=k} \quad k = 1, ..., D \\ x_{i,D+k} &= \mu_i + \sqrt{D}\sigma_{i,k} \mathbb{1}_{index=k} \quad k = 1, ..., D \end{aligned} \tag{5.5}$$

where $\mathbb{1}_{index=k}$ is a D-dimensional vector where the kth index is one and all other values are zero. Equations (5.3)-(5.5) are used to approximate the symmetric KL divergence between GMMs trained on clusters from the queries and GMMs trained on clusters from the reference recordings, which is subsequently used to determine whether the time associated with the cluster from the query is from one of the reference recordings.

5.3 Experimental setup

In this section, we describe the method of scoring and the datasets used to evaluate our duplicate detection system.

5.3.1 Scoring

In order to evaluate the results, the Receiver Operating Characteristic (ROC) Area Under Curve (AUC) value is computed. The ROC is a plot of the true positive rate versus the false positive rate. In order to compute the true positive and false positive rates, we threshold

the symmetric KL divergence between the GMMs trained on clusters from the queries and GMMs trained on clusters from the reference recordings. If the symmetric KL divergence for a given cluster pair, where one cluster is from the query and the other is from the original reference recording, is less than the threshold then the cluster from the query is classified as a duplicate of the original broadcast news recording. Otherwise, the cluster from the query is not a duplicate. Cluster pairs are labeled in the reference as a match if time annotated to a cluster from the query corresponded to time from the reference cluster. Note that the ROC plots were computed such that each query-reference cluster pair had equal weight.

For the TRECVID dataset, we computed the ROC AUC metric in a different manner. Other data duplication algorithms typically output a specific start and stop time from the reference recording associated with a given portion of a query. For the diarization based methodology, clusters from the queries are matched to clusters from the reference. Since the clusters likely contain non-contiguous time segments (or multiple segments throughout the query or reference recording) instead of a single segment with a single start and stop time, it is not possible to easily determine where exactly a specific query came from in terms of the reference recording. In order to compare the diarization based results to other data duplication methods, we simplify the task. Now, instead of matching the query to a specific portion of the reference recording, the goal is to determine which (if any) reference file a query is from.

5.3.2 Datasets

Broadcast news

The results are evaluated on approximately 6.5 hours of broadcast news video recordings, consisting of thirteen 30 minute recordings (which included commercials in addition to the news program). Though both video and audio were available, in this work we focus only on the audio.

In order to explore how the system works for a variety of audio queries, the system is evaluated using queries of variable duration and under different audio conditions. More specifically, 15, 30, and 60 second queries are extracted at regular intervals. The query midpoints were every 100 seconds with the first midpoint at 100 seconds and the last midpoint at 1600 seconds. We also investigated performance when the audio was unmodified, lowpass filtered with a 1750 Hz cutoff, downsampled from 44.1 kHz to 8kHz, and included reverberation. We use `sox` [3] to modify the audio recordings. More specifically, for the reverberation setting we use a 75% gain and a 75 ms delay.

The broadcast news recordings are split into a development set and test set. The development set consists of eight recordings and the test set consists of five records, resulting in a total of 1536 and 960 queries respectively. In Table 5.1, we show the breakdown of the development and test sets which were randomly chosen. The names given to each recording include the year, month, day, start time, end time, and network the program aired on.

Table 5.1: Development and test set broadcast news recordings.

Development	Test
19980513-1130-1200-CNN	19980515-1130-1200-CNN
19980513-1830-1900-ABC	19980518-1130-1200-CNN
19980518-1830-1900-ABC	19980519-1830-1900-ABC
19980519-1130-1200-CNN	19980520-1830-1900-ABC
19980520-1130-1200-CNN	19980522-1830-1900-ABC
19980523-1130-1200-CNN	
19980523-1830-1900-ABC	
19980524-1130-1200-CNN	

TRECVID

Performance is also investigated for the TRECVID 2011 content-based multimedia copy detection dataset [52]. This dataset contains video recordings similar to those commonly seen on video-sharing websites. Since the focus of this work is audio, only audio recording information is used to perform the duplicate detection. This dataset contains 201 queries under seven acoustic transforms. The acoustic transforms are original (1), mp3 compression (2), mp3 compression and multiband companding (3), bandwidth limit and single-band companding (4), audio mixed with speech (5), audio mixed with speech and multiband compressed (6), and bandpass filtered audio mixed with speech and compressed (7). Note that the transformations are represented in the figures according to the order it is presented in the previous sentence. There are over 11,000 reference recordings from which some of the queries are extracted. Note that some of the queries contain audio that is not from any of the reference recordings. More specifically, there are 3 types of queries. *Query type 1* is extracted from a single recording and that recording is one of the reference recordings. *Query type 2* contains 2 recordings. One recording (e.g. query A) is place in the middle of the second recording (e.g. query B). In other words, the query is arranged as follows: first half of query B, query A, second half of query B. In this setting, query A is in the reference and query B is not in the reference. *Query type 3* is taken from a single recording which is not in the reference. One third of the 201 queries are taken from each of the query types.

5.3.3 Results

In this section we describe results on both the broadcast news and TRECVID datasets.

5.3.4 Broadcast news

Development set results

First parameter settings are tuned for the diarization system, specifically the number of initial clusters k and minimum duration constraints t_{mindur}. These were the only parameters tuned for the duplicate detection system.

We first investigated the number of initial clusters k used to run the diarization system on the original broadcast news recordings. Experiments were run using 16, 32, 64, 128, and 256 initial clusters for the original 30-minute broadcast news recordings. Empirically, we found that 128 clusters performed best since it resulted in a number of clusters most similar to the number of speakers in the recording.

Next, we investigated performance for a number of minimum duration constraint values. We also varied the number of initial clusters for the queries. We ran the diarization system with 128 initial clusters and a number of minimum duration values (1, 1.5, 2, and 2.5 seconds) on the original broadcast news recordings. For each of the queries we used $k = 1, 2, ..., 8$ initial clusters and $t_{mindur} = 1, 1.5, 2, 2.5$ seconds minimum durations. We only computed the symmetric KL divergence between the GMMs from the queries and original broadcast news recordings that had the same minimum duration. We evaluated the results on the four audio settings (unmodified, lowpass filtered with a 1750Hz cutoff, downsampled from 44.1 kHz to 8kHz, and reverberation) and for the various duration queries (15, 30, and 60 seconds).

We found that a minimum duration $t_{mindur} = 2.5$ seconds worked best for the unmodified and reverberation setting while $t_{mindur} = 1.5$ seconds worked best for the resampled and lowpass filtered settings. Though the variances of the ROC AUC values were small for all of the settings, the results for the resampled and lowpass filtered settings had less variance so we set the minimum duration to 2.5 seconds. Based upon the development set results for the queries as well as the previous conclusion to use 128 initial clusters for the original 30-minute broadcast news recordings, we set the number of initial clusters to $k = \text{round}(\text{query duration in seconds}/14.0625 + 1)$, where 14.0625 was chosen since it is equal to $128/(30 \cdot 60)$. Thus, for the 15, 30, and 60 second queries we started with 2, 3, and 5 clusters respectively. Though again, we found that the variance of the ROC AUC values was very small when varying the number of initial clusters. Having small variance in the ROC AUC values when using a number of initial clusters and minimum durations is promising since the results are not too different based on the parameter selection. Figure 5.1 shows the results on the development set for all of the audio conditions using the diarization parameters settled upon in this section. The numbers included in the legend are the ROC AUC values for the respective settings. The unmodified and reverberation audio conditions perform well for the diarization based system with ROC AUC scores greater than 0.9. The resampled and lowpass filtered audio conditions perform much worse with ROC AUC scores slightly larger than 0.6. Although the performance degrades for the resampled and lowpass

filtered audio conditions, performance still exceeds random guessing which corresponds to an ROC AUC score of 0.5.

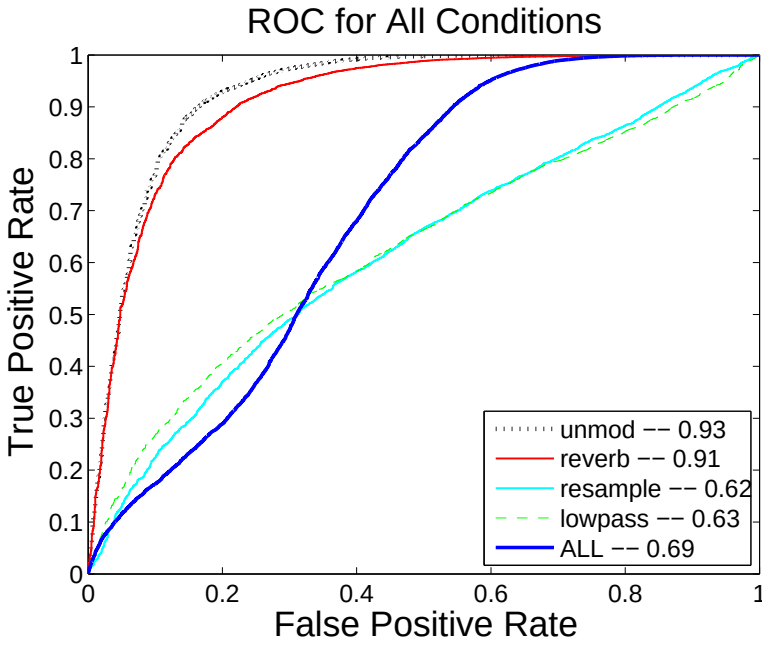

Figure 5.1: ROC plot for all audio conditions (unmodified, contains reverberation, resampled, and lowpass filtered) on the broadcast news development set.

Test set results

Using the parameters determined from the development set results, namely a 2.5 second minimum duration and 2, 3, and 5 initial clusters for the 15, 30, and 60 second queries respectively, we evaluated the test set queries. We compared the GMMs trained on each cluster from the test set queries to the GMMs trained on all of the clusters from the five original 30-minute broadcast news recordings which make up the test set. The ROC plots for the unmodified, lowpass filtered, downsampled, and reverberation settings are shown in Figure 5.2. Each plot shows the results for the 15, 30, and 60 second queries as well as the result when the all of the queries are included. We also included all of the audio conditions into a single ROC plot shown in Figure 5.3. The results are in line with the results from the development set. Again, the unmodified and reverberation audio conditions perform with ROC AUC scores near 0.9. The resampled and lowpass filtered audio conditions again perform much worse than the unmodified and reverberation conditions with ROC AUC scores slightly larger than 0.6.

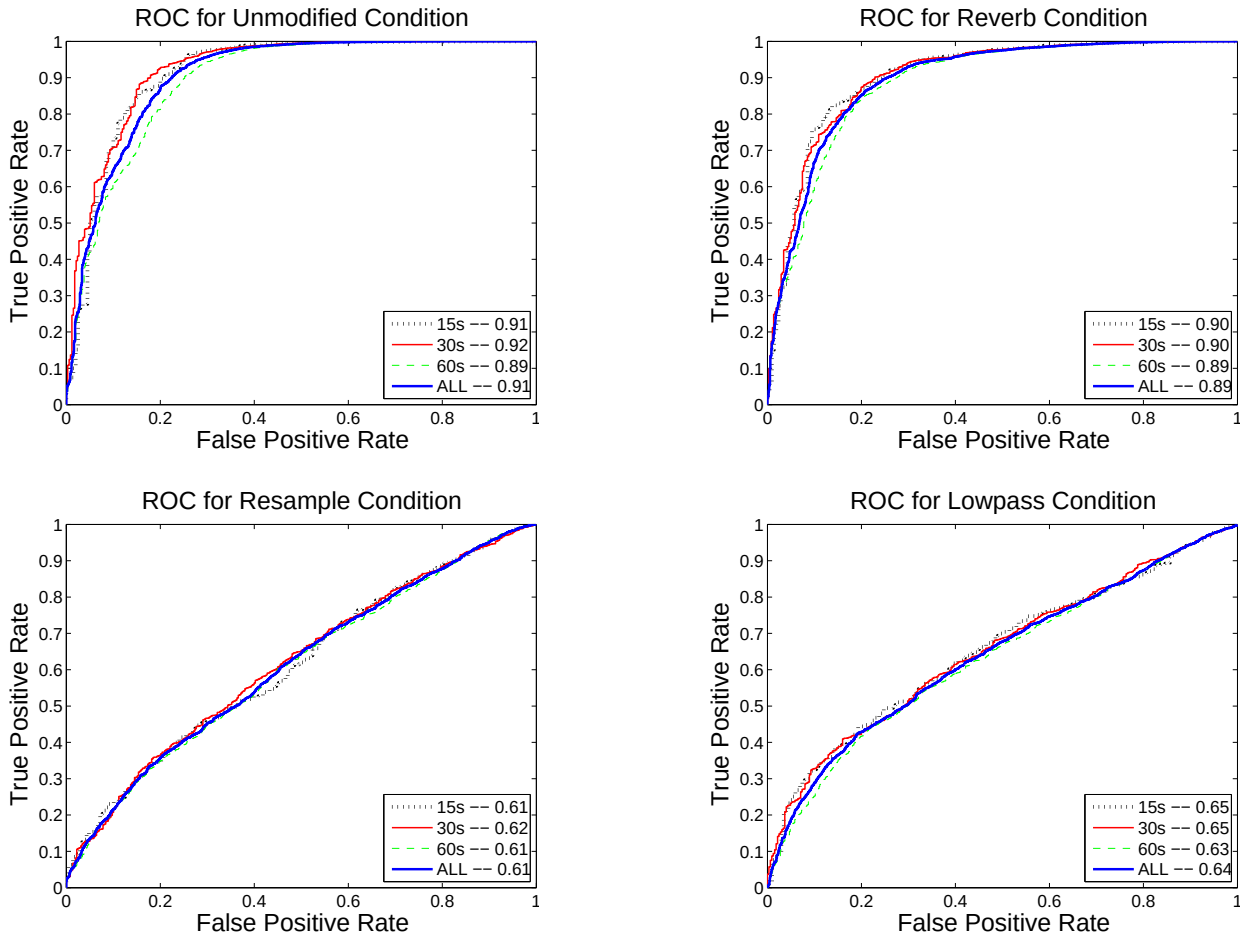

Figure 5.2: ROC plots when the broadcast news test set query audio is unmodified, contains reverberation, resampled, and lowpass filtered.

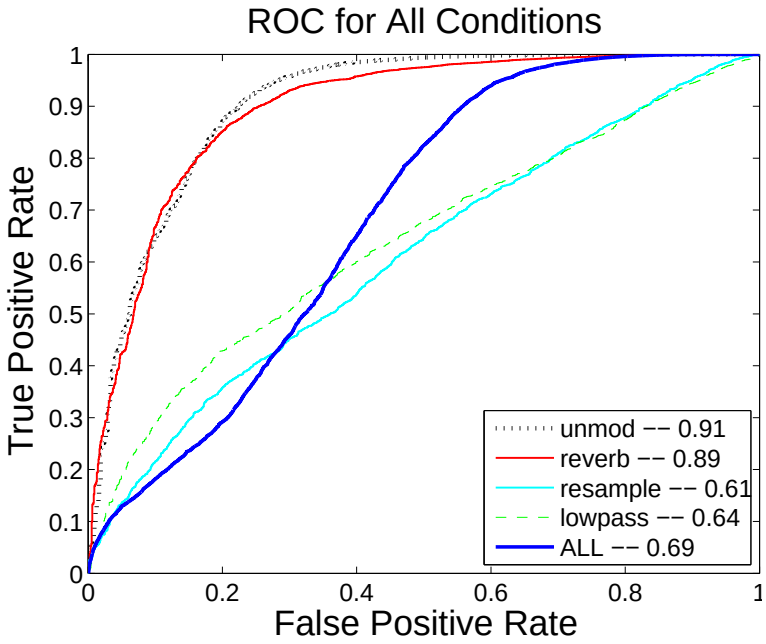

Figure 5.3: ROC plot for all audio conditions (unmodified, contains reverberation, resampled, and lowpass filtered) on the broadcast news test set.

5.3.5 TRECVID

The results are presented for the TRECVID data. The parameters for the diarization system are the same ones as used for the broadcast news test set. Results are shown for all of the queries, as well as for only type 1 queries and only type 2 queries. Results are not shown for only type 3 queries since there are no true positive results since type 3 queries do not match any of the reference recordings.

In order to compare results with a state-of-the-art content-based multimedia copy detection system, the scoring method is simplified as described in Section 5.3.1. Previously, the KL divergence is computed for each query-reference cluster pair. Since there are a variable number of clusters for each reference video, it is difficult to compare the results from the diarization-based system to another system. Instead of considering all query-reference cluster pairs, the scores for each query-reference cluster pair are averaged for each reference recording. Therefore, a query-reference recording is a match if the query is from the reference recording. The result using this setup is shown in Figure 5.6. Then, we compare results with the Telefonica Research audio-only system [7]. A brief description of the Telefonica system is given below.

The audio-only Telefonica Research content-based copy detection system is performed using Masked Audio Spectral Keypoint (MASK) [9] features. Like the Shazam [58] and

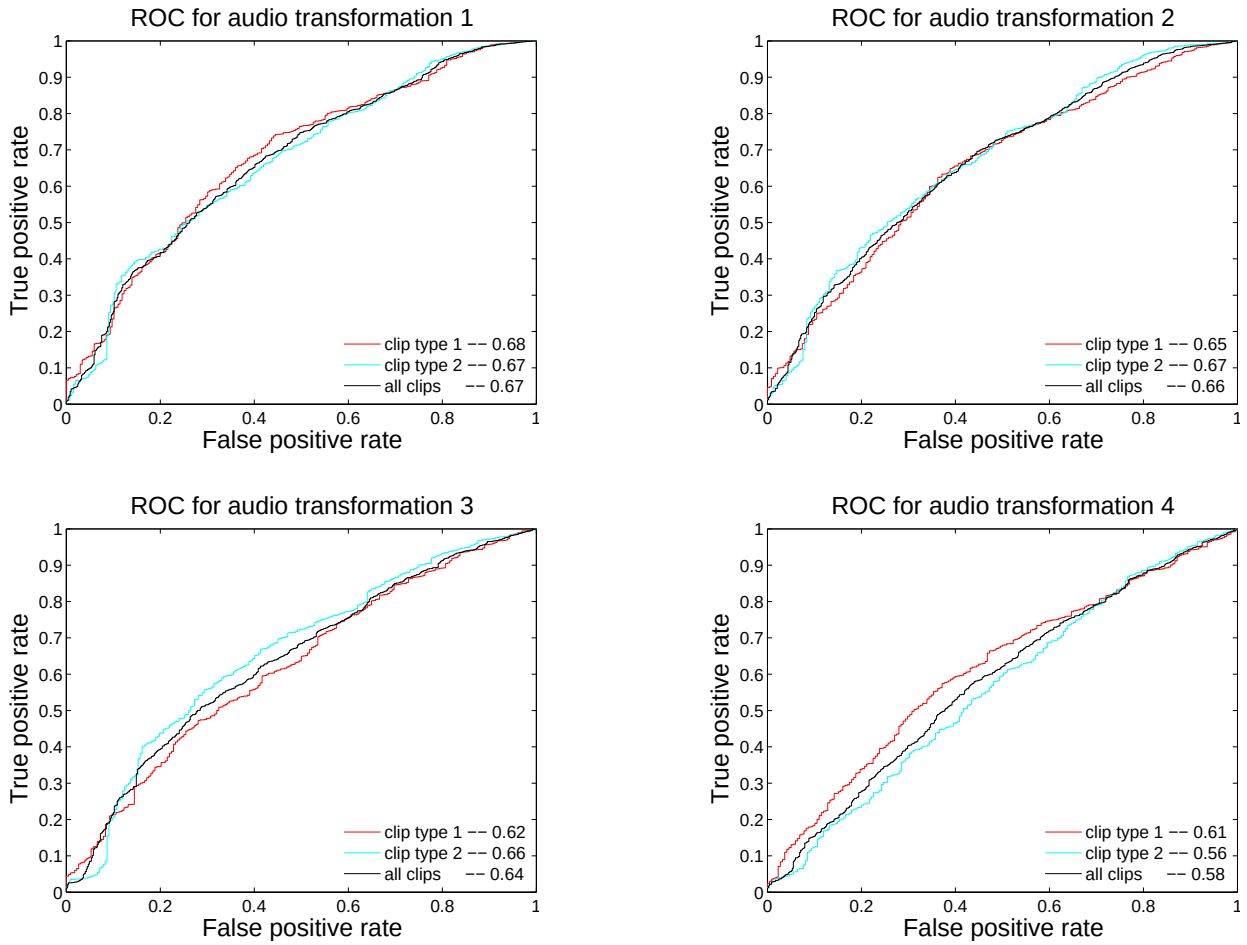

Figure 5.4: ROC plots for the TRECVID dataset for the first 4 audio transformations (original (1), mp3 compression (2), mp3 compression and multiband companding (3), bandwidth limit and single-band companding (4)). Each plot shows the ROC for audio queries of type 1, type 2, and all audio queries.

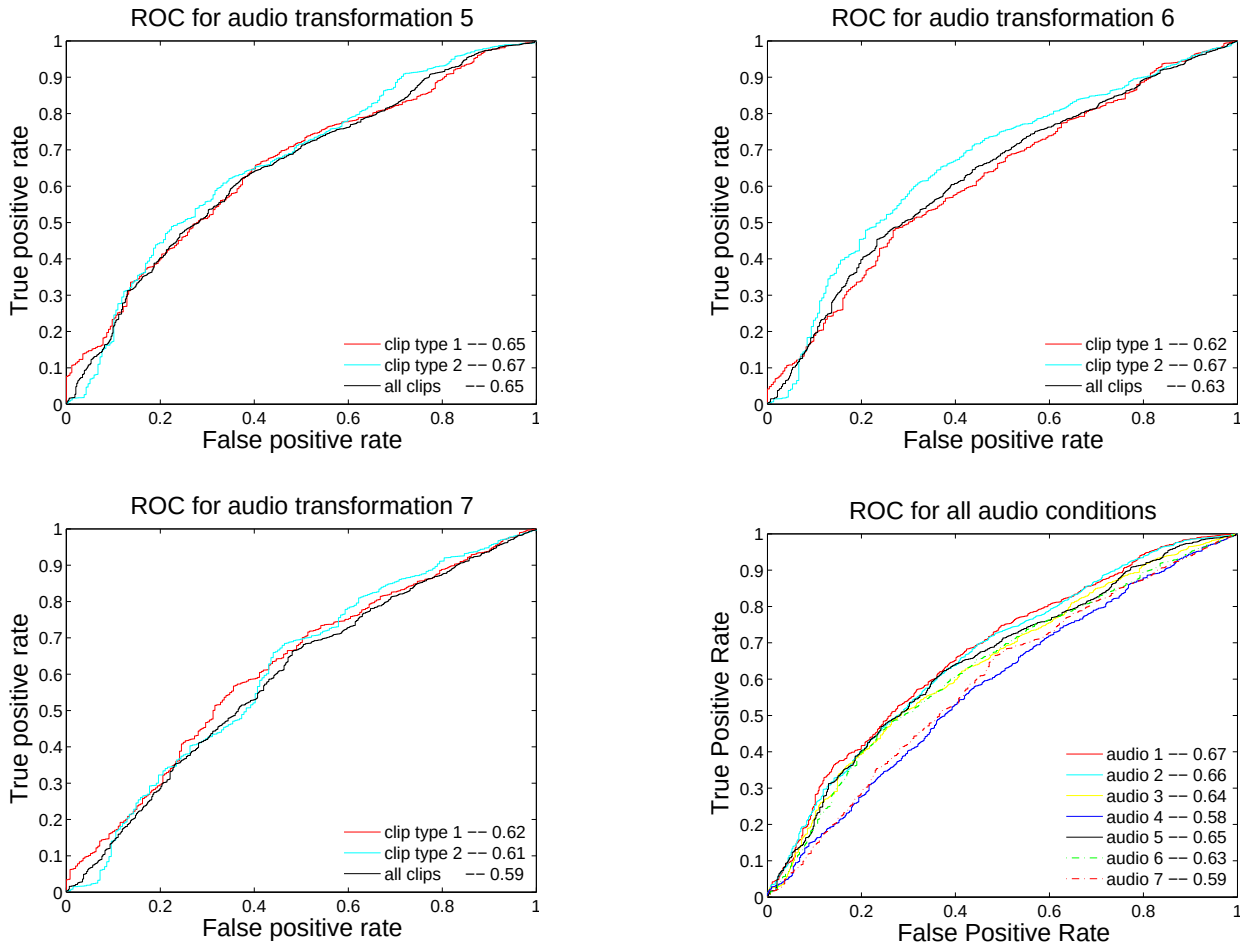

Figure 5.5: ROC plots for the TRECVID dataset for audio transformations 5 – 7 (audio mixed with speech (5), audio mixed with speech and multiband compressed (6), and bandpass filtered audio mixed with speech and compressed (7)) and all audio transformations. Each plot shows the ROC for audio queries of type 1, type 2, and all audio queries.

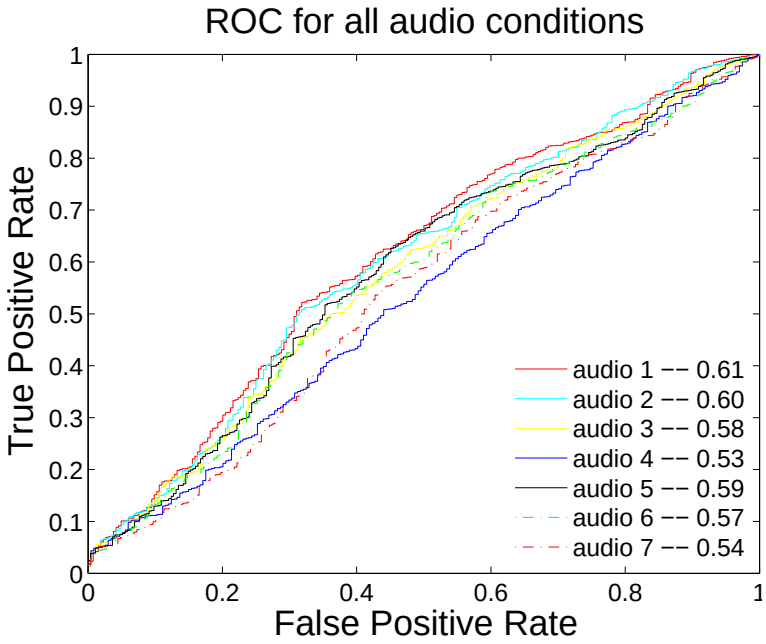

Figure 5.6: ROC plot for the TRECVID dataset for all audio transformations 1 – 7 when averaging scores for all clusters in the reference recording.

Philips [32] fingerprints, MASK fingerprints encode spectrogram local maxima information. More specifically, the Fast Fourier Transform is computed over 100ms of downsampled and bandpass filtered audio every 10ms. Then using the mel-frequency scale, the mel-spectrogram is computed. The features are computed at time-frequency peaks of the mel-spectrogram. The MASK fingerprint encodes the mel spectral band location of the peak as well as binary values describing differences in average energies for relative time-spectral region pairs. The fingerprints are extracted for the reference recordings and the queries. Then a histogram is created of the time difference between matching MASK fingerprints from the query and reference. Peaks in the histogram suggest that there is a match for the query in the reference. The Telefonica system outputs the top 20 matches for each query and a corresponding score.

The ROC curve for the Telefonica system and the diarization based system when only including the scores for the top 20 reference queries are shown in in Figure 5.7. The first observation is that the Telefonica system is really good with almost 100% accuracy. Also, for the majority of the audio conditions the diarization-based system never hypothesizes the correct reference recording. Similar plots for the diarization-based system when including more top scoring reference recordings are shown in Figure 5.8 and the results when including all of the reference recordings are shown in Figure 5.6.

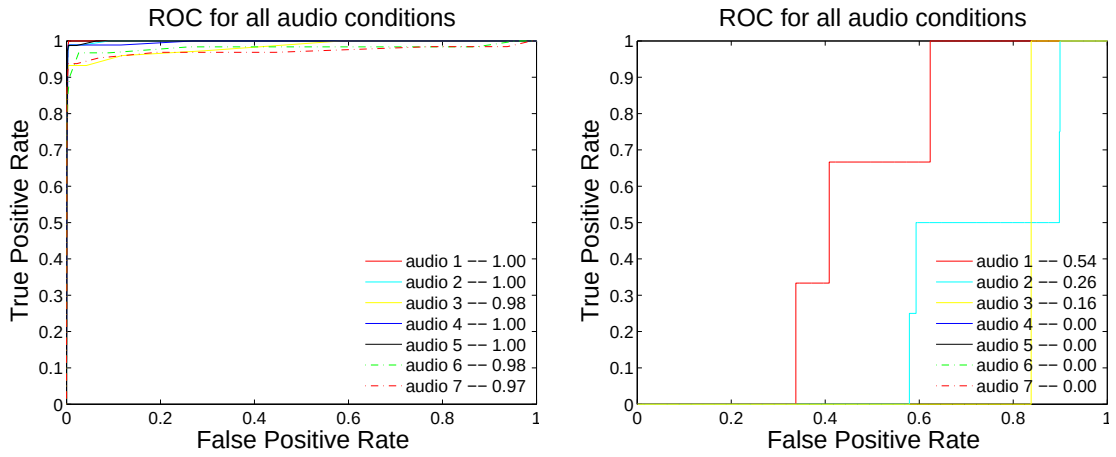

Figure 5.7: ROC plot for the TRECVID dataset for all audio transformations 1 – 7 when evaluating the top 20 best matched reference recordings for the Telefonica system (on the left) and the diarization-based system (on the right).

5.4 Discussion

We introduce a novel method utilizing diarization for identifying duplicate queries. The diarization system is used to split both the original reference recordings as well as the queries into homogeneous clusters. Then the symmetric KL divergence is used to determine whether the time annotated to the cluster from the query is a duplicate of the reference recording.

There are two tunable parameters in the diarization system, the number of initial gaussians k and the minimum duration constraint t_{mindur}. The results obtained on the development set are not very sensitive to the parameter settings. However, we settle on using a minimum duration of 2.5 seconds and $k = \text{round}(\text{query duration in seconds}/14.0625 + 1)$ initial clusters.

We test our method on a variety of queries. For the broadcast news dataset, our test set includes 15, 30, and 60 second queries and four audio conditions: unmodified, with reverberation, resampled, and lowpass filtered. We find that performance is best under the unmodified and reverberation conditions, achieving ROC AUC values of 0.9. Performance degrades under the resampled and lowpass filtered condition, however we are still able to achieve ROC AUC values of 0.6.

For the TRECVID dataset, we evaluate the results on seven audio conditions: unmodified, mp3 compression, mp3 compression and multiband companding, bandwidth limit and single-band companding, audio mixed with speech, audio mixed with speech and multiband compressed, and bandpass filtered audio mixed with speech and compressed. The best results for the TRECVID dataset are not as good as those from the broadcast news dataset. ROC AUC values are typically above 0.6. Results from the diarization-based system are

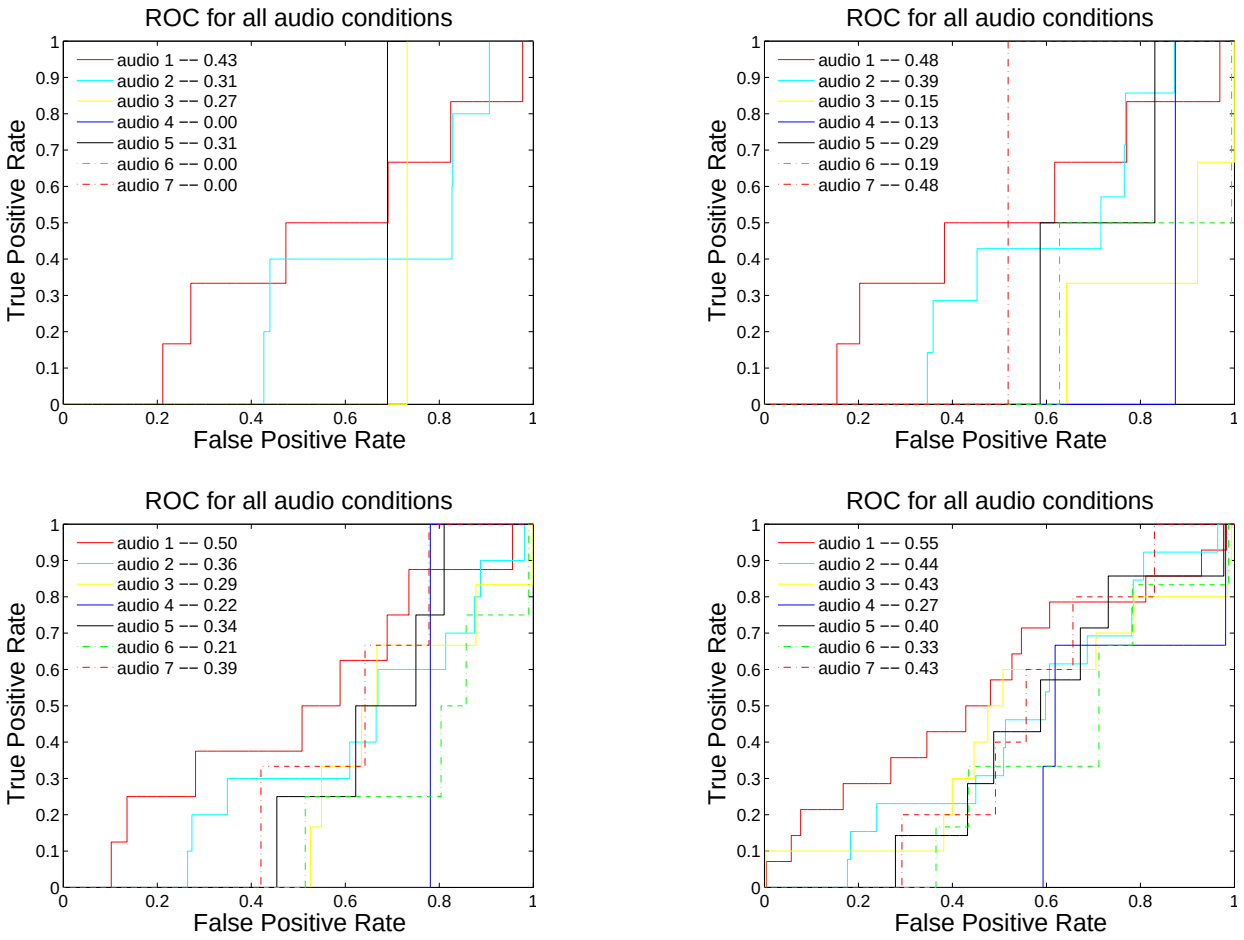

Figure 5.8: ROC plots for the TRECVID dataset for all audio transformations 1 – 7 when evaluating the top 40 (upper left), 60, (upper right), 100 (lower left), and 200 (lower right) reference recordings for the diarization based system.

much worse than the Telefonica system, which is near perfect. Though the results are poor in comparison, the diarization-based system is still able to perform better than random. Given the performance, perhaps a diarization-based system is not appropriate for the duplication detection task. Perhaps instead it would be better suited to a more general problem of finding acoustically similar data (as opposed to duplicate data).

Chapter 6

Conclusions and future directions

6.1 Conclusions

This thesis studies the state of speaker diarization. We analyze six state-of-the-art speaker diarization systems for two meeting audio conditions: multiple distant microphone (MDM) and single distant microphone (SDM). We find that for both the MDM and SDM conditions, all six systems perform poorly during short segments and near speaker change points. In fact, for the MDM condition over 40% of the Diarization Error Rate (DER) occurs within 0.5 seconds of a speaker change point for all of the systems. A large amount of the error during short segments and near speaker change points is due to overlapped speech, or multiple people speaking simultaneously. After removing the errors which occur during overlapped speech, over 22% of the MDM DER occurs within 0.5 seconds of the change point for all systems. This is still significant, especially considering only 12% of scored speech time occurs within 0.5 seconds of a speaker change point.

Although the DER decreases as the segment duration increases, this trend did not have as big of an impact on the total DER. Over 70% of the scored speech time occurs in segments greater than 2.5 seconds long. Therefore, despite the poor performance in DER for short segments, they contribute little to the overall DER.

After observing the speaker diarization error rate trends across multiple systems, we conduct a more detailed analysis for the ICSI speaker diarization system. The multiple system analysis is performed on the speaker diarization outputs for a given setup. However, in the ICSI speaker diarization system analysis, we are able to gain more insight into the behavior of the system by modifying various aspects of the algorithm, including changing the input feature vectors and modifying the minimum duration constraint, and then observing how the diarization performance changes. The focus of our investigation is the speaker error rate. Since identifying the appropriate speaker is one of the main tasks in speaker diarization, it is important to maintain speaker error rate performance.

We first explore the effects of changing the input feature vectors. The purpose of this

study is to determine if speakers change their speaking patterns near change points, thereby causing the speaker diarization system results to suffer at those times. We study three methods of modifying the feature vectors. The first method, referred to as resampled, replaces the feature vectors during hypothesized speech time with new features resampled from the GMM trained on that speaker. The second and third methods, referred to as flipQ and flip050, flip speaker homogeneous speech partitions inside out. For flipQ, each partition is split into quarters and the first quarter and second quarter are swapped, as are the third and fourth quarters. For flip050, the first and last 0.5 seconds of each speaker-homogeneous speech partition are exchanged with the rest of the first half and second half of the partition, respectively. These methods all demonstrate the same trends of worse performance for shorter segments and speech near speaker change points. These results lead us to believe that speakers do not change their speech near speaker change points, and that poor performance near speaker change points is due to the speaker diarization algorithm.

We then observe the effect of replacing the minimum duration constraint in the last iteration of the ICSI speaker diarization algorithm with mean smoothing the log-likelihoods for each of the hypothesized speakers. After doing so, we obtain an 11.9% relative improvement in speaker error rate for the MDM condition and a 2.8% relative improvement for the SDM condition. After further analysis of the MDM results, we determine that the biggest improvements occur closest to speaker change points, as predicted.

Next, we examine a number of attributes based on the log-likelihood score for each hypothesized speaker cluster. We find that the difference between the largest and second largest log-likelihood score is better than the maximum log-likelihood score for determining when the ICSI system is likely correct (and incorrect). We then incorporate the log-likelihood difference attribute for cluster purification, where speaker models are trained only on the speech regions the system is most confident. Unfortunately, we find that cluster purification is not effective for the ICSI speaker diarization system. In general, the log-likelihood difference attribute performed better than the maximum log-likelihood, despite neither method consistently improving overall DER results.

Finally, we use a speaker diarization based system for the problem of audio duplication detection, which determines whether an audio query is a duplicate from a set of reference recordings. For a broadcast news dataset, the method works well for the unmodified and reverberation audio conditions. However, for resampled and bandpass filtered audio the results degrade, but are still better than random. We then test our system on the TRECVID dataset and compare our results to a state-of-the-art duplication detection system. On this dataset the state-of-the-art system is near perfect and outperforms the diarization-based system. Although the diarization based system performance is worse than the state-of-the-art the algorithm, it is still better than random and seems to be able to find similar audio clips. It is possible that diarization-based systems are more appropriate for grouping similar clips than determining whether a specific audio query is a duplicate.

6.2 Future directions

This thesis motivates a number of questions for further research. The future areas are outlined below.

With respect to the minimum duration constraint, it would be interesting to see what happens when mean log-likelihood smoothing is incorporated throughout the algorithm, instead of only at the final iteration. Our results show that mean-smoothing improves both the MDM and SDM conditions, and this suggests that smoothing at each iteration may improve performance.

It would also be useful to test additional rearrangements and other modifications of the input features to assess the impact of changes in speech patterns near speaker change points. For instance, what happens when pure speaker models are trained based on the reference transcription, and feature vectors are sampled from the true reference speaker models? Do a significant amount of errors still occur near the speaker change point? Are the new models (trained on the resampled features) similar to the pure models (used to generate the features)? And finally, if models are trained separately for the beginning/end of speaker segments and the middle of speaker segments for each speaker, how do the within speaker and between speaker distances compare?

With respect to purification, our results indicate the log-likelihood difference is a valuable criterion for selecting high-confidence segments (i.e. segments which are well modeled by the system). However, we were unable to fully explore the utility of this statistic due to the fact that purification (regardless of the selection criteria) does not improve performance with the ICSI system. Thus, it would be interesting to see if the log-likelihood difference works well for systems whose performance has been shown to improve significantly by re-training on a subset of the features. We also note that the clusters with the greatest log-likelihood difference might not be the best clusters for retraining; it may be more effective to consider another subset of the features, such as the second tier of likely correct frames.

Finally, while studying the difference between the largest and second largest log-likelihood values and maximum log-likelihood values, we observed that when considering all recording time – instead of only speech time – many of the largest maximum values correspond to nonspeech time while very few of the largest difference values correspond to nonspeech time. Therefore, these difference attributes may also be useful for deciding which regions are speech and nonspeech.

Bibliography

[1] Hidden markov model toolkit (HTK).

[2] SHoUT toolkit web page.

[3] Sound eXchange.

[4] J. Ajmera, H. Bourlard, I. Lapidot, and I. McCowan. Unknown-multiple speaker clustering using HMM. In *ICSLP*, Denver, Colorado, 2002.

[5] J. Ajmera and C. Wooters. A robust speaker clustering algorithm. In *Proceedings of IEEE Speech Recognition and Understanding Workshop*, St. Thomas, US Virgin Islands, 2003.

[6] J. Ajot and J. Fiscus. RT-09 speaker diarization results, May 2009.

[7] X. Anguera, T. Adamek, D. Xu, and J.M. Barrios. Telefonica reseach at TRECVID 2011 content-based copy detection. In *NIST-TRECVID workshop*, 2011.

[8] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals. Speaker diarization : A review of recent research. *IEEE Transactions on Audio, Speech and Language Processing*, 20(2):356–370, February 2012.

[9] X. Anguera, A. Garzon, and T. Adamek. MASK: Robust local features for audio fingerprinting. In *ICME*, Melbourne, Australia, 2012.

[10] X. Anguera, C. Wooters, and J. Hernando. Robust speaker diarization for meetings: ICSI RT06s evaluation system. In *ICSLP*, Pittsburgh, Pennsylvania, September 2006.

[11] X. Anguera, C. Wooters, and J. Hernando. Acoustic beamforming for speaker diarization of meetings. *IEEE Transactions on Audio, Speech and Language Processing*, 15(7):2011–2023, September 2007.

[12] K. Boakye, O. Vinyals, and G. Friedland. Improved overlapped speech handling for speaker diarization. In *Interspeech*, Florence, Italy, 2011.

[13] P. Boersma and D. Weenink. Praat: doing phonetics by computer [computer program], 2008.

[14] S. Bozonnet, N. Evans, X. Anguera, O. Vinyals, G. Friedland, and C. Fredouille. System output combination for improved speaker diarization. In *Interspeech*, Makuhari, Japan, 2010.

[15] S. Bozonnet, N. Evans, and C. Fredouille. The LIA-Eurecom RT'09 speaker diarization system: Enhancements in speaker modelling and cluster purification. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, Texas, March 2010.

[16] C.J.C. Burges, D. Plastina, J.C. Platt, E. Renshaw, and H.S. Malvar. Using audio fingerprinting for duplicate detection and thumbnail generation. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 9–12, 2005.

[17] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of audio fingerprinting. In *Journal of VLSI Signal Processing*, volume 41, pages 271–284, 2005.

[18] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. The AMI meetings corpus. In *Proceedings of the Measuring Behavior 2005 Symposium on Annotating and Measuring Meeting Behavior*, 2005.

[19] M. Casey and M. Slaney. Fast recognition of remixed music audio. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Honolulu, Hawaii, 2007.

[20] S.S. Chen and P.S. Gopalakrishnan. Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*, pages 127–132, Lansdowne, Virginia, February 1998.

[21] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, 19(4):788–798, May 2011.

[22] M. Douze, H. Jegou, and C. Schmid. An image-based approach to video copy detection with spatio-temporal post-filtering. In *IEEE Transactions on Multimedia*, volume 12, 2010.

[23] E. Fox, E. Sudderth, M. Jordan, and A. Willsky. A sticky HDP-HMM with application to speaker diarization. *Anals of Applied Statistics*, June 2011.

[24] C. Fredouille, S. Bozonnet, and N. Evans. The LIA-EURECOM RT'09 speaker diarization system. In *RT'09, NIST Rich Transcription Workshop*, Melbourne, Florida, 2009.

[25] G. Friedland, L. Gottlieb, and A. Janin. Joke-o-mat: browsing sitcoms punchline by punchine. In *ACM Multimedia Conference*, pages 1115–1116, New York, New York, 2009.

[26] G. Friedland, A. Janin, D. Imseng, X. Anguera Miro, L. Gottlieb, M. Huijbregts, M. Knox, and O. Vinyals. The ICSI RT-09 speaker diarization system. *IEEE Transactions on Audio, Speech and Language Processing*, 20(2):371–381, February 2012.

[27] G. Friedland, C. Yeo, and H. Hung. Dialocalization: Acoustic speaker diarization and visual localization as joint optimization problem. *ACM Transactions on Multimedia Computing, Communications and Applications, Special Issue on Sensor Fusion*, 6(4), November 2010.

[28] J. Gauvain, L. Lamel, and G. Adda. Partitioning and transcription of broadcast news data. In *ICSLP*, Sydney, Australia, 1998.

[29] H. Gish, M.-H. Siu, and R. Rohlicek. Segregation of speakers for speech recognition and speaker identification. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 873–876, April 1991.

[30] J. Goldberger and H. Aronowitz. A distance measure between GMMs based on the unscented transform and its application to speaker recognition. In *Proceedings of Interspeech*, pages 1985–1988, 2005.

[31] V. Gupta, G. Boulianne, and P. Cardinal. Content-based audio copy detection using nearest-neighbor mapping. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 261–264, Dallas, Texas, 2010.

[32] J. Haitsma and T. Kalker. A highly robust audio fingerprinting system. In *ISMIR*, 2002.

[33] K. Han, S. Kim, and S. Narayanan. Strategies to improve the robustness of agglomerative hierarchical clustering under data source variation for speaker diarization. *IEEE Transactions on Audio, Speech and Language Processing*, 16(8):1590–1601, 2008.

[34] Y. Huang, O. Vinyals, G. Friedland, C. Muller, N. Mirghafori, and C. Wooters. A fast-match approach for robust, faster than real-time speaker diarization. In *IEEE ASRU*, pages 693–698, Kyoto, Japan, 2007.

[35] M. Huijbregts. *Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled*. PhD thesis, University of Twente, Enschede, Netherlands, 2008.

[36] M. Huijbregts, D. van Leeuwen, and F. deJong. Speech overlap detection in a two-pass speaker diarization system. In *Interspeech*, Brighton, United Kingdom, 2009.

[37] M. Huijbregts, D. van Leeuwen, and C. Wooters. Speaker diarization error analysis using oracle components. *IEEE Transactions on Audio, Speech and Language Processing*, 20(2):393–403, February 2012.

[38] M. Huijbregts and C. Wooters. The blame game: Performance analysis of speaker diarization system components. In *Interspeech*, Antwerp, Belgium, August 2007.

[39] H. Hung, Y. Huang, G. Friedland, and D. Gatica-Perez. Estimating dominance in multi-party meetings using speaker diarization. *IEEE Transactions on Audio, Speech and Language Processing*, 19(4):847–860, May 2011.

[40] D. Imseng and G. Friedland. Tuning-robust initialization methods for speaker diarization. *IEEE Transactions on Audio, Speech and Language Processing*, 18:2028–2037, 2010.

[41] Y. Jiao, B. Yang, M. Li, and X. Niu. MDCT-based perceptual hashing for compressed audio content identification. In *IEEE 9th Workshop on Multimedia Signal Processing*, pages 381–384, 2007.

[42] Q. Li, J. Wu, and X. He. Content-based audio retrieval using perceptual hash. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2008.

[43] R. Mertens, P-.S. Huang, L. Gottlieb, G. Friedland, and A. Divakaran. On the applicability of speaker diarization to audio concept detection for multimedia retrieval. In *IEEE International Symposium on Multimedia (ISM)*, pages 446–451, December 2011.

[44] N. Mirghafori and C. Wooters. Nuts and flakes: A study of data characteristics in speaker diarization. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1017–1020, Toulouse, France, 2006.

[45] N. Morgan, D. Baron, J. Edwards, D. Ellis, D. Gelbart, A. Janin, T. Pfau, E. Shriberg, and A. Stolcke. The meeting project at ICSI. In *Proceedings of the Human Language Technologies Conference*, San Diego, California, March 2001.

[46] T. Nguyen, E. Chng, and H. Li. T-test distance and clustering criterion for speaker diarization. In *Interspeech*, Brisbane, Australia, 2008.

[47] T. Nguyen, H. Sun, S. Zhao, S. Khine, H. Tran, T. Ma, B. Ma, E. Chng, and H. Li. The IIR-NTU speaker diarization systems for RT 2009. In *RT'09, NIST Rich Transcription Workshop*, Melbourne, Florida, 2009.

[48] NIST. The 2009 (RT-09) rich transcription meeting recognition evaluation plan, 2009.

[49] S. Shum, N. Dehak, E. Chuangsuwanich, D. Reynolds, and J. Glass. Exploiting intra-conversation variability for speaker diarization. In *Interspeech*, Florence, Italy, August 2011.

[50] S. Shum, N. Dehak, and J. Glass. On the use of spectral and iterative methods for speaker diarization. In *Interspeech*, Portland, Oregon, September 2012.

[51] M. Siegler, U. Jain, B. Raj, and R. Stern. Automatic segmentation, classification and clustering of broadcast news audio. In *Proceedings DARPA Speech Recognition Workshop*, pages 97–99, 1997.

[52] A.F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[53] A. Stolcke, G. Friedland, and D. Imseng. Leveraging speaker diarization for meeting recognition from distant microphones. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4390–4396, Dallas, Texas, March 2010.

[54] S. Sundaram and S. Narayanan. Audio retrieval by latent perceptual indexing. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 49–52, Las Vegas, Nevada, 2008.

[55] S. Tranter and D. Reynolds. An overview of automatic speaker diarization systems. *IEEE Transactions on Audio, Speech and Language Processing*, 14(5):1557–1565, September 2006.

[56] D. Vijayasenan, F. Valente, and H. Bourlard. An information theoretic approach to speaker diarization of meeting data. *IEEE Transactions on Audio, Speech and Language Processing*, 17:1382–1393, 2009.

[57] D. Vijayasenan, F. Valente, and H. Bourlard. Multistream speaker diarization beyond two acoustic feature streams. In *Proceedings of IEEE International Conference of Acoustics, Speech and Signal Processing (ICASSP)*, pages 4950–4953, Dallas, Texas, March 2010.

[58] A. Wang. An industrial strength audio search algorithm. In *Int. Conf. Music Info. Retrieval*, Baltimore, Maryland, 2003.

[59] C. Wooters and M. Huijbregts. The ICSI RT07s speaker diarization system. In *Proceedings of the RT07 Meeting Recognition Evaluation Workshop*, 2007.

[60] S.H. Yella and H. Bourlard. Improved overlap speech diarization of meeting recordings using long-term conversational features. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.