USC-SIPI Report #449

Multimodal and Self-guided Clustering Approaches Toward Context Aware Speaker Diarization

By
Tae Jin Park

May 2021

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
USC Viterbi School of Engineering
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Suite 400
Los Angeles, CA 90089-2564 U.S.A.

Multimodal and Self-guided Clustering Approaches Toward Context Aware Speaker Diarization

by

Tae Jin Park

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(Electrical Engineering)

May 2021

Table of Contents

List of Figures

Abstract

Speaker diarization has become an important field in recent years owing to the growing demand for conversational artificial intelligence and interactive entertainment systems. Within the process of a conversation analysis system, speaker labels should be predetermined before the natural language understanding units, allowing the system to achieve an accurate understanding regarding the content of the conversation. Thus, speaker diarization plays a crucial role in automatic discourse analysis systems as an essential preprocessing step.

In this dissertation, we propose techniques for multimodal speaker diarization approaches and self-guided clustering methods that can help in achieving the goal of a context aware speaker diarization system. Thus, our proposed speaker diarization system focuses on two aspects. First, we focus on a self-guided speaker diarization system that can determine the parameters on its own, based on the context of the input samples. This line of research includes a clustering phase and parameter tuning during speaker representation learning, and the determination of an adequate segment window length. We demonstrate that certain parameter tuning processes needed to perform a speaker diarization task can be automated. Second, we also investigate a method for incorporating other modalities, such as the lexical context, into a speaker diarization system. We show that, by incorporating the lexical context, the accuracy of the estimated speaker labels can be improved in the temporal domain. In doing so, we suggest a futuristic speaker diarization system that we will likely see in both industry and academia.

The overall objective of this dissertation is to propose novel techniques for improving the speaker diarization system using the aforementioned methods. In addition, we cover the machine

learning approaches behind the proposed techniques and how we can model the clustering and speaker recognition problems.

Chapter 1

Introduction

1.1 Introduction to Speaker Diarization

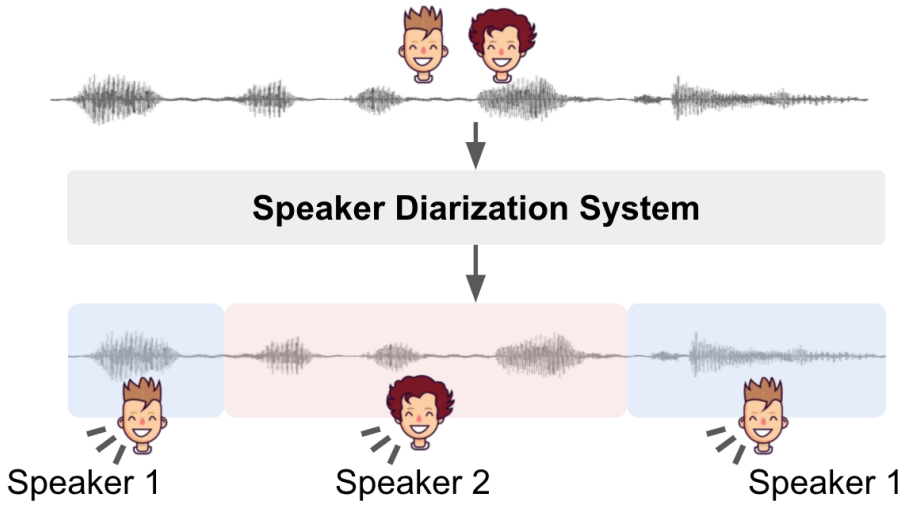


Figure 1.1: Input and output of speaker diarization system.

Speaker diarization is an essential component in speech applications under multi-speaker settings where spoken utterances need to be attributed to speaker-specific classes without prior knowledge of the speaker's identity. Speaker diarization is regarded as a task for determining "who spoke when" in a given audio recording. Fig. 1.1 shows the input and output of a speaker

diarization system. Speaker diarization technologies were initially developed as standalone approaches without much context of other components in a given speech application. Thus, speaker diarization has been considered an independent module that predicts speaker labels regardless of the output generated by the ASR module. However, as speech recognition technology becomes more accessible and advanced, there is an emerging trend of viewing speaker diarization as a crucial part of an overall speech recognition application, which can benefit from the speech recognition output to refine or improve the accuracy of the speaker diarization. Furthermore, the demand for speaker diarization technology has risen significantly as the demand increases for voice assistance in various industry applications such as autonomous vehicles, call centers, and smart homes.

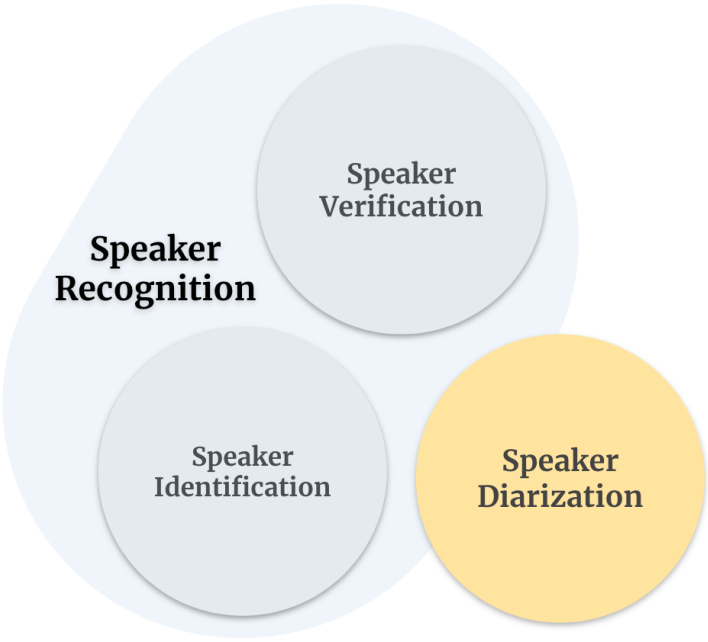


Figure 1.2: Categorization of speaker diarization field.

In terms of categorization, speaker diarization is occasionally categorized as a type of speaker recognition because the speaker diarization task itself is determining the identity of a short speech

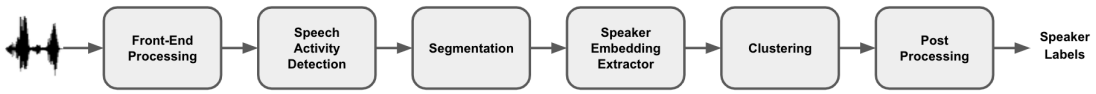


Figure 1.3: General speaker diarization process.

segment. However, speaker diarization should be clearly distinguished from the speaker recognition field, i.e., speaker verification and speaker identification. Speaker verification is concerned with accepting or rejecting the claim of a speaker's identity. For example, if a user claims the identity of person A, the speaker verification system can either reject or accept this hypothesis. Therefore, the output of the speaker verification system is true or false. By contrast, speaker identification is about determining the speaker. For example, if person A says something to a speaker identification system, the system estimates the identity of the individual and determines whether the speaker is person A, and not person B or person C. Thus, speaker verification and speaker identification apply the speaker identity recognition task at the utterance level, not at the segment level. Thus, speaker diarization is not included in the speaker recognition field in a strict sense. Fig. 1.2 depicts the categorization of the speaker diarization field in comparison to speaker recognition and speaker verification.

Traditional speaker diarization system consists of a few modules that can be optimized separately. Fig. 1.3 shows a conventional speaker diarization pipeline. The raw audio input is processed using the front-end processing system. In general, front-end processing systems for speaker diarization tasks include speech enhancement, speech reverberation, and speech separation. The output of the front-end processing system affects not only the following modules but also the final outcome of the speaker diarization task. The refined signal from the front-end processing module goes into the speech activity detector (SAD) module, which returns the time stamps of the regions where the speech activities are sensed using a trained machine learning model. Thus, after applying the SAD module, the speaker diarization system is expected to only deal with speech signals without silence or noise. Following the SAD module, the segmentation module

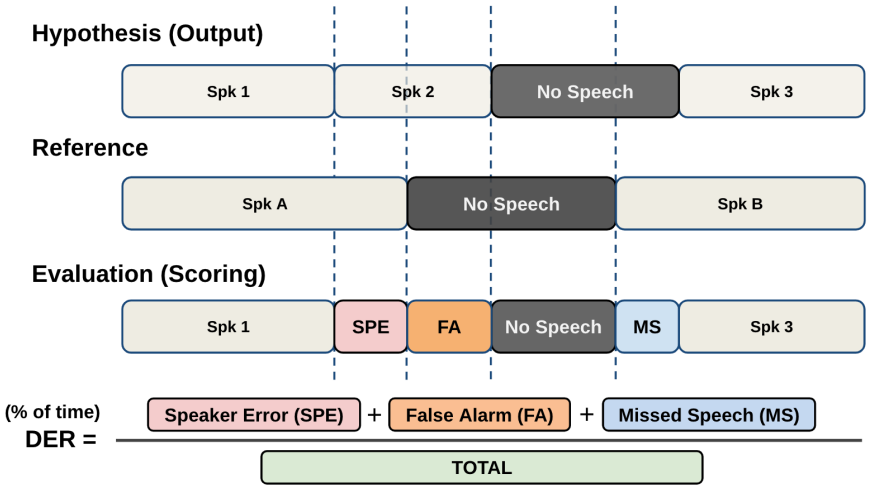


Figure 1.4: Example diagram of diarization error rate calculation.

returns approximately 0.5 to 2 s of short speech segments that will be used to determine the unit of the speaker label. The speech segments from the segmentation modules are fed into the speaker embedding extractor, and speaker representations are generated for each speech segment. These speaker representations are clustered using clustering algorithms to obtain separate groups that have a homogeneous speaker identity. Finally, the post-processing module refines the output from the clustering algorithm to reduce any errors that may occur. The error in a speaker diarization system output is mostly evaluated by calculating the diarization error rate (DER) [26] with the given reference transcript (ground truth). The DER is the sum of three different errors: a false alarm (FA) in the speech, a missed speech detection, and confusion between speaker labels.

$$\textbf{DER} = \frac{\text{FA} + \text{Missed} + \text{Speaker-Confusion}}{\textit{Total Duration of Time}} \tag{1.1}$$

Because speakers are not identified through speaker diarization, the hypothesis can include numerous combinations for matching the estimated speakers with the ground truth labels. To establish a one-to-one mapping between the hypothesis outputs and the reference transcript, the Hungarian algorithm [12] is employed. The most widely used DER calculation scheme is the method from

the Rich Transcription (RT) 2006 evaluation [26]. To mitigate the effects of an inconsistent annotation and human errors in the reference transcript, with the DER calculation scheme proposed in this document [26], a 0.25 s "no score" collar is set around every boundary of the reference segment. This "no score" collar mainly prevents the DER from depending too much on human errors made in the word boundaries in the reference transcripts.

1.2 Research Motivation

Despite the advancements in speaker diarization technology, a huge gap remains between the way humans process dialogue and the manner in which such dialogue is analyzed by a speaker diarization system. Human listeners do not need to train each functionality in a speaker diarization system, such as the speaker segmentation, speaker representation, or clustering. However, the prevailing speaker diarization systems usually require separate supervised tuning on clustering algorithms [72, 96, 126]. In addition, human listeners exploit a large amount of contextual information such as lexical context, speaker roles, and the information that can be captured from background noise. For example, human listeners can figure out that "I am doing great" will very likely be spoken from a speaker that is responding to a question such as "how are you doing?" spoken by another speaker. Human listeners also consciously or unconsciously presume the speaker roles even if such presumption is not completely accurate (e.g., a husband and a wife, or employees and a manager). Moreover, human listeners can obtain some clues about the location of the conversation. For example, typing sounds in the background or car noises from the street can help the human listeners have a better understanding of the surroundings. However, even a state-of-the-art speaker diarization system is incapable of incorporating such contextual information into a speaker diarization system. This leaves more room for improvement of the performance and robustness to the traditional speaker diarization systems. To deal with this gap

between the current speaker diarization paradigm and human listeners, we focus on two problems, i.e., multimodal approach and self-guided clustering method.

First, focusing on the limitation of the clustering algorithms based on supervised tuning, we cover the research topics related to self-tuning or autonomous clustering that reduce the burden of having a development set for clustering algorithms. We show that our proposed clustering approaches can achieve a competitive performance while not relying on a supervised tuning of the clustering algorithms.

Second, we focus on the fact that traditional speaker diarization systems do not employ contextual information such as the microphone layout or lexical content. We introduce studies regarding multimodal speaker diarization and show how different modalities can be integrated to enhance the performance of a speaker diarization system. Mostly, we focus on text and speech modalities and show how lexical information can contribute to the performance of the diarization.

Thus, our motivation is building a speaker diarization system that is less dependent on manual parameter tuning and closer to a context-aware speaker diarization system by employing lexical information. Thus, we aim to propose speaker diarization techniques that can contribute to building more advanced conversational AI systems that can perceive and recognize speakers similar to the way humans do.

1.3 Dissertation Outline

The remainder of this dissertation is structured as follows. In Chapter 2, we introduce previous studies related to our research, covering the early stage of speaker diarization as well as state-of-the-art diarization systems. In addition to previous studies, we briefly introduce the concepts of speaker diarization techniques and the evaluation datasets used in our research. Some passages in Chapter 2 have been quoted verbatim from the paper [78]. In Chapter 3, our proposed methods used to determine the parameters of the speaker diarization systems are introduced. In Chapter

4, we introduce our proposed methods of incorporating speaker turn probability into speaker segmentation and speaker diarization. In Chapter 5, we discuss the experiment setup used to evaluate our proposed speaker diarization framework along with the experimental results. In Chapter 6, we show the experimental results of the diarization system we propose in this dissertation. In Chapter 7, we discuss a few remaining issues that have yet to be resolved regarding the problem of speaker diarization and areas of future investigation. Finally, in Chapter 8, we discuss how our research is related to our initial aim and how it has directly or indirectly helped achieve our goal. Furthermore, we address the limitations of our proposed approaches and areas that can be improved in the future.

Chapter 2

Literature Survey and Background

2.1 Brief History of Speaker Diarization

Speaker diarization studies started being investigated and published in the 90's. Most of the speaker diarization studies in the 90's focused on unsupervised speech segmentation and clustering of audio segments that include speaker change or environmental or background changes [34, 90, 109, 102, 11]. During the early years of speaker diarization technology, Generalized Likelihood Ratio (GLR) and Bayesian Information Criterion (BIC) based methods were mainly employed as a way to model speaker characteristics for segmentation and clustering. Moreover, in this era, most of the speaker diarization works were expected to benefit Automatic Speech Recognition (ASR) on broadcast news recordings, by adaptively training acoustic models to speaker characteristics. [102, 31, 33, 32, 56]. These speaker diarization studies contributed to the activities across research groups around the world, leading to several research consortia and evaluations in the early 2000s such as organizations such as Augmented Multi-party Interaction (AMI) Consortium [3] supported by the European Commission and the Rich Transcription Evaluation [69] hosted by the National Institute of Standards and Technology (NIST), which is still considered as a major contributor for speech recognition and speaker recognition community. These aforementioned organizations have made efforts to contribute on speaker diarization research across different domains from

Conversational Telephone Speech (CTS) [86, 91, 57, 116, 85, 46] and broadcast news [2, 115, 128, 87, 10, 64, 63] to meeting conversations [4, 82, 1, 43, 44, 6, 53, 5]. The technologies and new approaches were not limited to modeling of speaker characteristics or speaker clustering but also include, but not limited to, beamforming [5], Information Bottleneck Clustering (IBC) [123], Variational Bayesian (VB) approaches [85, 119], Joint Factor Analysis (JFA) [48, 46]. Specifically, i-vector [21] approach was the most popular method that was applied for speaker verification and speaker diarization.

After the advent of deep learning in the early 2010s, the speaker diarization field also employed numerous advanced modeling capabilities of the neural networks trained on large-scale data. Owing to the superior feature extraction capabilities of deep neural networks (DNNs), many of DNN based speaker embedding extractor models such as d-vectors [120, 37, 127] or the x-vectors [106] gained popularity. Compared to i-vector[21, 100, 24, 99] which is based on JFA, these DNN based embeddings improved the quality of speaker representations and also made the training process significantly easier with more data [133], and robustness against speaker variability and challenging acoustic conditions. More recently, End-to-End Neural Diarization (EEND) was proposed to replace individual sub-modules in the traditional speaker diarization pipeline with a single neural network model [28, 29]. Although the performance of EEND still lags behind the state-of-the-art modular speaker diarization systems, End-to-end structure is expected to open up unprecedented opportunities to address challenges such as joint optimization or multi-task training schemes.

2.2 Previous Studies on Speaker Modeling

2.2.1 GMM Based Speaker Modeling

The early days of speaker diarization systems were mostly based on relatively simple statistical models such as Gaussian Mixture Model (GMM) [88, 89] built on acoustic features such as the

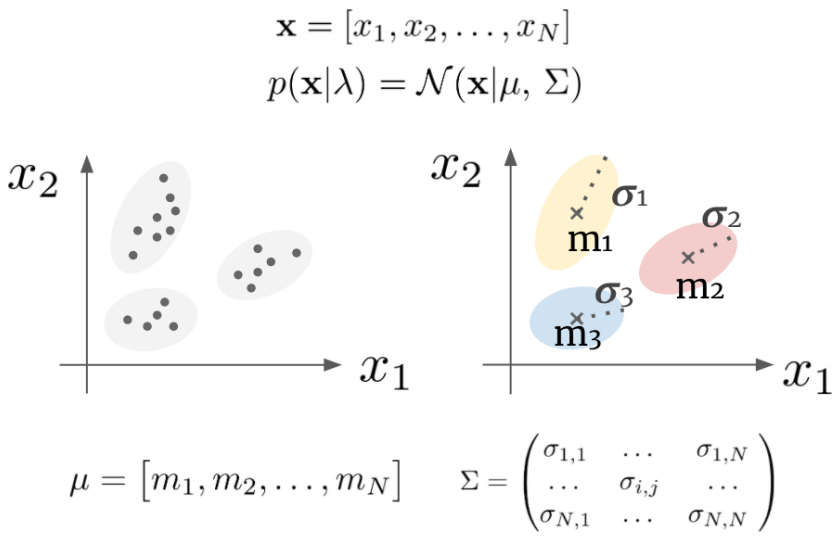


Figure 2.1: Gaussian mixture model on an acoustic feature space.

Mel-frequency cepstrum coefficients (MFCCs). Fig. 2.1 depicts how a GMM model can be built on a feature space with a set of mean and variance values. While there are many hypothesis testing methods for speech segment clustering processes such as greedy BIC [17], GLR [118] and KL [92] methods, greedy BIC method was the most popular approach. The BIC approach can numerically gauge how two Gaussian models are close to each other. In general, the BIC approach is applied to two separate speech segments as follows: Let $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ be the sequence of speech features sampled from the given audio recording and $\mathbf{x}$ is drawn from from an independent multivariate Gaussian process. Then each feature vector $\mathbf{x}_i$ can be written as follows:

$$\mathbf{x}_i \sim N\left(\mu_i, \Sigma_i\right), \tag{2.1}$$

where μ_i, Σ_i is the mean and covariance matrix of the i-th feature window. For the two segments of length N and M, two hypotheses H_0 and H_1 are established that can be denoted as follows:

$$H_0 : \mathbf{x}_1 \cdots \mathbf{x}_N, \mathbf{x}_{N+1} \cdots \mathbf{x}_{N+M} \sim N(\mu, \Sigma) \tag{2.2}$$

$$H_1 : \mathbf{x}_1 \cdots \mathbf{x}_N \sim N(\mu_1, \Sigma_1) \tag{2.3}$$

$$\mathbf{x}_1 \cdots \mathbf{x}_M \sim N(\mu_2, \Sigma_2) \tag{2.4}$$

Thus, hypothesis H_0 models two sample windows with one Gaussian while hypothesis H_1 models two sample windows with two Gaussians. Using the mean and variance values in Eq. (2.2) to (2.4), BIC equation can be written in the following way:

$$BIC = N \log |\Sigma| - N_1 \log |\Sigma_1| - N_2 \log |\Sigma_2| - \lambda P, \tag{2.5}$$

where P is the penalty term [17] defined as

$$P = \frac{1}{2}\left(d + \frac{1}{2}d(d+1)\right) \log N, \tag{2.6}$$

and d is the dimension of the feature. The penalty weight λ is generally set to $\lambda = 1$. Along with the BIC approach, agglomerative hierarchical clustering (AHC) was most popularly employed for clustering, resulting in the speaker homogeneous clusters. This method was refered to as greedy BIC [17] because the closest segment pairs are repeatedly merged to form speaker homogeneous clusters. GMM based hypothesis testing method with bottom-up AHC method was popularly used until i-vector and DNN-based speaker representations dominate the speaker diarization research scene.

2.2.2 Joint Factor Analysis and i-vector

Before the concept of speaker representations (*e.g.,* i-vector [21] or x-vector [106]) gained popularity, "Universal Background Model" (UBM) [89] framework was widely used for speaker recognition tasks by employing a few hundred of mixtures of Gaussians, which requires a large amount of speech data. The modeling of voice characteristics with GMM-UBM [89] is largely improved by the concept of JFA [49, 50]. GMM-UBM based hypothesis testing had a problem of Maximum a Posterior (MAP) adaptation that the modeling is not only affected by speaker-specific characteristics but also other unwanted factors such as channel and background noise. Therefore, the concept of supervector generated by GMM-UBM method left much room for improvement in terms of modeling capability. JFA targeted this problem by decomposing a supervector into speaker independent, speaker dependent, channel dependent and residual components. In Eq. (2.7), an ideal speaker supervector s is decomposed into multiple factors where $\mathbf{m}$ represents speaker independent component, $\mathbf{U}$ represents channel dependent component matrix, and $\mathbf{D}$ represents speaker-dependent residual component matrix. Along with these component matrices, vector $\mathbf{y}$ is for the speaker factors, vector $\mathbf{x}$ is for the channel factors and vector $\mathbf{z}$ is for the speaker-specific residual factors. All the vectors in the Eq. (2.7) have a prior distribution of unit Gaussian.

$$\mathbf{M(s)} = \mathbf{m} + \mathbf{Vy} + \mathbf{Ux} + \mathbf{Dz}. \tag{2.7}$$

The idea of JFA approach is later simplified by employing the concept of "Total Variability" matrix $\mathbf{T}$ which models both the channel and the speaker variability at the sametime. The vector $\mathbf{w}$ which is referred to as the "i-vector" [21]. Therefore, the Eq. (2.7) is simplified as follows:

$$\mathbf{M} = \mathbf{m} + \mathbf{Tw}. \tag{2.8}$$

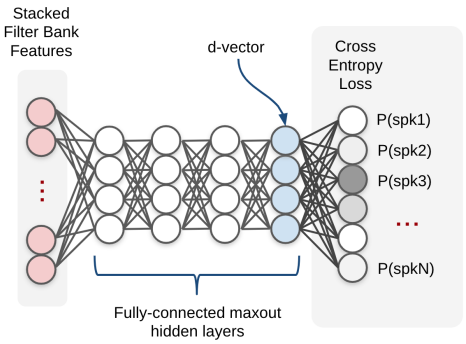


Figure 2.2: Diagram of d-vector model.

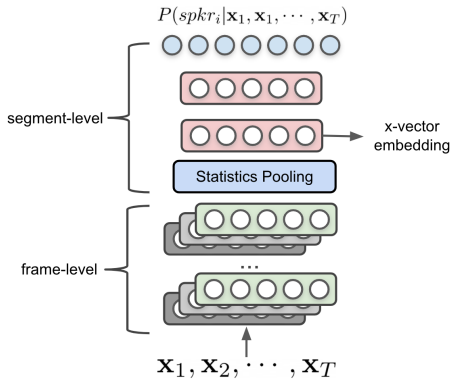


Figure 2.3: Diagram of x-vector embedding extractor.

In Eq. (2.8), $\mathbf{m}$ is the session and channel-independent component of the mean supervector. Similarly to JFA, $\mathbf{w}$ is assumed to follow unit Gaussian distribution and calculated by the MAP estimation process which is described in [47]. i-vectors had been considered to be the most effective speaker representations for speaker recognition and speaker diarization before the advent of DNN based speaker representations.

2.2.3 Neural Network Based Speaker Representations

The advent of neural networks and deep learning largely influenced the field of speaker representations learning. Thus, many of the recent speaker diarization systems employed DNN based speaker representations. The idea of representation learning or embedding learning appeared in face recognition field [111, 113]. The underlying idea of representation learning is that the neural network architecture can map the raw input signal source (*e.g.,* image or audio clip) to a dense floating point vector by feeding the input through multiple non-linear activation layers in a neural network model. Compared to the JFA approach, the DNN based representations do not require a hand-crafted design of the intrinsic factors such as factor analysis model. Moreover, DNN based models do not require an assumption of Gaussianity for the input feature data unlike GMM-UBM, JFA and i-vector models. Thus, thanks to the advance in artificial neural networks and

deep learning, the speaker representation learning process has also become more straight-forward and concise. On top of that, the inference speed is greatly improved compared to the i-vector approach since DNN based models do not involve a heavy computation such as matrix inversion in the inference process.

Among many DNN based speaker embedding extractors, d-vector [121] is one of the most well known early speaker embedding extractor models. d-vector employs stacked filter-bank features that are fed with context frames as an input feature. In terms of architecture, d-vector model is based on multiple fully connected layers and trained by cross entropy loss. In inference mode, the d-vector embedding vector is collected in the last fully connected layer as in Fig. 2.2. The d-vector model appeared in numerous speaker diarization studies, *e.g.,* , in [127, 133] showing superior performance over i-vector.

More recently, x-vector [105, 106] was proposed with more advanced architectures that are solely designed for time series signal. x-vector is based on time-delay neural network (TDNN) which models long term temporal dependencies better than Recurrent Neural Network (RNN) or multi-layer perceptron (MLP) based models with context windows. Moreover, x-vector employs a statistical pooling layer that mitigates the dependency on the utterance length. The statistical pooling approach is especially advantageous when it comes to speaker diarization since the speaker diarization systems are bound to process segments that are shorter than the regular window length. The structure of x-vector model is shown in Fig. 2.3. In terms of speaker representation quality, x-vector showed a superior performance by winning the NIST speaker recognition challenge [124] and the first DIHARD challenge [96].

2.2.4 Distance Measures for Speaker Embeddings

The most basic way to calculate the distance between two speaker embedding vectors is calculating cosine similarity between two vectors.

$$\cos(\theta) = \frac{\mathbf{x}_A \cdot \mathbf{x}_B}{\|\mathbf{x}_A\|\|\mathbf{x}_B\|} \tag{2.9}$$

Cosine similarity is still used in many of speaker diarization systems with spectral clustering or mean shift clustering algorithms. Cosine similarity does not require any training or parameter tuning but at the same time, it does not have any capability to project or weight the embedding vectors to enhance or refine the similarity measurement.

Another popular distance measurement technique for speaker embedding is Probabilistic Linear Discriminant Analysis (PLDA). PLDA has been widely used with x-vector or i-vector as a distance measurement method. PLDA models the given speaker representation ϕ_{ij} of the i-th speaker and j-th session into multiple factors as below:

$$\phi_{ij} = \mu + \mathbf{F}\mathbf{h}_i + \mathbf{G}\mathbf{w}_{ij} + \epsilon_{ij}. \tag{2.10}$$

Here, $\mathbf{m}$ represents mean vector, $\mathbf{F}$ represents speaker variability matrix, $\mathbf{G}$ represents channel variability matrix and ϵ represents residual component. $\mathbf{h}_i$ and $\mathbf{w}_{ij}$ are latent variables for $\mathbf{F}$ and $\mathbf{G}$, respectively. While training a PLDA model, $\mathbf{m}, \mathbf{\Sigma}, \mathbf{F}$ and $\mathbf{G}$ are estimated using expectation maximization (EM) algorithm where $\mathbf{\Sigma}$ is a covariance matrix. Based on the estimated variability matrices $\mathbf{F}$ and $\mathbf{G}$ and the latent variables $\mathbf{h}_i$ and $\mathbf{w}_{ij}$, the distance between two embedding vectors is calculated by hypothesis testing: hypothesis H_0 assumes that two samples are from the

same speaker and hypothesis H_1 assumes that the case that two samples are from two different speakers. The hypothesis H_0 can be written as follows:

$$\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} \mu \\ \mu \end{bmatrix} + \begin{bmatrix} \mathbf{F} & \mathbf{G} & 0 \\ \mathbf{F} & 0 & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{h}_{12} \\ \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \end{bmatrix}. \tag{2.11}$$

On the other hand, The hypothesis H_1 that assumes that the two speaker embedding vectors are from two different speakers is modeled as the following equation:

$$\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} \mu \\ \mu \end{bmatrix} + \begin{bmatrix} \mathbf{F} & \mathbf{G} & 0 & 0 \\ 0 & 0 & \mathbf{F} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{w}_1 \\ \mathbf{h}_2 \\ \mathbf{w}_2 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \end{bmatrix}. \tag{2.12}$$

In the above equation, the PLDA model projects the given speaker embedding vector ϕ_{ij} onto the subspace $\mathbf{F}$ to co-vary the most and de-emphasize the subspace $\mathbf{G}$ which pertains to channel variability. Using the above hypotheses, we can calculate a log likelihood ratio as the following equation.

$$s(\phi_1, \phi_2) = \log p(\phi_1, \phi_2 \mid H_0) - \log p(\phi_1, \phi_2 \mid H_1) \tag{2.13}$$

The likelihood ratio value $s(\phi_1, \phi_2)$ is usually employed with the AHC method by selecting the pairs with the biggest likelihood ratio value and merging the pair first. Ideally, stopping criterion should be $s(\phi_1, \phi_2)$=0, but in practice, the best performing stopping criterion varies over datasets and speaker embedding extractors. Therefore, the stopping criterion needs to be tuned on a development set to get the best performance. Setting the adequate stopping criterion is very crucial to the accuracy of the estimated number of speakers because the clustering process stops

when the distance between closest pairs reaches the threshold and the number of clusters is determined by the number of remaining clusters when the clustering process is stopped.

2.3 Previous Studies on Clustering Algorithms for Speaker Diarization

Clustering process is essential for the speaker diarization pipeline since the clustering algorithm finally groups the speech segments into the speaker homogeneous groups. Therefore, clustering algorithms largely determine the performance of a speaker diarization system. While numerous clustering algorithms have been proposed for speaker diarization, we review the most widely used clustering algorithms, AHC and spectral clustering.

2.3.1 Agglomerative Hierarchical Clustering

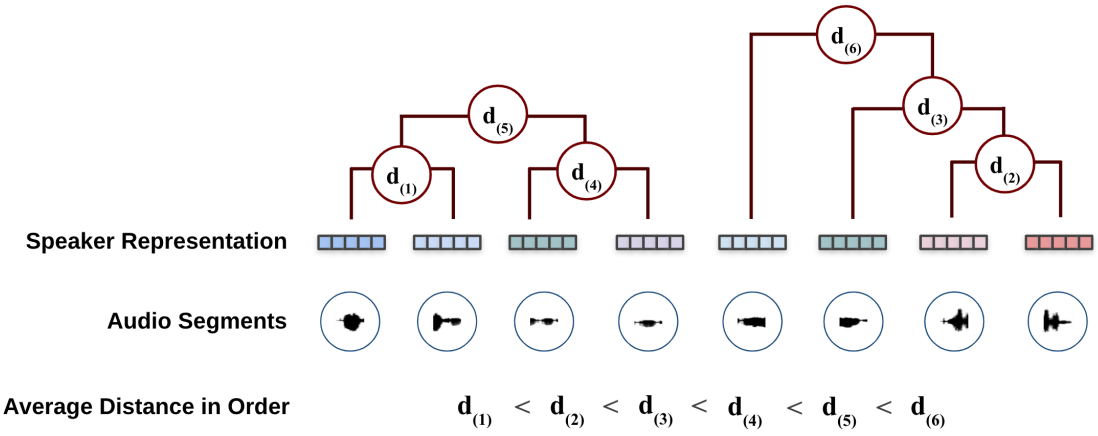


Figure 2.4: Agglomerative Hierarchical Clustering.

AHC is a clustering algorithm that has been widely used in many speaker diarization systems [7] with a number of different distance metric such as BIC [17, 36], KL [92] and PLDA [96, 8, 70]. AHC is based on an iterative process of merging the existing clusters of speech segments until the distance of the closest cluster pair meets a predetermined stopping criterion. The AHC process starts by calculating the similarity between N singleton clusters. At each step of the iteration, a pair of clusters that has the highest similarity is merged to form a new cluster. The iterative merging process of AHC can be described as a dendrogram in Fig. 2.4. One of the most crucial

aspects of AHC is the stopping criterion. Especially for speaker diarization tasks, the AHC process can be stopped using either a threshold for similarity or a target number of speakers. It is a common practice to adjust the stopping criterion to get an accurate number of clusters based on a development set. On the other hand, if the number of speakers is known or estimated in advance, the AHC process can be stopped when the number of merged clusters reaches the predetermined number of speakers.

2.3.2 Spectral Clustering

Spectral Clustering is another popular clustering approach for speaker diarization. Unlike AHC, spectral clustering is based on the similarity graph and the clustering is done by spectral embedding that is obtained by eigen decomposition of the affinity matrix. While there are many variations, spectral clustering involves the following steps.

1. **Affinity Matrix Creation**: In the context of speaker diarization, an affinity matrix contains each and every similarity between speech segment pairs in the given session. There are numerous ways of generating an affinity matrix $\mathbf{A}$ depending on the way the affinity value is generated and processed. For example, kernel methods were used in [68, 61, 100]. In these studies, the raw affinity value d is processed by a kernel such as $\exp\left(-d^2/\sigma^2\right)$ where σ is a scaling parameter. On the other hand, in the masking method [127], the raw affinity value d could also be masked by zeroing the affinity values below a threshold to only keep the prominent affinity values that actually affect the clustering process.

2. **Laplacian Matrix Calculation**: The graph Laplacian can be calculated in two different ways [125]: Normalized and unnormalized. The degree matrix $\mathbf{D}$ contains diagonal elements $d_i = \sum_{j=1}^{n} a_{ij}$ where a_{ij} is the element of the i-th row and j-th column in an affinity matrix $\mathbf{A}$.

(a) Normalized Graph Laplacian:

$$\mathbf{L} = \mathbf{D^{-1/2}AD^{-1/2}}. \tag{2.14}$$

(b) Unnormalized Graph Laplacian:

$$\mathbf{L} = \mathbf{D} - \mathbf{A}. \tag{2.15}$$

3. **Eigen Decomposition**: The graph Laplacian matrix $\mathbf{L}$ is decomposed into the eigenvector matrix $\mathbf{X}$ and the diagonal matrix that contains eigenvalues. Thus, $\mathbf{L} = \mathbf{X\Lambda X}^\top$.

4. **Re-normalization** (optional) : the rows of $\mathbf{X}$ is normalized so that $y_{ij} = x_{ij} / \left(\sum_j x_{ij}^2\right)^{1/2}$ where x_{ij} and y_{ij} are the elements of the i-th row and j-th column in matrix $\mathbf{X}$ and $\mathbf{Y}$, respectively.

5. **Speaker Counting**: Speaker number is estimated by finding the maximum eigengap in the eigengap vector $\mathbf{e}_{gap}$ in the following equation.

$$\mathbf{e}_{gap} = [\lambda_2 - \lambda_1, \lambda_3 - \lambda_2, \cdots, \lambda_N - \lambda_{N-1}] \tag{2.16}$$

   We refer to the estimated number of speakers as k in this article.

6. **k-means Clustering**: The spectral embedding is collected by taking the k-smallest eigenvalues λ_1, λ_2,..., λ_n and the corresponding eigenvectors $\mathbf{v}_1$, $\mathbf{v}_2$,..., $\mathbf{v}_k$ to make matrix $\boldsymbol{U} \in \mathbb{R}^{m \times n}$ where m is dimension of the row vectors in $\mathbf{U}$. Finally, the row vectors $\mathbf{u}_1$, $\mathbf{u}_2$,...,$\mathbf{u}_n$ are clustered by k-means algorithm.

There are many variations of spectral clustering methods that are used in the field of speaker diarization. Among these variations of spectral clustering algorithms, Ng-Jordan-Weiss (NJW)

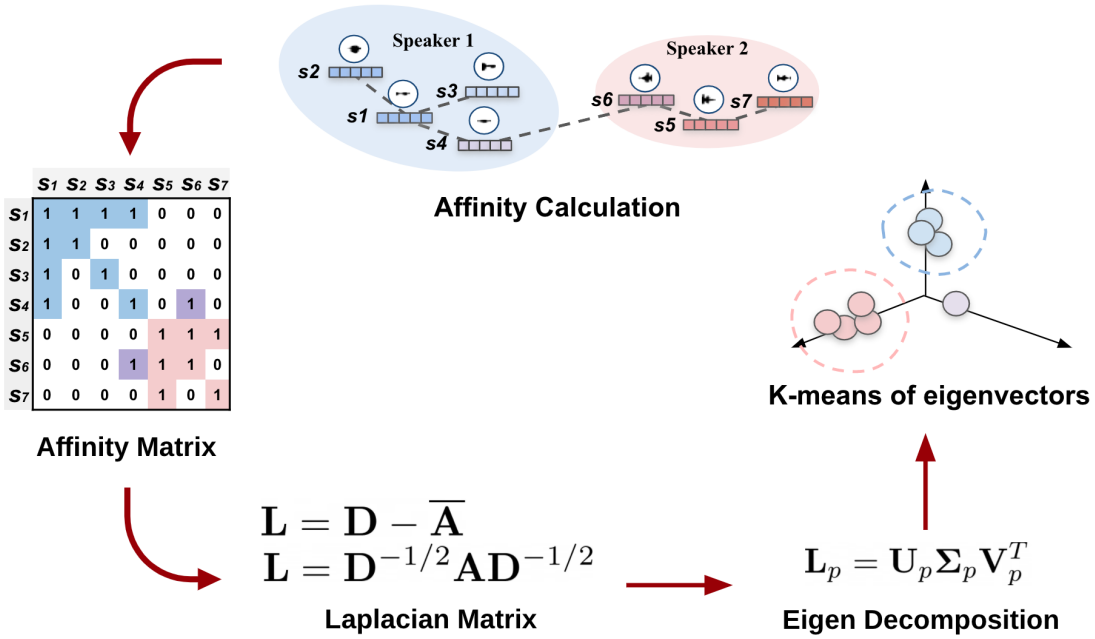


Figure 2.5: General steps of spectral clustering.

spectral clustering algorithm [66] is the most widely employed algorithm for speaker diarization. The NJW algorithm employs a kernel, $\exp\left(-d^2/\sigma^2\right)$, where d is a raw distance for calculating an affinity matrix. The affinity matrix, which can be regarded as a similarity graph, is used for calculating a normalized graph Laplacian. Moreover, NJW spectral clustering algorithm includes re-normalization before the k-means clustering process. The speaker diarization system in [68] used NJW algorithm while choosing σ by using predefined scalar value β and variance values from the data points. On the other hand, the speaker diarization system in [61] did not use β value for NJW algorithm. The speaker diarization system in [100], used fixed scalar value $\sigma^2 = 0.5$ for NJW algorithm.

Aside from the NJW algorithm, there are a few other types of spectral clustering methods that were successfully applied to speaker diarization tasks. The speaker diarization system in [127] used Gaussian blur for processing the affinity values by referring it to as diffusion process, $\mathbf{Y} = \mathbf{X}\mathbf{X}^{\mathbf{T}}$, and employed row-wise max normalization ($Y_{ij} = X_{ij} / \max_k X_{ik}$). In the spectral clustering approach appeared in [55], similarity values that are predicted from a DNN model were

used. We will discuss the importance of affinity value processing and parameter tuning in spectral clustering in Chapter 3.

2.4 Previous Study on Using Lexical Information for Speaker Diarization

Speaker diarization has long been considered a pre-processing procedure for ASR. In the traditional system structure for speaker diarization, speech inputs are processed sequentially without considering the ASR objective and thus separately trained and predicted. However, the traditional stand-alone type of speaker diarization has an issue that the tight boundaries of speech segments as the outcomes of speaker diarization are very likely to cause unexpected word truncation or deletion errors in ASR decoding. This is mainly because the unit of speaker diarization is based on the segments that are created from the speaker diarization module while word boundaries are generated from the decoder of ASR. In this section, we introduce the previous studies that employed speaker diarization systems in the context of ASR. These speaker diarization methods include the speaker diarization systems that are designed to refrain from truncating the word boundaries by taking the ASR output and process it with speaker diarization result. In addition, a few studies investigated on the joint modeling of speaker diarization and ASR to simultaneously. Thus, we will review the previous works that attempted the integration of ASR output or lexical information into speaker diarization systems.

2.4.1 Early Studies on Using Word Boundary Information

The lexical information from ASR output has been employed for speaker diarization system in a few different ways. First, the earliest approach was RT03 evaluation [117] which used word boundary information for segmentation purpose. In [117], a general ASR system for broadcast news data was built where the basic components are segmentation, speaker clustering, speaker

adaptation and system combination after ASR decoding from the two subsystems with the different adaptation methods. To understand the impact of the word boundary information, they used ASR outputs to replace the segmentation part and compared the diarization performance of the each system. In addition, ASR result was also used for refining SAD in IBM's submission [39] for RT07 evaluation. The system appeared in [39] incorporates word alignments from speaker independent ASR module and refines SAD result to reduce false alarms so that the speaker diarization system can have better clustering quality. The segmentation system in [103] also takes advantage of word alignments from ASR. The authors in [103] focused on the word-breakage problem where the words from ASR output are truncated by segmentation results since segmentation result and decoded word sequence are not aligned. Therefore, word-breakage (WB) ratio was proposed to measure the rate of change-points that are detected inside intervals corresponding to words. The DER and WB were reported together to measure the influence of word truncation problem. While the fore-mentioned early works of speaker diarization systems that are leveraging ASR output are focusing on the word alignment information to refine the SAD or segmentation result, the speaker diarization system in [15] created a dictionary for the phrases that commonly appear in broadcast news. The phrases in this dictionary provide the identity of who is speaking, who will speak and who spoke in the broadcast news scenario. For example, "This is [name]" indicates who was the speaker of the broadcast news section. Although the early speaker diarization studies did not fully leverage the lexical information to drastically improve DER, the idea of integrating the information from ASR output has been employed by many studies to refine or improve the speaker diarization output.

2.4.2 Recent Studies on Using Lexical Information for Diarization

More recently, a few speaker segmentation and diarization studies that take advantage of the lexical information from ASR by leveraging the turn probability of each word. The speaker change detection system proposed in [41] employed Long-Short Term Memory (LSTM) to estimate the

speaker change point by using i-vector [22] and word sequence together. The speaker change point detection in [41] should be categorized differently from the systems we introduced in Section 2.4.1 since the early studies only employed word boundaries or phrases that are commonly used while the method that use sequence of words to leverage the turn probability that can be inferred from long-term pattern of the multiple words.

The speaker diarization system proposed in [27] employed neural network models to utilize the linguistic patterns in the training data to improve the speaker diarization outcome. In addition, The proposed system in [27] targets speaker diarization tasks where participants in the given dialogue have distinct roles that are already known. To achieve this, a neural text-based speaker change point detector and a text-based role recognizer are employed to the speaker diarization system. Thus, the speaker diarization result was improved compared to the conventional acoustic-only speaker diarization system by employing both linguistic and acoustic information.

In addition, a joint training approach was recently applied to speaker diarization by simultaneously training an ASR model with speaker diarization system [98]. To train a joint ASR-SD model, a speaker tag in the transcription is fed to the end-to-end ASR models in the output of a recurrent neural network-transducer (RNN-T)-based ASR system. Although the benefit of jointly trained speaker diarization model over traditional speaker diarization model trained solely on speaker label, the proposed joint training method in [98] still has a potential to be an approach to leverage the lexical modality to create an additional benefit for the speaker diarization output.

2.5 Evaluation Datasets

The following datasets are used in the studies appearing in this dissertation.

1. **NIST SRE 2000 (LDC2001S97)** NIST SRE 2000 has been the most widely used dataset for speaker diarization in recent studies, and is referred to as CALLHOME. CALLHOME contains two to seven speakers for each utterance. For the CALLHOME dataset, a 2-fold

cross validation is conducted to match the test conditions with those in [106, 104] for all experiments.

2. **CALLHOME American English Speech (CHAES) (LDC97S42)** CHAES is a corpus that contains only English speech data and each conversation involves two to four speakers. CHAES is divided into training (train), development (dev), and evaluation (eval) sets, and the DER of the eval set is reported. Both the train and dev sets are used for parameter turning. A subset of CHAES, which contains only two speakers, is referred to as CH109 in the literature, and the CH109 dataset is tested by providing the number of speakers in advance to all tested systems. Since CH109 only contains two-speaker conversations, estimation of number of speakers is not performed for CH109 dataset. The remaining utterances in CHAES are used as a dev set for CH109.

3. **RT03 (LDC2007S10)** RT03 is an English telephonic speech dataset. The conversations in this dataset contain two to four speakers. The evaluations are based on the 14-vs-58 dev and eval split provided by the authors in [127].

4. **AMI meeting corpus** The AMI database consists of meeting recordings from multiple sites. We evaluate our proposed systems on the subset of the AMI corpus, which is a commonly used evaluation set that has appeared in many of the previous studies, and we use the splits (train, dev, and eval) appeared in these studies [71, 110, 129].

Chapter 3

Self-guided Clustering Approaches

3.1 Self-guided Approach for Multistream Diarization Task

In this section, we introduce our earliest clustering method work that targets a self-guided approach while having two streams of feature inputs: Acoustic feature (MFCCs) and amplitude (volume level). Specifically, the proposed method in this research [74] focuses on the specific task of speaker diarization from two information streams where two microphones are assigned to two speakers of interest. Some figures and formulas in [74] were used in this chapter and some passages in this chapter have been quoted verbatim from the paper [74].

In real-life scenarios, speakers are oftentimes co-located in the same room or close range in noisy environments with interfering background speakers. To tackle such speaker diarization tasks, the proposed multistream diarization system can exploit additional information. We propose

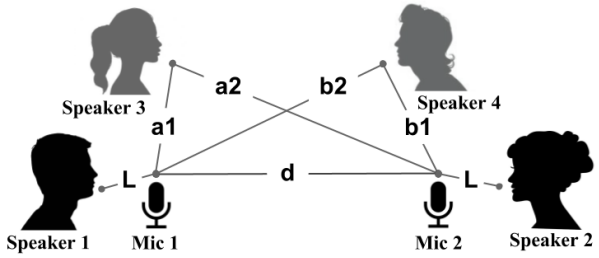


Figure 3.1: Illustration of main speakers co-located with the interfering speakers.

Minimum Variance of Bayesian Information Criterion (MVBIC) method to combine information from multiple diarization streams. To show the benefit of the multistream diarization system, we use a 2-microphoone setting and Root Mean Square Energy (RMSE) and MFCC features as our two diarization streams to evaluate the proposed self-tuning multistream diarization method.

3.1.1 Diarization Fusion: MVBIC

In this research, we investigate the MVBIC technique that efficiently weights BIC distances according to their reliability towards improved clustering accuracy. The concept of minimum variance optimization has also appeared in the studies from other fields, such as finance [94] or acoustics [60]. We assume that there is an underlying correct BIC stream that we are observing through a noisy channel. Thus, the hidden, correct BIC stream will be represented by b and its two observed, noisy versions by $\tilde{b}_i$, where in our case $i \in [1, M]$ and $M = 2$. Therefore:

$$\tilde{b}_i = b + n_i \tag{3.1}$$

where the above three are all random variables. With the above model (3.1), we want to obtain the optimal fusion weights that will lead to accurate estimation of the true b value:

$$\widehat{b} = \sum_{i=1}^{M} \omega_i \, b_i = \mathbf{w}^T \mathbf{b} \tag{3.2}$$

where i is index of vector representations and M is the number of feature vector representations. If we consider all N_s speech segments as given data points, we can calculate the sample variance of $\widehat{b}$ from given N_s segments as below:

$$Var\big[\widehat{b}\big] = \mathbf{w}^T \Sigma_b \mathbf{w} \tag{3.3}$$

Here, we make an assumption that the noise random variable n_i is mean zero and the two noise streams are uncorrelated. This assumption mostly holds if the features are exploiting diverse information as is in the case of MFCC and RMSE. In addition, we also assume that the random variable b, which is the hidden and correct BIC value, and noise random variable n_i are uncorrelated. Thus, the M by M covariance matrix Σ_b in equation (3.3) has elements described as:

$$\begin{aligned} \sigma_{b,i}^2 &= \sigma^2 + \sigma_{n,i}^2 \\ \sigma_{b,ij} &= \sigma_{b,ji} = \sigma^2 \\ &\text{where } i \neq j \text{ and } i, j \in [1, M] \end{aligned} \tag{3.4}$$

where $\sigma_{b,i}^2$, σ^2 and $\sigma_{n,i}^2$ are variances of b_i, b and n_i respectively. Using the above assumptions and constraining the sum of weights to 1, we can rewrite the variance of $\widehat{b}$ as follows:

$$Var[\widehat{b}] = \left(\sum_{i=1}^{M} \omega_i\right)^2 \sigma^2 + \sum_{i=1}^{M} \omega_i^2 \, \sigma_{n,i}^2 \tag{3.5}$$

$$= \sigma^2 + \sum_{i=1}^{M} \omega_i^2 \, \sigma_{n,i}^2. \tag{3.6}$$

Thus, minimizing variance of $\widehat{b}$, we can also minimize the variance of noise $\sigma_{n,i}^2$ on the assumption we make while keeping the σ^2 intact. Thus, we can set up a minimization problem as:

$$\begin{aligned} \textbf{Minimize:} \quad & Var\big[\widehat{b}\big] = \mathbf{w}^T \Sigma_b \mathbf{w} \\ \textbf{Subject to:} \quad & \mathbf{w}^T \mathbf{1} = 1. \end{aligned} \tag{3.7}$$

The solution to the equation (3.7) would be given as below:

$$\widehat{\mathbf{w}} = \frac{\Sigma_b^{-1} \mathbf{1}}{\mathbf{1}^T \Sigma_b^{-1} \mathbf{1}} \tag{3.8}$$

With the solution in equation (3.8), we estimate the weight in equation (3.2) to obtain $\widehat{b}$.

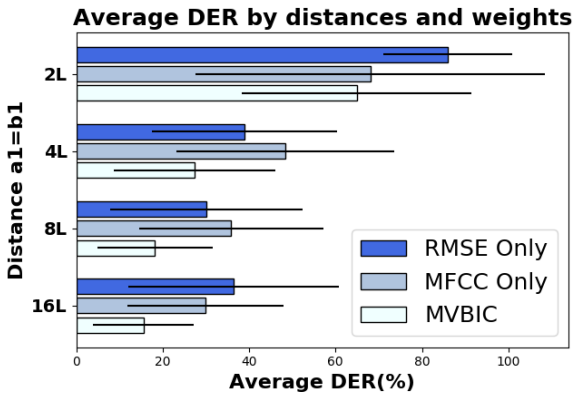


Figure 3.2: Average DER by distances of interfering speakers from primary speakers.

3.1.2 Performance of MVBIC Approach

We verify the performance of the MVBIC method on the simulated data, USCDiarLibri2,4 [74], and on the real-life recording, RT-06S data. For the individual diarization streams we perform BIC value based clustering down to 4 clusters. All the experimental results below are tested with md-eval software in RT06S dataset [26]. The following results are only evaluated for the primary speakers.

In this experiment described in the Fig.3.2, we evaluate the effect of the distance of the interfering speakers from the microphone locations. For this experiment, we use a rectangular arrangement for the 4 speakers and generated 20 sessions per distance. We keep the distance between the two primary speakers fixed (to 5L) and vary the distance a_1 and b_1, as in Fig. 3.1, keeping $a_1 = b_1$. As Fig.3.2 shows, MVBIC keeps the DER lower than the single feature diarization methods regardless of the location of the interfering speakers. Furthermore, this experiment indicates that both features perform worse when interfering speakers are near the primary speakers. Importantly we note that the distance of the interfering speaker greatly influences the relative accuracy of each diarization stream and hence the weight the stream should hold in case of fusion. This points further to the need for a dynamic fusion stream, such as MVBIC.

To verify the performance of the proposed MVBIC technique, we randomly assign the distance between all sources to be between 2 and 20 times L, as in Fig.3.1 and generate 50 sessions.

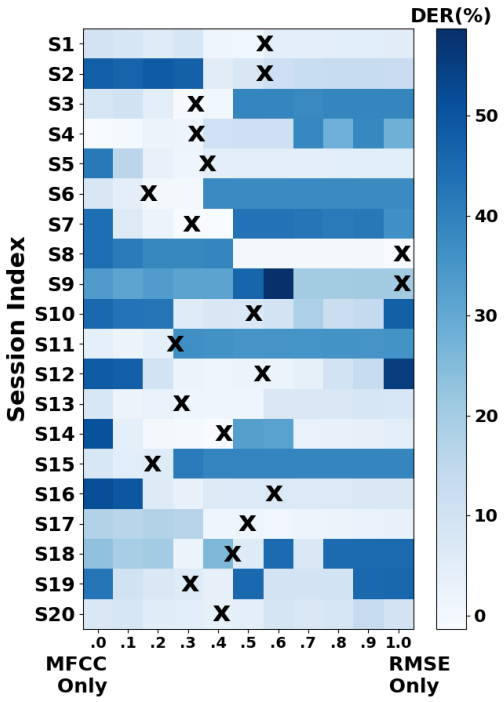


Figure 3.3: Plot of the estimated weights (×) layered on the results by each weight for the first 20 sessions.

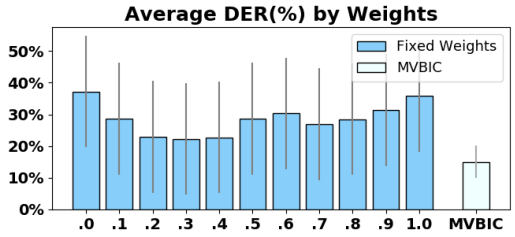


Figure 3.4: Average DER by fixed weights and estimated weights from MVBIC for generated dataset

Using this test dataset, the performance of the proposed MVBIC method is compared with fixed BIC weights. In Fig. 3.3 the x-axis represents w, the weight of the RMSE stream as follows: $\mathbf{w}^{\mathrm{T}} = [w_{\mathrm{RMSE}}, w_{\mathrm{MFCC}}] = [w, 1-w]$. We use $w = [0, 0.1, \ldots, 1.0]$. The DER results are visualized for each session and each weight in Fig. 3.3. Note that to keep the figures readable, only the first 20 sessions are depicted. The "×" marks in Fig. 3.3 describes BIC weights that MVBIC technique estimated for each session. We can see that the choice of $\mathbf{w}$ can play a significant role in the DER for each session. We also observe that the estimated weights by proposed MVBIC,

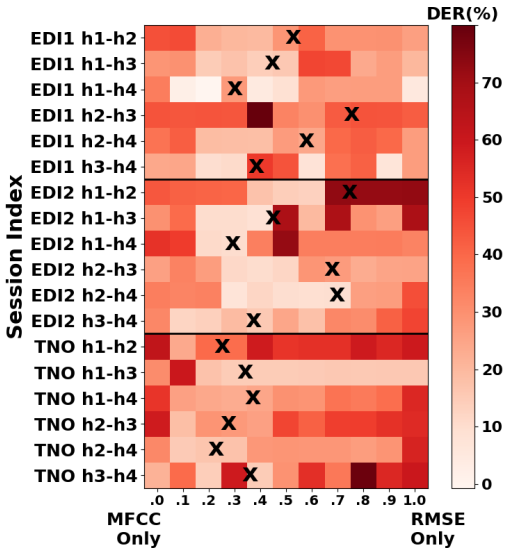


Figure 3.5: Plot of the estimated weights ($\times$) layered on the results by each weight for the subset of RT06S dataset. Indexes h1-h4 refer to index of microphones.

marked "$\times$", are mostly tracking the minima of DER (whitest regions of each row). This outcome indicates that MVBIC can estimate the values of the fusion vector $\mathbf{w}$ from given BIC streams that result in near optimum fusion DER. In Fig. 3.4, the DER results are shown for 50 sessions. The DER averages are plotted for the 50 for the different values of $\mathbf{w}$ as above. The last bar shows the result with the average DER based on the proposed MVBIC method.

By optimizing on the test set a fixed $\mathbf{w}$ we can see that we can obtain significant benefits over individual streams ($w = 0$ or $w = 1$) or equal weights ($w = 0.5$). The best performing value in this case would be $\mathbf{w}^{\mathrm{T}}$ =[0.3, 0.7]. However such optimization is not possible as the test data are not available at training time, but only serves as an upper bound for the static $\mathbf{w}$ fusion. The MVBIC method in contrast, even without optimization on the test data, can outperform any static fusion weight $\mathbf{w}$ as we can see from the last bar. This result points out that if the data is of high variability or mismatched to the training and development data, the proposed MVBIC can perform significantly better than a static, pre-tuned weight.

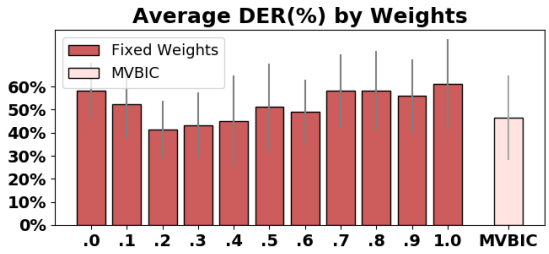


Figure 3.6: Average DER by grid-searched weights and esimated weights from MVBIC for subset of RT06S dataset

3.1.3 Evaluation on Real-life Data: RT06S

We test the performance of the proposed MVBIC system with individual head microphones for each session in RT06S dataset [26], which is real-life recording. We pick three meetings (EDI1: EDI_20050216-1051, EDI2: EDI_20050218-0900, TNO: TNO_20041103-1130) which have the same number of total speakers in USCDiarLibri2,4. Among the four speakers in each meeting, two speakers are regarded as primary speakers and the rest of two speakers are regarded as interfering speakers. Thus, total 6 ($4C2$) microphone combinations are tested for each of the three meetings.

Fig. 3.5 shows the same type of visualization as Fig. 3.3. We see that the MVBIC method does not pick as good candidates as we would expect. We also see that there seem to be multiple minima in the DER vs $\mathbf{w}$ space. This is likely due to the longer length of the sessions and the varying acoustic conditions. Since we only find one $\mathbf{w}$ using MVBIC per session, this is suboptimal. Fig. 3.6 shows the result for RT06S dataset in the same format as 3.4. The proposed method shows 46.5% of DER while the most accurate fixed weight result showed 41.5% of DER. Again we observe that the MVBIC method approaches the optimize-on-test-set performance of the static weight. Despite the highly mismatched conditions of this experiment, i.e. assuming stationary environment throughout the length of the session, which is false, and obtaining a single MVBIC weight w per session, and the higher-quality head-worn microphones, we still see significant benefits in using MVBIC.

In summary, we introduced a new simulated dataset USCDiarLibri for evaluating Diarization algorithms that enables tunable task difficulty and conditions. We described and employed a subset of the proposed dataset. More importantly, we introduced a MVBIC method to estimate the fusion weights among multiple diarization streams. The proposed technique does not require any tuning data to determine the weights while it closely estimates the ideal weights, optimally according to the minimum variance criterion. This has significant benefits in real-world environments where the recording conditions are highly variable and heterogeneous. Moreover, the proposed method allows to exploit any available diarization stream dynamically, *i.e.,* increasing the fusion information streams if appropriate.

3.2 Auto-tuning Clustering Method

Spectral clustering has been widely adopted in numerous speaker diarization studies [68, 61, 100, 101, 127, 55], and is a graph-based clustering technique that applies an affinity matrix, each element of which is the distance between a pair of speaker embeddings. Throughout the Laplacian matrix computations, the affinity matrix is converted into spectral embeddings, which are clustered using the k-means algorithm [58]. Despite its popularity, spectral clustering has a limitation in that its performance is sensitive to the quality of the affinity matrix. Owing to the noisy nature of the speaker embeddings and distance metrics, it is highly likely for some elements of the affinity matrix to possess noisy signals that can degenerate the clustering process. To tackle this issue, the spectral clustering algorithms applied in recent studies have employed either a scaling parameter [68, 100, 61] or a row-wise thresholding parameter [127] to place different weights across the elements in the affinity matrix. The downside of these approaches is that those parameters for either scaling or thresholding need to be optimized on a development set to obtain the maximize the performance. The burden of such hyper-parameter tuning during spectral clustering makes it more difficult to achieve a generalization of the clustering algorithm in an unseen testing environment.

In this section, we introduce a novel spectral clustering framework we proposed in [77]. Some figures, formulas and passages have been reused from [77]. The auto-tuning spectral clustering method is designed for self-tuning the clustering parameters that avoids the need for a hyper-parameter tuning when applying a development dataset. In this section, our proposed autotuing spectral clustering is compared with a well-known spectral clustering approach [67] described in a number of speaker diarization studies [68, 100, 61], and an AHC approach coupled with a PLDA [42, 84], which has also appeared in recent studies [30, 96, 104]. In addition, the performance of the development-set-optimized version of the proposed spectral clustering method was also tested to verify the benefit of our auto-tuning approach. The experimental results reveal that

the proposed auto-tuning approach achieves a comparable or even better performance than other widely used clustering algorithms with an optimized development set.

3.2.1 Traditional Spectral Clustering Algorithm

Spectral clustering is a graph-based clustering technique based on an affinity matrix and its eigenvalues. The affinity matrix is a similarity matrix for a given set of data points, where each element is determined based on the distance between a pair of data points in a given input. This algorithm has been widely used in a wide range of fields, such as image segmentation [132], multi-type relational data [59], and speaker diarization [68, 61, 100, 101, 127, 55], owing to its simple implementation and decent performance. Among the many variants of spectral clustering algorithms, the Ng-Jordan-Weiss (NJW) algorithm [67] has been the most widely used for speaker diarization tasks. The NJW algorithm consists of three steps, namely, the creation of an affinity matrix, Laplacian matrix computations, and k-means clustering [58]. The NJW algorithm employs a kernel method to form an affinity matrix. The similarity measure, which we refer to as $d(\mathbf{w}_i, \mathbf{w}_j)$, between two speaker embeddings from two speech segments is obtained through the following cosine similarity measure:

$$d(\mathbf{w}_i, \mathbf{w}_j) = \frac{\mathbf{w}_i \cdot \mathbf{w}_j}{\|\mathbf{w}_i\| \, \|\mathbf{w}_j\|}. \tag{3.9}$$

Each entry in affinity matrix $\mathbf{A}$ is defined as follows:

$$a_{ij} = \begin{cases} \exp\left(\frac{(-d(\mathbf{w}_i, \mathbf{w}_j)^2}{\sigma^2}\right) & \text{if } i \neq j \\ 0 & \text{if } i = j, \end{cases} \tag{3.10}$$

where σ is a scaling factor requiring tuning, and $\mathbf{A}$ can be considered an undirected graph G=(V, E). In this graph G, V represents the vertices (rows and columns in $\mathbf{A}$) and E represents

undirected edges (elements in $\mathbf{A}$). In the NJW algorithm, this affinity matrix $\mathbf{A}$ is normalized with the diagonal matrix $\mathbf{D}$ as follows:

$$\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}, \tag{3.11}$$

where $\mathbf{D} = \text{diag}\{d_1, d_2, ..., d_N\}$ and N indicates the number of dimensions of $\mathbf{A}$.

While the NJW algorithm has been widely employed for numerous applications, the NJW algorithm has inherent limitations in the context of speaker diarization. The similarity values we obtain from the distance measurements of a speaker's voice, for example, the cosine similarity described in (3.9), and used for an affinity matrix are merely estimated distant measurements and are dependent on how representative the speaker embeddings are in terms of speaker's particular voice characteristics. Thus, it is likely for some entries in the affinity matrix to have noisy signals that may degenerate the clustering process later during the process. Thus, without having a proper scheme to mitigate the effects of such inaccurate information from the affinity matrix, noisy similarity values could lead to a poor clustering result.

To deal with the above issue, there have been a number of schemes proposed in the literature that place different weights across the elements in the affinity matrix. In previous studies, only those entries in each row of the affinity matrix within the p-percentile are chosen [127], or scaling factors are used to control the weights of each element of the affinity matrix [68, 100]. The downside of such approaches is that the parameters for either thresholding or scaling need to be tuned using a development dataset, which can lead to a dependency of the clustering performance when selecting the development data. However. requiring such hyper-parameter tuning will become a burden in generalizing the clustering algorithm under unseen testing conditions. Thus, our proposed auto-tuning method addresses the problem of tuning the parameter without relying on additional development dataset. Our auto-tuning method is based on normalized Maximum Eigengap (NME) analysis and the following chapter describes the NME analysis in detail.

3.2.2 Normalized Maximum Eigengap Analysis

The following is an itemized description of the applied procedure:

1) Affinity Matrix: Unlike with the NJW algorithm, the affinity matrix $\mathbf{A}$ in the proposed framework is formed using raw cosine similarity values in (3.9) without applying a kernel or scaling parameter. From all N speech segments in the given input utterance, we obtain N^2 similarity values as described below:

$$a_{ij} = d(\mathbf{w}_i, \mathbf{w}_j), \tag{3.12}$$

where i and j are indexes of the speech segments.

2) p-Neighbor Binarization: The cosine similarity values in the affinity matrix A are binarized as either 0 or 1 to mitigate the effect of unreliable similarity values. This can be achieved by converting the p largest similarity values in each row into 1 while zeroing out the remaining values. In addition, p is an integer, and is determined based on the NME analysis described later.

$$\mathbf{A}_p = \mathit{binarize}(\mathbf{A}, p) \tag{3.13}$$

3) Symmetrization: To transform the affinity matrix $\mathbf{A}_p$ into an undirected adjacency matrix from a graph theory perspective, we apply symmetrization by taking the average of the original and transposed versions of $\mathbf{A}_p$ as follows:

$$\bar{\mathbf{A}}_p = \frac{1}{2}(\mathbf{A}_p + \mathbf{A}_p^T). \tag{3.14}$$

4) Laplacian: For a graph Laplacian, we use the unnormalized graph Laplacian matrix [125] as follows:

$$\begin{aligned} d_i &= \sum_{k=1}^{N} a_{ik} \\ \mathbf{D}_p &= \text{diag}\{d_1, d_2, ..., d_N\} \\ \mathbf{L}_p &= \mathbf{D}_p - \bar{\mathbf{A}}_p, \end{aligned} \tag{3.15}$$

where N is the size of the matrix $\bar{\mathbf{A}}_p \in \mathbb{R}^{N \times N}$ in (3.14).

5) Eigenvalue Decomposition (EVD): Apply an EVD to obtain eigenvalues for the Laplacian matrix $\mathbf{L}_p$:

$$\mathbf{L}_p = \mathbf{V}_p \mathbf{\Lambda}_p \mathbf{V}_p^{-1}. \tag{3.16}$$

6) Eigengap Vector: Create an eigengap vector $\mathbf{e}_p$ using the eigenvalues from $\mathbf{\Lambda}_p$ in (3.16) as follows:

$$\mathbf{e}_p = [\lambda_{p,2} - \lambda_{p,1}, \lambda_{p,3} - \lambda_{p,2}, \cdots, \lambda_{p,N} - \lambda_{p,N-1}], \tag{3.17}$$

where $\lambda_{p,i}$ is the i-th sorted eigenvalue in ascending order, given p for the binarization process in step *2)*.

7) Normalized Maximum Eigengap (NME): To compare the size of the maximum eigengap, $\max(\mathbf{e}_p)$, over different p values, we normalize $\max(\mathbf{e}_p)$ based on the maximum eigenvalue $\lambda_{p,N}$ to obtain the relative size of $\max(\mathbf{e}_p)$ compared to the scale of the eigenvalues. Thus, the NME value g_p for the given p is defined as follows:

$$g_p = \frac{\max(\mathbf{e}_p)}{\lambda_{p,N} + \epsilon}, \tag{3.18}$$

where $\lambda_{p,N} = \max(\mathbf{\Lambda}_p)$ and ϵ is an extremely small value (ϵ=1×10^{-10}). We obtain the ratio $r(p)$ between the pruning threshold p for the row-wise binarization and the NME value g_p as follows:

$$r(p) = \frac{p}{g_p}. \tag{3.19}$$

8) Estimation of p, $\hat{p}$: The value r_p is calculated throughout every $p \in \mathbb{N} \cap [1, P]$ and is stored in the list $\mathbf{r}$ as described below:

$$\mathbf{r} = [r(1), r(2), \cdots, r(P)]. \tag{3.20}$$

We find the index of $r(p)$ that is the minimum value in $\mathbf{r}$ and name the index as $\hat{p}$. Consequently, the parameter $\hat{p}$ attempts to minimize the DER:

$$\hat{p} = \text{argmin}(\mathbf{r}). \tag{3.21}$$

With this $\hat{p}$, we estimate the number of clusters k:

$$k = \text{argmax}(\mathbf{e}_{\hat{p}}). \tag{3.22}$$

9) Spectral Embedding: We take the smallest k eigenvalues and their corresponding eigenvectors to obtain the matrix of k-dimensional spectral embeddings $\mathbf{S} \in \mathbb{R}^{k \times N}$:

$$\mathbf{S} = \mathbf{V}_{\hat{p}}[1, N; 1, k]^T = [s_1, s_2, ..., s_N]. \tag{3.23}$$

10) k-means Clustering: We use the k-means clustering algorithm [58] to obtain k clusters from $\mathbf{S}$.

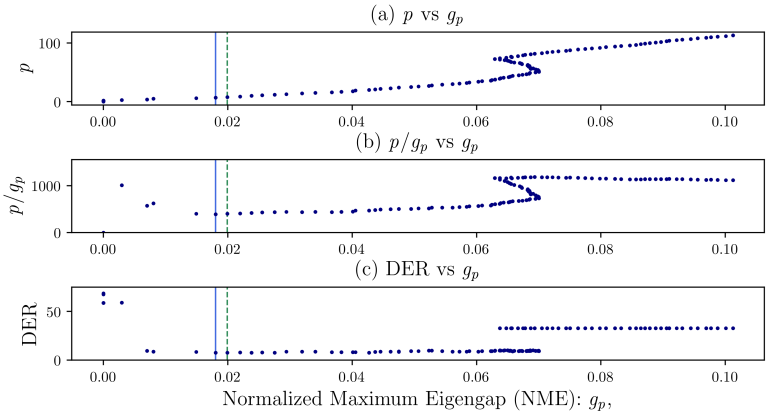


Figure 3.7: An example plot showing the relationship between the NME value of (a) g_p versus p, (b) p/g_p versus p, and (c) DER versus p. This example is from the utterance *iacg* in the CALLHOME dataset.

Since our approach for pruning the graph connections of the affinity matrix based on the p-neighbor binarization scheme is heavily dependent on the value of p, an in-depth analysis is required for the relationship between the NME value g_p and the pruning parameter p. In the previous study [125], it has been discovered that the size of the eigengap can be used as a quality criterion for a spectral clustering. More specifically, the relationship between the size of the eigengap and the purity of the clusters was investigated in [125, 107] using the perturbation theory and the Davis-Kahan theorem. In this context, We use the NME value g_p to gauge the purity of the clusters because such purity is directly linked to the speaker diarization performance. In so doing, we search for the most probable k and the most adequate p together using the eigenvalues. The most important part of the NME analysis is the relationship between p and g_p. Having a higher p value in an affinity matrix $\mathbf{A}$ generally leads to a larger g_p value with a higher purity measure of the clusters because the graph obtains more connections within each cluster. However, these binarized connections should be added with consideration on the accuracy of the estimation of the number of clusters because the binarization process makes all connections have an equal weight of 1, an excessive number of connections (i.e., a high p value) gives rise to a poor estimation of the number of clusters followed by a poor diarization result, although it gives a high g_p value. The worst case

can be easily understood by thinking of an affinity matrix whose elements are all equal to 1, which will always yield only a single cluster regardless of the actual number of clusters. As shown in Fig. 5.5=(a), we can see a gradual increase in g_p as p increases, whereas this tendency stops at approximately $p = 50$. As we increase p even more from $p = 50$, the estimated number of clusters drops and g_p increases again, meaning that we obtain a higher g_p value with a smaller estimated number of clusters.

To take advantage of the aforementioned trend, we focus on $r(p) = p/g_p$ value. The p value should be minimized to obtain an accurate number of clusters, whereas the g_p value should be maximized to obtain a higher purity of clusters. Thus, we calculate the ratio $r(p) = p/g_p$ to find the best p value by obtaining a p value in proportion to g_p. It is clearly shown in Fig. 5.5(b) and Fig. 5.5-(c) that the ratio of p to g_p, $r(p)$ follows the trend of DER. As described in (3.19), $r(p)$ indicates the slope in the p versus g_p plot. The lowest $r(p)$ value indicates that the resulting clusters have the highest purity measure g_p in proportion to p. In Fig. 5.5, the solid vertical lines indicate the estimated point of the lowest DER, whereas the dotted vertical lines show where the actual DER is the lowest value.

3.2.3 Experimental Results

To test the contributing performance of the clustering algorithms, we use the same speaker embedding extractor proposed in [106, 104] for all experiments described in this study. The evaluation method and metrics follow the approach described in [26]. The estimated number of speakers is limited to a maximum of eight speakers for all experiments. We test the following five different clustering algorithms:

1. COS+NJW-SC: This setup is the NJW algorithm from [67], which incorporates a cosine similarity measurement. The number of clusters is estimated using the method in [68].

Table 3.1: Experimental results using Oracle SAD

	COS+NJW-SC	COS+AHC	PLDA+AHC	COS+B-SC	COS+NME-SC
Oracle SAD	Spk. Err. (DER)	Spk. Err. (DER)	Spk. Err. (DER)	Spk. Err. (DER)	Spk. Err. (DER)
CALLHOME	24.05	21.13	8.39	8.78	**7.29**
CHAES-eval	30.31	31.99	24.27	4.4	**2.48**
CH109	13.06	29.8	9.72	**2.25**	2.63
RT03	6.56	5.66	1.73	**0.88**	2.21

Table 3.2: Experimental results with the system SAD

	COS+NJW-SC		COS+AHC		PLDA+AHC		COS+B-SC		COS+NME-SC	
System SAD	DER	Spk. Err.	DER	Spk. Err.	DER	Spk. Err.	DER	Spk. Err.	DER	Spk. Err.
CALLHOME	26.99	20.67	20.14	13.82	12.96	6.64	13.23	6.91	11.73	**5.41**
CHAES-eval	12.04	7.73	9.96	5.85	5.52	1.45	5.07	1.00	5.04	**0.97**
CH109	5.85	1.56	28.92	24.63	6.89	2.6	5.75	1.46	5.61	**1.32**
RT03	6.42	3.88	6.24	4.7	3.53	0.99	3.1	**0.56**	3.13	0.59

2. COS+AHC: This setup is identical to the setup in [106, 104], which uses the AHC algorithm; however, we use the cosine similarity instead of the PLDA for this setup.

3. PLDA+AHC: This setup, which is identical to [106, 104], is the AHC algorithm coupled with the PLDA. The stopping criterion of the AHC was grid-searched on each development set. We used the PLDA model from [104], and the mean vector and transform matrix for whitening were obtained from each development set.

4. COS+B-SC: This is the proposed spectral clustering framework using the p-neighbor binarization scheme, without the NME-based auto-tuning approach. i.e., p is optimized on each development set instead of applying $\hat{p}$ from (3.21).

5. COS+NME-SC: This is the proposed NME-based clustering algorithm, which includes the proposed auto-tuning approach. No hyper-parameter tuning or optimization is applied. For each utterance, $\lfloor \frac{N}{4} \rfloor$ of p values are searched by setting the P value in Algorithm 1 to $\lfloor \frac{N}{4} \rfloor$, where N is the number of total segments in a given input utterance. This search process requires P operations of the EVD, incurring a complexity of $O(PN^3)$.

Table 3.1 shows the experimental results when using Oracle SAD. Note that, except for the RT03 dataset, NME-SC shows a competitive performance with no parameter tuning at all. The DER of NME-SC is impressive, particularly for the CALLHOME dataset, where each utterance

has a varying number of speakers, and the proposed auto-tuning approach gains numerous advantages. Table 3.2 shows the experimental results for the system SAD. We used the ASpIRE SAD model [83], which is publicly available. With the system SAD setting, which is closer to scenarios found in a real-world environment, NME-SC outperforms all other methods except for RT03, where it shows an extremely close performance to the dev-set-optimized COS+B-SC method.

The performance gain from NJW-SC to B-SC indicates that the p-neighbor binarization scheme applying an unnormalized Laplacian approach can be effective because it demonstrates an extremely distinctive performance. More importantly, the performance gain from B-SC to NME-SC shows that the value of p can be effectively auto-tuned even without optimization on a development set. We also found a performance improvement of NME-SC over PLDA+AHC, indicating that the proposed clustering scheme can still obtain a competitive speaker diarization result without employing PLDA as a distance measure, all of which validate the effectiveness of the proposed auto-tuning spectral clustering framework using an NME analysis.

To summarize, a new framework for spectral clustering with auto-tuning was introduced in this research. The experimental results show that the proposed NME-based spectral clustering method is competitive in terms of performance, while not requiring any hyper-parameter tuning. Promisingly, the proposed method outperforms the widely used AHC method applying a PLDA. We will show how the auto-tuning approach can be exploited for multimodal speaker diarization in the later chapters.

3.3 Multi-scale Speaker Diarization

In general, the speaker diarization pipeline consists of speech activity detection (SAD), segmentation, speaker representation extraction, and clustering. The segmentation process largely determines the accuracy of the final speaker label because the segmentation determines the unit of diarization output that cannot be altered during the clustering process. In terms of segmentation, a speaker representation faces an inevitable trade-off between the temporal accuracy and speaker representation quality. It has been shown in many previous studies that the speaker representation accuracy improves as the segment length increases [105]. However, specifically in the context of speaker diarization, a longer segmentation means a lower resolution in the temporal domain because a segment is the unit of the process that determines the speaker identity.

In the early days of speaker diarization, the clustering process was BIC [17], which employs MFCCs as a form of representation for speaker traits. With BIC-based clustering and MFCCs, speech segmentation techniques [102] with a variable segmentation length have been employed because the benefit of having a proper segment length for input speech outweighs the performance degradation from variable segment lengths. This trend has changed with the increase in newer speaker representation techniques, such as i-vector [101, 95] and x-vector [106, 96], where fixing the length of the segments improves the speaker representation quality and reduces additional variability. For this reason, many previous studies have made a point of compromise at 1.0 [97] to 1.5 s [96, 52] depending on the domains they target. However, a fixed segment length has inherent limitations in terms of the temporal resolution because the clustering output can never be finer than the predetermined segment duration.

In this chapter, we introduce the method that we proposed in [80] which is our proposed multi-scale approach that addresses the problem arising from such a trade-off and applies a new segmentation approach. We notify that some passages, formulas and figures in this chapter have been reused from [80]. The proposed method employs a multi-scale diarization solution

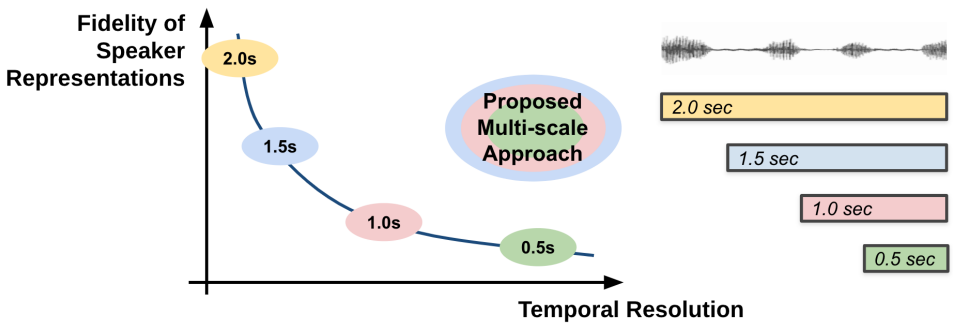


Figure 3.8: Trade-off curve between fidelity of speaker representations and temporal resolution.

where affinity scores from multiple scales of segmentation are fused using a neural score fusion system. The graph in Fig. 3.8 shows the trade-off between segment length and fidelity of speaker representations from two segments. Our goal is for our system to be located on the graph above the trade-off curve with a higher temporal resolution while at the same time achieving a superior accuracy of the affinity measure. We categorize the multi-scale approach as one of the self-guided clustering approaches since it instantaneously determines the weights between the given scales without manually setting up the proper segment length for each input. However, this does not indicate that the multi-scale speaker diarization approach we propose is an unsupervised method.

There have been few studies related to the problem discussed herein. In terms of speaker embedding extraction, few studies have employed a multi-scale concept for speaker embedding [45, 114] in the pursuit of processing short utterance lengths. These studies apply multi-scale aggregation [45] or multilevel pooling [114] in the feature level in the neural network models. Because the proposed neural network model does not generate speaker embeddings, feature-level multi-scale approaches are far from our focus.

By contrast, there are a few studies in which diarization systems aggregate the output of multiple modules. In [40], the authors employed a majority voting scheme on multiple segmentation streams. In [13], the authors introduced a cluster matching procedure that can integrate multiple diarization systems. In addition, in [108], a diarization output voting error reduction (DOVER) was presented for improving the diarization of a meeting speech. These previous studies deal with

either a feature-level multi-scale concept of a neural network [45, 114] or a diarization system integration [40, 13, 108], whereas the proposed method focuses on the score fusion of multi-scale speech segments.

Our proposed multi-scale approach has the following novelties. First, unlike conventional varying-length speech segmentation or single-scale segmentation modules, our system employs multiple discrete segment lengths and proposes a method to integrate the given scales. Second, the proposed method can attentively weigh the affinity from multiple scales depending on the domain and characteristics of the given speech signal. This distinguishes our work from approaches that require fusion parameters to be manually tuned on a development set. [72, 130]. In addition to these novelties, the proposed multi-scale approach outperforms a single-scale diarization system and achieves a state-of-the-art performance on the CALLHOME diarization dataset.

3.3.1 Motivation of the Multi-scale Speaker Diarization System

Fig. 3.9 shows a block diagram of the proposed method as opposed to the conventional speaker diarization pipeline. For the embedding extractor, we employ an x-vector in [106, 104]. We replace the segmentation process with a multi-scale segmentation process followed by a neural affinity score fusion (NASF) system, which will be described in the following sections. The NASF module outputs an affinity matrix similar to that in a conventional speaker diarization framework. In the proposed diarization, we employ the clustering method presented in [77].

Our proposed segmentation scheme for each scale is based on the segmentation scheme that appeared in a previous study [96, 104]. Fig. 3.10 shows how the proposed multi-scale segmentation scheme works. Although many different scale lengths and numbers of scales can be adopted, we employ three different segment lengths: 1.5, 1.0, and 0.5 s. The hop-length is half the segment length, which is 0.75, 0.5, and 0.25 s, respectively. In addition, the minimum segment length of each scale is set to 0.5, 0.25, and 0.17 s, respectively.

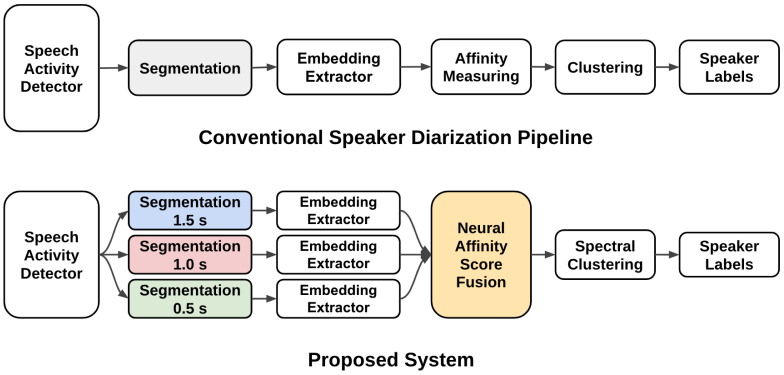


Figure 3.9: Comparison of multi-scale segmentation scheme to the traditional speaker diarization pipeline.

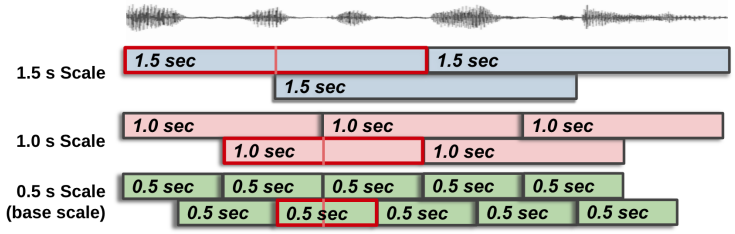


Figure 3.10: Example of multi-scale segmentation and mapping scheme.

We refer to the finest scale, 0.5 s, as the *base scale* because the unit of clustering and labeling is determined by base scale. For each base scale segment, we select and group the segments from the lower temporal resolution scales (1.0 s and 1.5 s) whose centers are the closest to the center of the base scale segment. This mapping is shown by the red bounding boxes in Fig. 3.10. By selecting the segments as in Fig. 3.10, the clustering results are generated based on the base scale segments, whereas measuring the affinity for the clustering process is achieved using the distance obtained from all three scales.

3.3.2 Neural Affinity Score Fusion Model

For the speaker diarization task, learning an affinity fusion model is not a straightforward downstream task unlike training speaker embedding from speaker labels because the diarization output is obtained through a clustering (unsupervised learning) approach. Thus, we derived an indirect

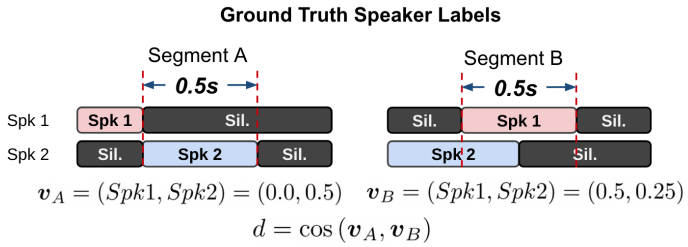


Figure 3.11: Example of training data label generation.

method that can learn a model for estimating the desirable weights for the affinity scores from multiple scales.

To represent the ground-truth composition of the speakers in the given segments, we employ a concept of a *speaker label vector* based on the duration of each speaker. The dimensions of the speaker label vector are determined based on the total number of speakers in a session. Fig. 3.11 shows an example of how we create labels of training data. Let segments A and B be a pair of segments for which we want to obtain an affinity score label. In Fig. 3.11, the speaker label vector v_A obtains values of $(0, 0.5)$ and $(0.5, 0.25)$ from the duration of the speaker labels from segments A and B, respectively. Since the speaker label vectors are always positive, the ground truth cosine distance value ranges from zero to one. To match the range, the cosine similarity value from the speaker embeddings are min-max normalized to the (0, 1) scale. In total, for L segments in the given session, we obtain ${}_LC_2$ ground truth affinity score labels, which were created for the base scale that has a segment length of 0.5 s.

To tackle the affinity weighting task, we employ a neural network model optimized using the Mean Square Error (MSE) between the ground truth cosine similarity d and weighted cosine similarity value y. We expect the estimated weight to minimize the gap between the ideal cosine similarity and the weighted sum of the given cosine similarity values (c_1, c_2, c_3). To achieve this, we employ an architecture similar to that of a Siamese network [51], which shares the weights of the networks to process the two different streams of information. Thus, we build a neural network

model that can capture the non-linear relationship between a set of affinity weights and a pair of speaker representations by setting up a pair of cloned neural networks.

Fig. 3.12 shows the architecture of the proposed affinity score fusion network. After the multi-scale segmentation process, speaker embeddings for each scale are extracted for the three segment scales. The set of embeddings (segment set A) are then processed using three parallel multi-layer perceptrons (MLPs) and the output of the MLPs is merged to form an embedding from all three scales. The forward propagation of the input layer to the merging layer is also applied to another set of segments (segment set B) to obtain a merged embedding for this set. After forward propagation of two streams of input, the difference between two merged embeddings are passed to the shared linear layer, which outputs the softmax values. We then take the mean of the softmax values from N input pairs.

$$\mathbf{w} = \left(\frac{1}{N} \sum_{n=1}^{N} w_{1,n}, \frac{1}{N} \sum_{n=1}^{N} w_{2,n}, \frac{1}{N} \sum_{n=1}^{N} w_{3,n} \right) \tag{3.24}$$

The set of averaged softmax values $\mathbf{w} = (\bar{w}_1, \bar{w}_2, \bar{w}_3)$ weights the cosine similarity values, $\mathbf{c} = (c_1, c_2, c_3)$, which are calculated using the speaker representations to obtain the weighted cosine similarity value as follows:

$$y_n = \sum_{i=1}^{3} \bar{\omega}_i c_{i,n} = \mathbf{w}^T \mathbf{c}_n, \tag{3.25}$$

where y_n is the output of the affinity weight network for the n-th pair out of N pairs. Finally, the MSE loss is calculated using the ground truth cosine similarity value d as follows:

$$\mathcal{L}(\mathbf{y}, \mathbf{d}) = \frac{1}{N} \sum_{n=1}^{N} (y_n - d_n)^2, \tag{3.26}$$

where d_n is the n-th ground-truth cosine score for the n-th pair of segments. In inference mode, we also take the mean of N sets of softmax values to obtain a weight vector $\mathbf{w}$.

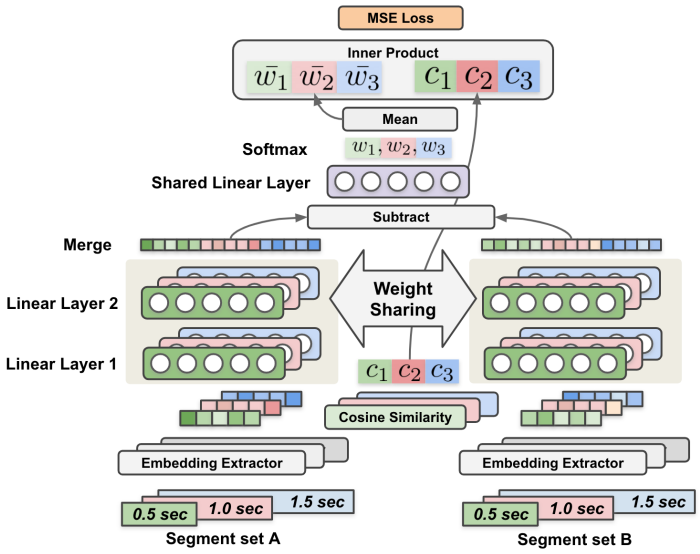


Figure 3.12: Neural multi-scale score fusion model

The NASF model estimates a weight vector $\mathbf{w}$ for each input session (an independent audio clip under a real-world scenario). For inference of the affinity weight, we randomly select $N{=}5 \cdot 10^5$ samples out of ${}_LC_2$ pairs per session, which has L base scale segments, and weigh the given affinity matrices as indicated in Fig. 3.13. The weighted affinity matrix is then passed to the clustering module. In our previous study about auto-tuning clustering in the previous section, we showed that the cosine similarity when applying the NME-SC method can outperform that of the prevailing clustering approaches, such as a PLDA coupled with AHC. Thus, we employ cosine similarity and NME-SC method to verify the efficacy of the proposed multi-scale affinity weight model by showing the additional improvement from the results in [77]. In addition, we compare the performance with systems based on single-scale segmentation methods.

3.3.3 Experimental Results

For the training of the proposed neural network model, we use the CHAES- and AMI-train splits. We also apply the CHAES-dev and AMI-dev sets to tune the hyper-parameters of the network. We use MLPs with 2 hidden layers and 128 nodes and apply the Adam optimizer with a learning

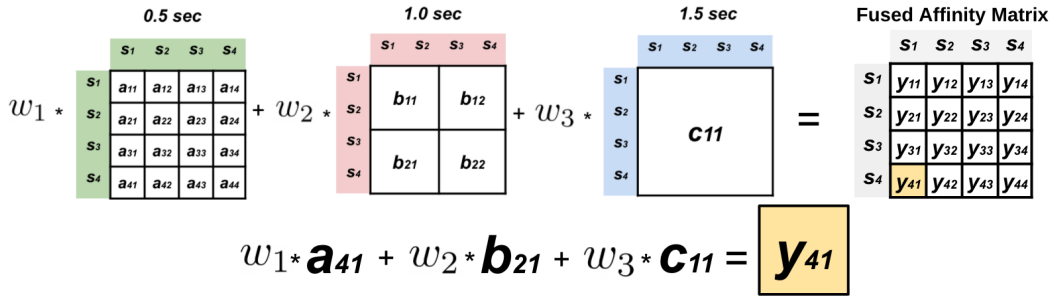


Figure 3.13: Example of weighted sum of affinity matrices

Table 3.3: Experimental results of baselines and the proposed methods

Dataset	# of Sessions	PLDA+AHC 0.5s	PLDA+AHC 1.0s	PLDA+AHC 1.5s	Previous Studies	COS+NME-SC 0.5s	COS+NME-SC 1.0s	COS+NME-SC 1.5s	Multi-scale COS+NME-SC EW [1]	Multi-scale COS+NME-SC NASF-D	Multi-scale COS+NME-SC NASF-S
AMI	12	38.42	20.07	10.55	8.92 [71]	26.96	9.82	3.37	6.51	3.89	**3.32**
CHAES-eval	20	4.58	3.15	3.28	2.48 [77]	8.71	3.35	2.48	2.52	2.47	**2.04**
CALLHOME	500	17.89	9.13	8.39	6.63 [54]	20.96	7.81	7.29	6.64	7.02	**6.46**

rate of 0.001. In this experiment, all systems employ a speaker embedding extractor (x-vector) that appeared in [106, 104]. The following baselines are for the distance measure and clustering method.

1. PLDA+AHC This approach is based on the AHC algorithm coupled with the PLDA as it appeared in [106, 104]. The stopping criterion of the AHC was selected based on a grid-search for each development set. We use the PLDA model provided in [104].

2. COS+NME-SC As stated in [77], NME-SC does not require a development set to tune the clustering algorithm. We use the same set of segments and speaker embeddings as PLDA+AHC but replace the distance measure with NASF from three different scales and replace the clustering algorithm with NME-SC. In this study, we do not evaluate combinations such as PLDA+NME-SC or COS+AHC because such combinations of algorithms have under-performed PLDA+AHC and COS+NME-SC in our previous experiments in [79].

- Equal Weight: This system is evaluated to show the efficacy of the NASF method over naive cosine similarity averaging. An equal weight system does not use any inference and applies equal affinity weights $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ for all sessions in all datasets.

- NASF-D: This system divides the input session into three equal-length sub-sessions and estimates six different affinity weight vectors ($\mathbf{w}$) for the six different affinity matrices ($3! = 6$), which are intra-sub-session (three sessions) and inter-sub-session (three sessions) affinity matrices. Finally, we calculate the weighted sum of these matrices and join the affinity matrices to cluster the integrated matrix as a single affinity matrix.

- NASF-S: This system estimates a set of affinity weights for an entire session. Thus, we have one affinity weight vector $\mathbf{w}$ for each session, and the entire affinity matrix is weighted using this single weight vector.

To gauge the performance of speaker diarization accuracy, we use oracle SAD output that excludes the effect of SAD module. For all evaluations and datasets, we estimate the number of speakers in the given session without additional information about speaker numbers. We employ an evaluation scheme and software that appeared in [26] to calculate Diarization Error Rate (DER).

We compare the DER obtained from the proposed method with the DER values obtained from each segment scale. Table 3.3 shows the DER from numerous settings and datasets. We show the DER values of the PLDA+AHC approach for three different segment scales (1.5, 1.0, and 0.5 s) and how the performance of the diarization changes with the distance measure and clustering method. We also list the lowest DER value that we could find that has appeared in a published paper on speaker diarization [71, 54], including the CHAES-eval results of our previous study [77].

Most importantly, we compare the COS+NME-SC methods with segment lengths of 0.5, 1.0, and 1.5 s with the proposed method. The best performing system, NASF-S, obtains relative improvements with error rates of 1.5%, 17.3%, and 11.4% for AMI, CHAES-eval, and CALLHOME,

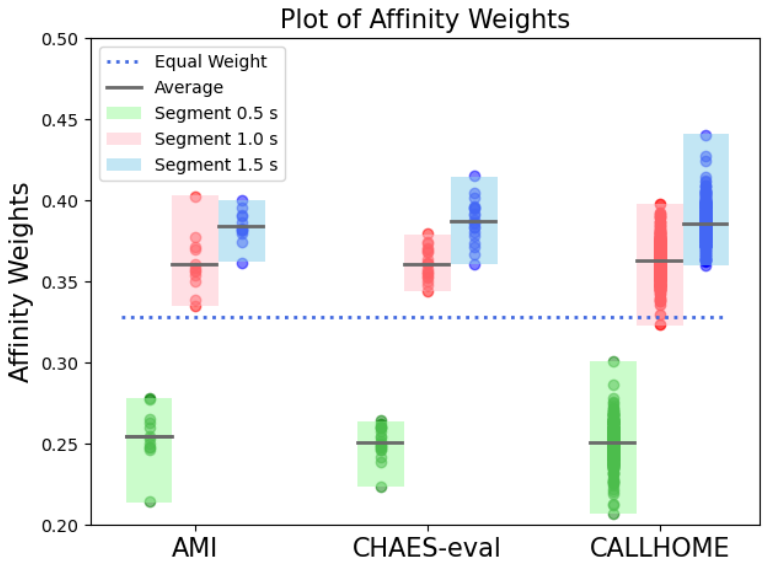


Figure 3.14: Plot of affinity weights by datasets.

respectively, over the 1.5-s COS+NME-SC baseline. For the AMI corpus, the improvement was minor whereas the CALLHOME and CHAES-eval sets showed a significant improvement given that the DER result from COS-NME-SC with 1.5-s segments is already competitive compared to the results appearing in previous published studies. In Fig. 3.14, we show the ranges and averages of the estimated weights over the sessions in each dataset. We can see that only the CALLHOME dataset shows a range that includes equal weight within the weight range for 1.0-s segments, whereas the weight ranges from AMI and CHAES-eval show no overlap with an equal weight. We conjecture that this is related to the result in which an equal weight shows an improvement for only CALLHOME.

From the experiment results, we can obtain a few valuable insights. The equal weight experiment gives conflicting results for AMI and CALLHOME. Nevertheless, from the equal weight experiment, we can verify that the desirable affinity weight cannot be simply found by averaging it and that the NASF approach can be a solution for estimating the desirable weights. The difference in performance gains between AMI and CALLHOME also shows the characteristics of a multi-scale approach. Because the longest segment we employ in our system is 1.5 s, we can argue

that the DER reduction comes from the higher resolution of the segments. This becomes clear if we compare the proposed method with the DER we obtain from 0.5-s segments. However, the gain from the proposed method was not that significant with the AMI corpus. We speculate that this is caused by the characteristics of the dataset because the average length of the continuous speaker homogeneous region in the AMI corpus is 2.56 s, whereas the lengths for CALLHOME and CHAES are 2.17 and 2.07 s, respectively. In this sense, we can argue that the CALLHOME and CHAES datasets are more likely to benefit from the proposed multi-scale diarization approach because a higher resolution can capture shorter speaker homogeneous regions. Another important finding obtained from this study is that varying the affinity weights in a session (i.e., a diarization session that is being clustered) does not lead to a good performance. Having a constant affinity weight in a single affinity matrix leads to a better performance, as we can see from the NASF-S outperforming NASF-D. In summary, the proposed neural network model, NASF module, estimates a set of weights that minimizes the gap between the weighted sum of the cosine affinity and the ground-truth affinity between a pair of segments. The proposed NASF system has a temporal resolution of 0.5 s and improves the diarization performance over conventional single-scale systems, achieving state-of-the-art performance on the CALLHOME dataset. In the later chapters, we will show how the multi-scale approach can benefit the multimodal speaker diarization system we propose in this dissertation.

Chapter 4

Lexical Modality for Speaker Diarization

4.1 Multimodal Speaker Segmentation and Diarization

While aforementioned methods show promising performance for speaker segmentation problems, very few studies have been published regarding the use of lexical information for the segmentation problem. This bias in speaker segmentation research is stemmed from the role that speaker diarization before ASR helps the performance of ASR performance. The motivation of the research we did in [73] started from changing the order of speech processing pipeline by putting speaker diarization after the ASR module. Thus, we were motivated to investigate a system that incorporates both lexical cues (language model) and acoustic cues. Moreover, incorporating both lexical cues and acoustic cues would be close to the way humans process speech signals. In this chapter, we introduce the system we proposed in [73] and its performance by comparing the proposed system to the traditional segmentation and diarization system. Some passages, formulas and figures in this chapter have been reused from [74].

4.1.1 Previous Studies

Most of the studies that involve lexical information or transcript is related with speaker identity detection [15, 25] by directly extracting the information from transcript to specify who the speaker is. The closest research so far is the approach with LSTM using character level Convolutional Neural Network (CNN) and i-vector training on transcript [41]. Unlike previous studies, this work targets the way to improve segmentation and the resulting diarization performance by integrating both language model and acoustic cues. To achieve this goal, we investigate the way to integrate both lexical cues and acoustic cues to perform speaker segmentation and speaker diarization using the output of the sequence-to-sequence model. Sequence-to-sequence models have been widely used for translation [112], end-to-end ASR systems [16] and text summarization[65]. The advantage of sequence-to-sequence model over Recurrent Neural Network (RNN) based models (LSTM [38], GRU [18]) is that it can summarize the whole sequence into an embedding and then pass it to the decoder. Moreover, sequence-to-sequence models can integrate information and process variable length sequences.In doing so, such a model can capture the information from both before and after the speaker change points. Moreover, attention structure helps the decoder create an accurate sequence by directly relating the speaker change points and the input sequence.

In addition, we also experimented with the effect of the performance degradation from the ASR output. In real life scenario, word by word transcription with word alignment information will never be given to the speaker diarization system. Therefore, we tested the performance of the proposed system with ASR output and compared the result with diarization performance based on transcription. Although ASR output hugely deteriorates the performance of diarization systems, we showed that sequence-to-sequence models with acoustic features can improve the diarization result over lexical information alone. The following sections explain the multimodal system we proposed in [73].

4.1.2 Proposed Sequence-to-sequence Model

Our proposed sequence-to-sequence model in [73] consists of an encoder, decoder and attention model that connects encoder and decoder. We use GRU with a 256 hidden layer and an attention model that has been applied to many state-of-the-art machine translation systems [9]. In this work, all the feature extraction was done along with word boundaries. In our proposed method the features are time-synchronous. All the features align with the word boundaries as follows:

WORD: The word sequences we use are obtained either from the reference transcripts or from an ASR output. We use a linear layer to convert one-hot word vector into word embedding as described in Fig. 4.1. The source sequence is 32 words in the reference transcript or ASR output. The target sequence for training is 32 words and added speaker turn tokens.

MFCC: We use 13-dimensional MFCCs extracted with a 25ms window and 10ms shift. Detailed specifications follow the feature extraction method proposed in [62]. We then average the MFCC features for the word-segment and thus derive a 13×1 vector for each word.

In the proposed system, the encoder is the part where all the features are integrated. Fig.4.1 shows how the proposed encoder is structured. Word embeddings and MFCC features are connected through linear layers. After the fully-connected layers, the embeddings are concatenated to be passed simultaneously. The concatenated vector is then fed to the GRU that is the encoder of the sequence-to-sequence system. We use 256 hidden unit size, word embedding size and output layer of linear layer for MFCC vector.

On the other hand, decoder only outputs the word sequence and the turn token and also trained with the word sequence and the speaker turn token. The speaker turn token is obtained from the transcription data. Note that we use "♯A" and "♯B". Fig. 4.2 describes the decoder side in our proposed system. Unlike word tokens, the loss of the speaker turn tokens are calculated in a different way that ignores the speaker IDs and only focuses on speaker groupings. For example, the speaker turn sequence of "♯A ♯B ♯A" is considered the same as "♯B ♯A ♯B". Between these two

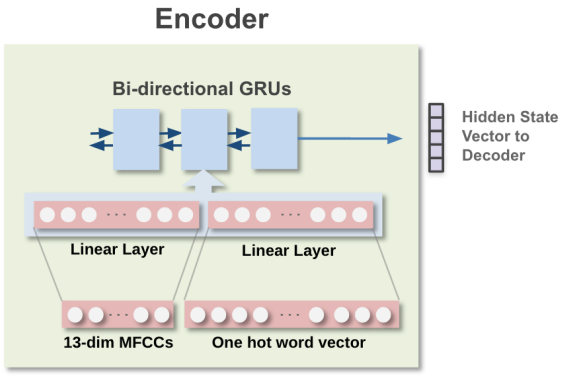


Figure 4.1: Encoder side of the proposed network.

versions of losses, our loss function selects the smaller loss. This loss function also avoids learning the probability between speaker turn tokens and words in the target sequences in the training set.

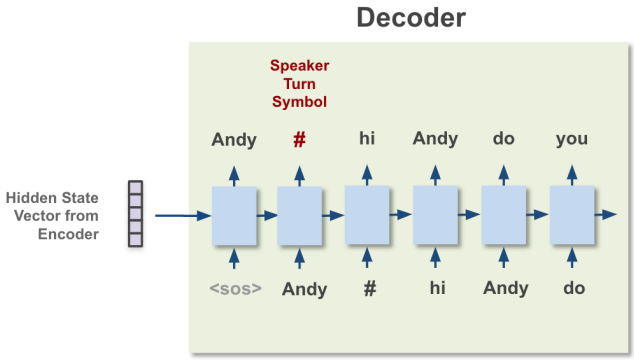


Figure 4.2: Decoder side of the proposed sequence-to-sequence model.

4.1.3 Speaker Turn Estimation

To maximize the accuracy of speaker turn detection, we employ shift and overlap schemes to predict the speaker turn. Fig. 4.3 explains how speaker turn prediction is done. A target window that has 32 word length sweeps the whole session from the beginning to the end. For each target window, we predict speaker turn tokens with our trained sequence-to-sequence model. At each prediction, we extract 32 words and 32 MFCC vectors from transcript and audio stream,

respectively. After we get the speaker turn vector, this vector is compared with the cumulative speaker turn sequence which is a matrix that stores all the speaker turn vectors obtained so far. The speaker turn vector is flipped if flipping the speaker vector gives less hamming distance. After we collect all the speaker vectors in the session, we take a majority vote for each word and finally make a decision for each speaker turn. A set of speaker turns for the given session is estimated through this process.

4.1.4 Clustering

We employ our BIC based agglomerative clustering algorithm based on [17] to perform the segment clustering. For the agglomerative clustering we employ the raw frame-level MFCC as features. We obtain the segmented MFCC streams using speaker turn information that is obtained from the speaker turn estimation process described in the previous section.

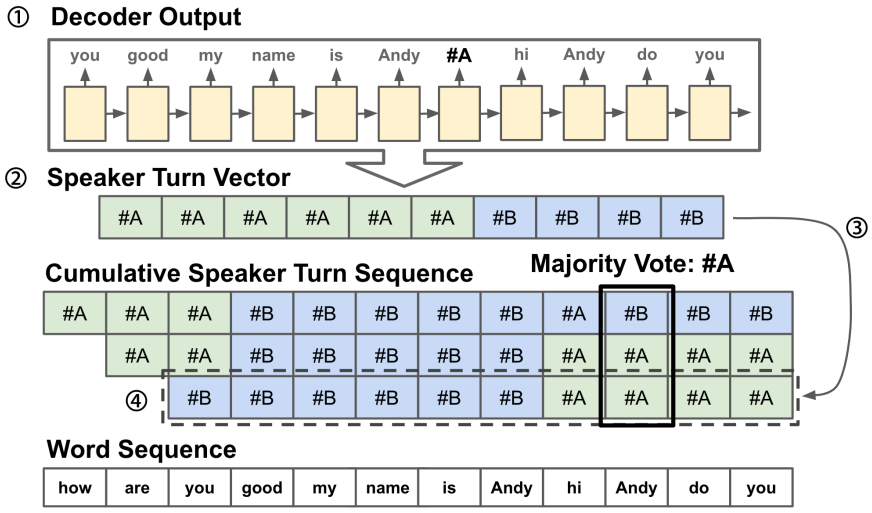


Figure 4.3: Decoder output and overlapping speaker turn vectors.

4.1.5 Experimental Results

Our proposed system is tested with two different lexical data: transcription and ASR output. We trained our system on Fisher English Training Speech Part 1 and Part 2 [20]. For experiment with ASR output, we used Switchboard-1 Telephone Speech Corpus [35] as a testset. The datasets we used in this set of experiments contain word level alignments and speaker turn level alignment information. We created ground truth diarization labels and evaluated the performance of the proposed system. The second experiment was based on ASR output, which is bound to be far less accurate than transcription data. To benchmark the performance of the proposed method, we used the speaker segmentation software in LIUM speaker diarization tool [93] to perform the speaker segmentation task. The software we used performs feature extraction, Speech Activity Detection (SAD) and speaker segmentation sequentially to obtain speaker segmentation result. In the research, we intend to test the diarization performance rather than segmentation accuracy. Therefore, we implemented a BIC based clustering algorithm based on [17] to perform clustering tasks. This clustering algorithm is applied to all of the models in the research.

Before training the proposed system, we have randomly chosen and separated 20 sessions as test set and 567 sessions as dev set from the original Fisher dataset. Thus, in total, we trained 11112 sessions with approximately 19 million words. We used unit training sentence length of 32 words. We trained and tested three different models separately. The first model is trained only on word embeddings only (for convenience, we will refer to it as W model), the second one is trained on both word embeddings and MFCC (WM model) and the third one is trained on word embeddings, MFCC and pitch (WMP model) feature. We trained each model for 20 epochs. Figure 4.4 shows the dev-set accuracy while training. The WM model clearly showed improved performance over W model while WMP model showed very minor improvement over WM model. Note that accuracy in Figure 4.4 is accuracy measured between word sequences that

contain speaker turn tokens and output from the decoder. Thus, this accuracy does not always mean a superior segmentation or diarization accuracy.

The first experiment is the performance test based on transcription data. For transcript based experiment, MFCCs are obtained within the word boundary using the word alignment from transcript. Thus, we use 100% accurate word embedding and temporal information of each word. Table 4.1 shows the result we obtained from transcript data. The result clearly shows that integrating MFCC features helps the performance of diarization when word embedding and temporal information is 100% correct. We also tested the diarization system with ground truth speaker label per word and it showed 16.22% and 18.06% for Fisher and Switchboard data respectively. This is due to the frequent overlaps in dialogues and inaccurate labeling of speaker turn level transcript data. Therefore, the "Ideal" DER is the best performance we can achieve with word level. To check the performance of the proposed system from different angles, we also measured Word-level Diarization Error Rate (WDER) which means "who says this word". Table 4.2 shows WDER result for transcript based experiment. WDER also shows similar results with DER result where WM model and WMP model shows nearly 4% improvement over W model.

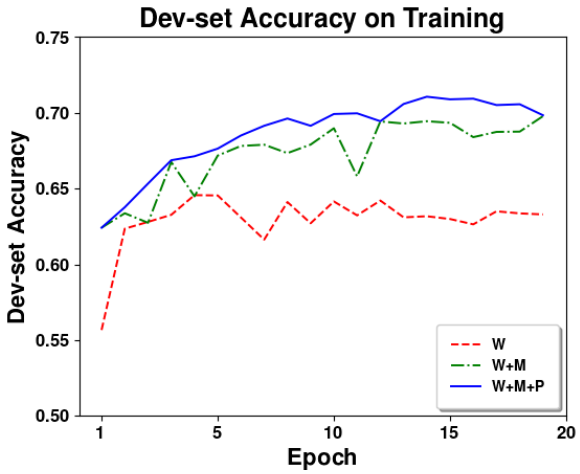


Figure 4.4: Training and validation set accuracy during training.

Table 4.1: DER on transcription data.

DER(%)	**W**	**WM**	**WS**	Oracle	**LIUM**
Fisher	28.02	24.26	44.53	16.22	77.45
Switchboard	27.89	22.44	46.4	18.06	66.57

Table 4.2: WDER on transcription data.

WDER(%)	**W**	**WM**
Fisher Transcript	16.42	12.32
Switchboard Transcript	12.4	8.56

For ASR transcript, we use the Kaldi Speech Recognition Toolkit [83] and ASR model trained on whole Fisher English Speech data. As a test-set, we choose the 30 audio files that have the lowest index in each of 30 folders in the Switchboard-1 dataset for reproducibility of our experiment. Table 4.3 shows the result from the ASR based experiment. Unlike in the case of reference transcripts, WM models did not improve the performance. However, ASR based results are still better than diarization based on segmentation results obtained from LIUM Speaker Diarization Tools. Moreover, WS model also performed better than LIUM Speaker Diarization Tools, which shows that using word-level segmentation from ASR can outperform the BIC based segmentation system.

Since we test the improvement by incorporating acoustic cues with transcript data, performance degradation in the experiment with ASR transcript is entirely caused by poor ASR Word Error Rate (WER). The average WER for 30 Switchboard sessions is 35.15%. Fig. 4.5 shows the scatter plot between WER vs DER for the experiment with ASR transcript (Table 4.3). As depicted in Fig. 4.5, no session shows good speaker diarization performance when WER is high. However, in some cases, although WER is pretty low DER can be very high. Based on this outcome, we could conclude that low WER is a necessary condition for low DER, not the sufficient condition.

Table 4.3: DER on ASR transcript and baseline system.

DER(%)	**W**	**WM**	WS	Oracle	**LIUM**
Switchboard ASR	38.64	50.95	46.02	18.06	66.57

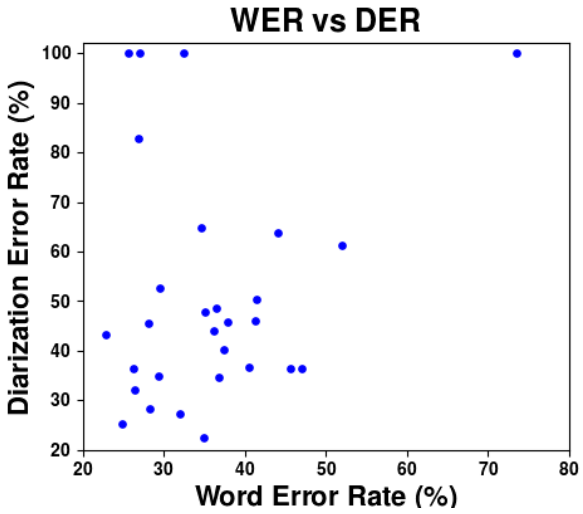


Figure 4.5: Scatter plot of WER vs DER.

The two experiments on transcript and ASR output with the proposed system show that ASR performance hugely affects the performance of DER. However, the experiment with transcript still shows that acoustic cues can improve the diarization performance. From the experimental results, we can conclude that acoustic cues can be integrated with lexical cues but the ASR performance is critical. In addition, the training-set and test-set mismatch also affected the degradation with the ASR output since the proposed model is only trained on the ground-truth transcript, not the actual ASR output.

In this research, we investigated the way to integrate lexical cues and acoustic cues to improve speaker diarization performance. The experiment result showed that if word embeddings and word alignment information are accurate, we can improve the speaker diarization system by incorporating lexical cues and acoustic cues. However, in real life scenario, ASR performance plays a crucial role to the performance of the proposed system and poor WER degrades the proposed

system trained on both acoustic features and word embeddings. In the following section, we introduce a system that incorporates the lexical information directly into the clustering process.

4.2 Multimodal Diarization in Clustering Phase

4.2.1 Acoustic and Lexical Modalities

In the research we introduced in the previous chapter, we focused only on the segmentation process in the research we introduced in the previous section [73]. In this chapter, we introduce the exploitation of lexical information provided by an ASR system to a *speaker clustering* process in speaker diarization. Thus, the lexical information is more directly integrated with acoustic information and influences the performance of the speaker diarization result. The challenge of employing lexical information to speaker clustering is multifaceted and requires practical design choices. In our proposal, we use *word-level speaker turn probabilities* as lexical representation and combine them with acoustic vectors of *speaker embedding* when performing *spectral clustering* [125]. In order to integrate lexical and acoustic representations in the spectral clustering framework, we create *adjacency matrices representing lexical and acoustic affinities between speech segments respectively* and combine them later with a per-element max operation. It is shown that the proposed speaker diarization system improves a baseline performance on two evaluation datasets.

The data flow of the proposed multimodal speaker diarization system is depicted in Fig. 4.6. In the proposed system, there are two streams of information: lexical and acoustic. On the lexical information side, we use the automated transcripts with the corresponding time stamps for word boundaries from an available ASR system. This text information, which is a sequence of words with timestamps, is passed to the speaker turn probability estimator to compute word-level speaker turn probabilities. On the acoustic information side, we perform a general speaker diarization task that is only based on acoustic signal. In this general speaker diarization module, MFCCs are extracted from the input speech signal after speech activity detection (SAD). Following SAD, we uniformly segment the SAD outputs. These uniform segments are relayed to the speaker

embedding extractor that provides speaker embedding vectors. We use the publicly available Kaldi ASpIRE SAD Model[1] [83] for SAD in the proposed diarization pipeline.

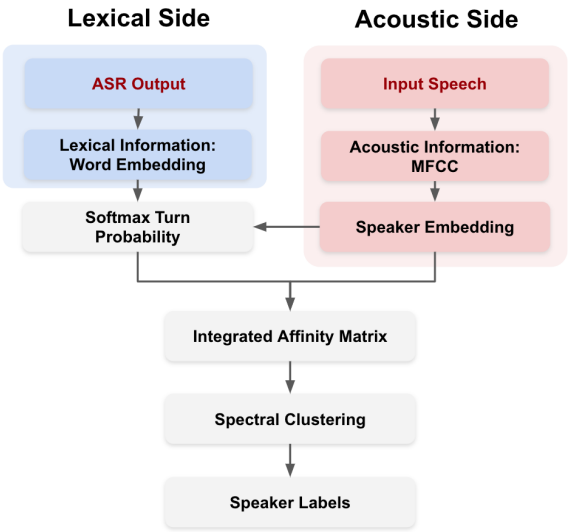


Figure 4.6: Data flow of the proposed system.

After processing the two streams of information, we create two adjacency matrices which hold lexical as well as acoustic affinities between speech segments, respectively, and combine them with a per-element max operation to generate the integrated affinity matrix. Thus, the integrated matrix contains lexical and acoustic information in a comprehensive space. With the integrated adjacency matrix, we obtain speaker labels using a spectral clustering algorithm.

4.2.2 Acoustic Information Stream: Speaker Embedding Extractor

We employ the x-vector model[2] [106] as our speaker embedding generator that showed the state-of-the-art performances for speaker verification and diarization tasks. For windowing of the speech

[1] http://kaldi-asr.org/models/m4
[2] http://kaldi-asr.org/models/m6

signal, we use 0.5 second window, 0.25 second shift and 0.5 second minimum window size for 23-dimensional MFCCs. Note that we do not focus on improving the the performance of speaker embedding since it is out of the scope of this research.

4.2.3 Lexical Information Stream: Speaker Turn Probability Estimator

While acoustic speaker characteristics can be used for speaker turn detection tasks [14], our proposal of word-level speaker turn probability estimation comes behind the reasoning that lexical data can also provide an ample amount of information for similar tasks. It is likely for words in a given context (i.e., utterance) to have different probabilities on whether speaker turns change at the time of being spoken. For example, the words in the phrase "how are you" are very likely to be spoken by a single speaker rather than by multiple speakers. This means that each word in this phrase "how are you" would likely have lower speaker turn probabilities than the word right after the phrase would have. In addition to lexical information, we also fuse a speaker embedding vector per each word to increase the accuracy of the turn probability estimation.

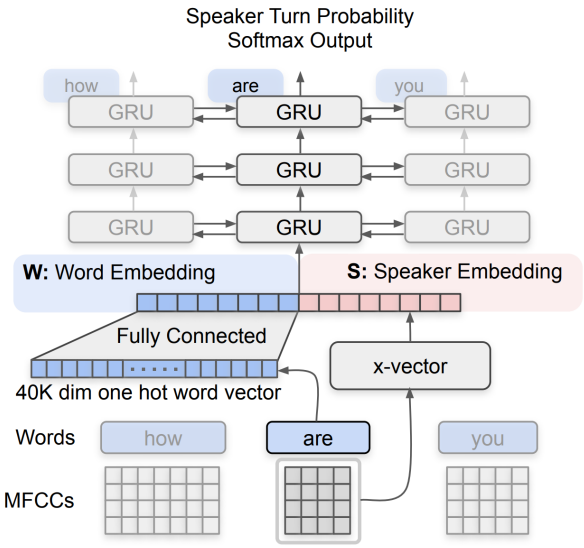


Figure 4.7: Illustration of the proposed speaker turn probability estimator.

To estimate speaker turn probability, we train bi-directional three-layer gated recurrent units (GRUs) [19] with 2,048 hidden units on the Fisher [20] and Switchboard [35] corpora using the force-aligned texts. The actual inputs to the proposed speaker turn probability estimator would be the decoder outputs of the ASR. The words and the corresponding word boundaries are used to generate word embedding and speaker embedding vectors respectively, as follows:

- **Speaker embedding vector (S)**: With the given start and end timestamps of a word from ASR, we retrieve the speaker embedding vector using the speaker embedding extractor described in Section 4.2.2. The x-vector speaker embedding is 128-dimensional.
- **Word embedding vector (W)**: We map the same word input to a 40K-dimensional one-hot vector, which is fully connected to the word embedding layer shown in Fig. 4.7. The dimension of the embedding layer is set to 256.

These two vectors are appended to make a 384-dimensional vector for every word and fed to the GRU layer. The softmax layer has one node and, during inference, outputs speaker turn probability. The parameters of the speaker turn probability estimator are trained with the cross entropy loss. The ASR system used in the research for decoding is the Kaldi ASpIRE recipe[3] [83] that is publicly available.

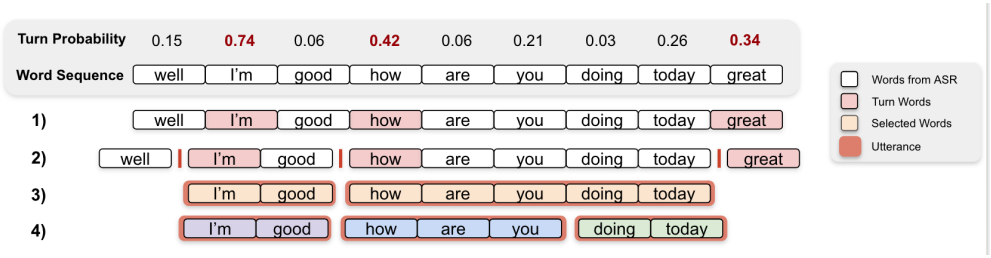


Figure 4.8: Example of the word sequence processing for the adjacency matrix calculation using the speaker turn probabilities.

[3] http://kaldi-asr.org/models/m1

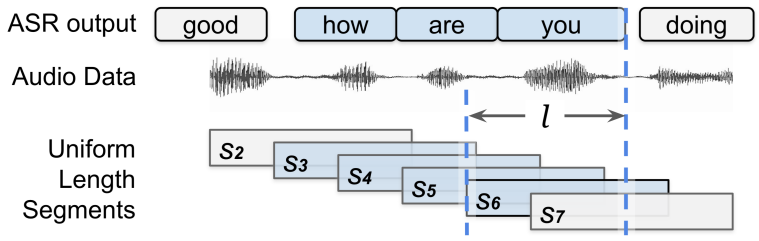


Figure 4.9: Example of the speech segment selection process using the utterance boundary information.

4.2.4 Adjacency Matrix Calculation and Integration

The most challenging part of integrating speaker turn probabilities (from lexical information) and speaker embedding vectors (from acoustic information) in the spectral clustering framework is the heterogeneity of the information sources for these representations. That is, the two quantities do not share any common ground that could be used to measure one quantity against the other. To tackle this challenge, we first create two independent adjacency matrices that contain lexical and acoustic affinities between speech segments, respectively, and then combine them with a per-element max operation to handle the two different types of information from the two different sources. For each adjacency matrix, we employ undirected graphs to represent the corresponding affinities between the speech segments.

- **Adjacency matrix using speaker embeddings**

1) Initially compute the cosine similarity $p_{i,j}$ between speaker embedding vectors for segments s_i and s_j to form the adjacency matrix $\mathbf{P}$, where $1 \leq i, j \leq M$ and M is the total number of segments in a given audio signal.

2) For every i-th row of $\mathbf{P}$, update $p_{i,j}$ as follows:

$$p_{i,j} = \begin{cases} 1 & \text{if } p_{i,j} \leq W(r) \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

where $W(r)$ is the cosine similarity value that is at r-percentile in each row and r is optimized on the dev set. This operation converts $\mathbf{P}$ to a discrete-valued affinity matrix through N nearest neighbor connections.

3) Note that $\mathbf{P}$ is asymmetric and can be seen as an adjacency matrix for a directed graph where each node represents a speech segment in our case. As spectral clustering finds the minimum cuts on an *undirected* graph in theory [125], we choose an undirected version of $\mathbf{P}$, $\mathbf{P_{ud}}$, as the adjacency matrix for speaker embeddings by averaging $\mathbf{P}$ and $\mathbf{P^T}$ as below:

$$\mathbf{P_{ud}} = \frac{1}{2}(\mathbf{P} + \mathbf{P^T}) \tag{4.2}$$

The pictorial representation of $\mathbf{P_{ud}}$ is given on the left side of Fig. 5.

- **Adjacency matrix using speaker turn probabilities**

The following steps 1) to 4) match to the numbered illustrations in Fig. 4.8, where $c = 0.3$ and $\nu = 3$ are given as example parameters.

1) For a given threshold c, pick all the *turn words* that have speaker turn probabilities greater than c in the word sequence provided by ASR. The colored boxes in Fig. 4.8-1) indicate the turn words. The threshold c is determined by the eigengap heuristic that we will discuss in Section 5.2.

2) Break the word sequence at every point where the turn word starts as in Fig. 4.8-2). The given word sequence is broken into multiple utterances.

3) Select all the utterances that have more than one word since a duration spanning one word may be too short to carry any speaker-specific information. For example, in Fig. 4.8-3), the words "well" and "great" are thus not considered for further processing. Additionally, we always consider the seven back channel words ("yes", "oh", "okay", "yeah", "uh-huh", "mhm", "[laughter]") as independent utterances regardless of their turn probabilities.

4) To mitigate the effect of any miss detection by the speaker turn probability estimator, we perform over-segmentation on the utterances by limiting the max utterance length to ν. In Fig. 4.8-4), the resulting utterances are depicted with different colors. Maximum utterance length ν is optimized on the dev set in the range of 2 to 9.

5) Find all the speech segments that fall into the boundary of each utterance. Fig. 4.9 explains how speech segments within the boundary of the example utterance "how are you" are selected and matched. If a segment partly falls into the utterance boundary and its overlap (l in Fig. 4.9) is greater than 50% of the segment length, the segment is considered to fall into the utterance boundary.

6) Let s_m be the first segment and s_n be the last segment falling into the utterance boundary (e.g., segments s_3 and s_6, respectively, in Fig. 4.9). For the elements $q_{i,j}$ in an adjacency matrix $\mathbf{Q_c}$ (with the threshold c) being initialized with zeros, we do the following operation for all the utterances:

$$q_{i,j} = \begin{cases} 1 & \text{if } \; m \leq i, j \leq n \\ q_{i,j} & \text{otherwise} \end{cases} \tag{4.3}$$

The right side of Fig. 4.10 shows an example of $\mathbf{Q_c}$ by the utterance "how are you" in Fig. 4.9.

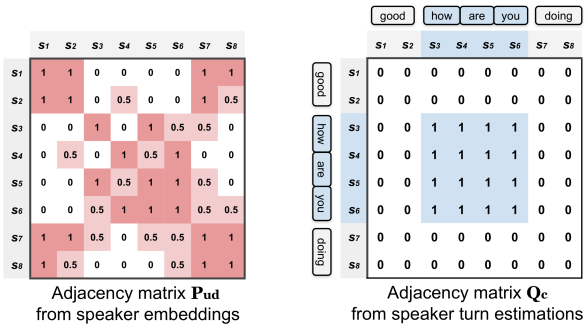


Figure 4.10: Examples of the two adjacency matrices.

- **Combining adjacency matrices**

We combine the two adjacency matrices:

$$\mathbf{A_c} = \max(\mathbf{P_{ud}}, \mathbf{Q_c}) = \max(\frac{\mathbf{1}}{\mathbf{2}}(\mathbf{P} + \mathbf{P^T}), \mathbf{Q_c}) \tag{4.4}$$

where max is a per-element max operation.

We employ a spectral clustering algorithm to obtain the speaker clusters from the integrated affinity matrix. In spectral clustering, the Laplacian matrix is employed to get the spectrum of the given adjacency matrix. We employ the unnormalized graph Laplacian matrix $\mathbf{L_c}$ [125] as below:

$$\mathbf{L_c} = \mathbf{D_c} - \mathbf{A_c} \tag{4.5}$$

where $\mathbf{D_c} = \text{diag}\{d_1, d_2, ..., d_M\}$, $d_i = \sum_{k=1}^{M} a_{ik}$ and a_{ij} is the element in the i^{th} row and j^{th} column of the adjacency matrix $\mathbf{A_c}$. We calculate eigenvalues from $\mathbf{L_c}$ and set up an eigengap vector $\mathbf{e_c}$:

$$\mathbf{e_c} = [\lambda_2 - \lambda_1, \lambda_3 - \lambda_2, \cdots, \lambda_M - \lambda_{M-1}] \tag{4.6}$$

where λ_1 is the smallest eigenvalue and λ_M is the largest eigenvalue. The resulting adjacency matrix $\mathbf{A_c}$ is passed to the spectral clustering algorithm, for which we use the implementation in [81].

The number of clusters (in our case, number of speakers) is estimated by finding the arg max value of the eigengap vector $\mathbf{e_c}$ as in the following equation:

$$\widehat{n_s} = \arg\max_{n}(\mathbf{e_c}) \tag{4.7}$$

where $\widehat{n_s}$ refers to the estimated number of speakers.

Table 4.4: DER (%) on the RT03-CTS dataset.

Number of Speakers		Unknown			
Dataset Split(Quantity)		**Dev**(14)		**Eval**(58)	
Quan *et al.* [127] System SAD		-	-	12.3	3.76
Baseline	**M1**	4.00	1.03	6.97	2.90
Proposed	**M2** W	3.97	1.00	5.19	1.93
	M3 W+S	3.79	0.82	5.11	1.85

We evaluate the proposed system (M3) with the baseline system configuration (M1) on the two evaluation datasets (CH-109 and CH-Eval) as well as the RT03-CTS dataset. To evaluate the systems in terms of diarization error rate (DER) and speaker error rate (SER), we use the *md-eval* software presented in [26]. The gap between DER and SER originates from the false alarms and missed detections that are caused by SAD. The systems compared in the tables above are configured in the following manners:

- **M1**: This baseline system configuration only exploits $\mathbf{P_{ud}}$ as $\mathbf{A_c}$ for spectral clustering (i.e., $\mathbf{A_c} = \mathbf{P_{ud}}$). This is the general speaker diarization system utilizing acoustic information only in speaker embeddings. The results of this system would contrast how much lexical information can contribute to the speaker clustering process to enhance the overall speaker diarization accuracy in M2 and M3.
- **M2**: This configuration for the proposed system excludes the speaker embedding part for the speaker turn probability estimator in Fig. 2 to show the contribution of lexical information in the speaker turn probability estimation process.
- **M3**: This is the full-blown configuration, as explained throughout this chapter.

4.2.5 Experimental Results

The performance of the proposed system is compared to previously published results [127, 131] on the same dataset. However, it should be noted that results in [127] and the proposed system

Table 4.5: DER (%) on the CHAES dataset.

Number of Speakers		Unknown		Known	
Dataset(Quantity)		**CH-Eval**(20)		**CH-109**(109)	
Error Type		**DER**	**SER**	**DER**	**SER**
Quan *et al.* [127] System SAD		12.54	5.97	12.48	6.03
Zajíc *et al.* [131] Oracle SAD		-	-	-	7.84
Baseline	**M1**	7.00	2.94	6.42	2.13
Proposed	**M2** W	7.04	2.97	5.96	1.67
	M3 W+S	6.97	2.9	6.03	1.73

are based on system SAD that is bound to give higher DER than the systems based on oracle SAD. On the other hand, the system in [131] uses oracle SAD which makes DER equal to SER.

- **Table4.4 (RT03-CTS):** The M3 system improves the performance over M2, but the relative improvements are minimal as compared to the improvements of M2 over M1. This shows that most of the performance gain by the proposed speaker diarization system comes from employing lexical information to the speaker clustering process.

- **Table4.5 (CH-Eval, CH-109):** This table compares the proposed speaker diarization system with the recently published results [127, 131] on the CHAES datasets. For a fair comparison, we applied the eigengap analysis based speaker number estimation in Eq. (3.17) only to the CH-Eval dataset while fixing the number of speakers to 2 in the CH-109 dataset (since CH-109 is the chosen set of the CHAES conversations with only 2 speakers). It is shown in the table that the proposed system (M3) outperforms the previously published results in [127, 131] on both CH-Eval and CH-109. It is worthwhile to mention that the proposed system did not gain the noticeable improvement in the CH-Eval dataset as compared to the baseline configuration (M1). As for the CH-109 dataset, on the other hand, M3 seems to provide a noticeable jump in SER over M1. Given our observation that in the CH-109 evaluation most of the performance improvement from M1 to M3 was from the worst 10 sessions that the baseline system performed poorly on, we presume that the proposed system improves the clustering results on such challenging data.

The experimental results show that the baseline system outperforms the previously published results due to the performance of ASpIRE SAD [83] and x-vector [106]. However, the proposed system still improves the competitive baseline system by 36% for RT03-Eval and 19% for CH-109 in terms of SER. As we discussed in this chapter, the experimental results based on the multimodal segmentation approach showed that the proposed system provides meaningful improvements on both of the CHAES and RT03-CTS datasets outperforming the baseline system which is already competitive against the previously published state-of-the-art results. This supports our claim that lexical information can improve diarization results by incorporating turn probability and word boundaries. We inform that some passages, formulas and figures in this chapter have been reused from [76].

Chapter 5

Proposed Speaker Diarization Framework

In this chapter, we propose a speaker diarization system that takes advantage of multimodal information and self-guided clustering approach. The system that we propose in this dissertation is partially based on the study [75] we introduced in the previous chapter. Our proposed speaker diarization framework consists of two main parts: Acoustic side and lexical side. Fig. 5.1 describes the overall structure and the data flow in our proposed system. Unlike the speaker diarization system we introduced in the Chapter 4, we apply multi-scale speaker diarization approach [80] and auto-tuning spectral clustering [77] in the proposed speaker diarization system to build self-guided multimodal speaker diarization system.

The acoustic part of the system, which is described on the left side of the Fig. 5.1, is basically the same as the conventional speaker diarization system where segmentation is led by speaker embedding extraction. An affinity matrix calculated from the extracted speaker embedding is passed to the affinity integration module to be integrated with speaker representation affinity matrix. On the other hand, on the lexical side of the system which is described on the right side of Fig. 5.1, the word sequence from the input transcript is converted to one-hot vectors then fed into a neural language model that estimates the speaker turn probability for each word. The speaker turn probabilities are matched with the speech segments based on the timestamps that can specify the word boundaries. Thus, we group the speech segments that belong to the word

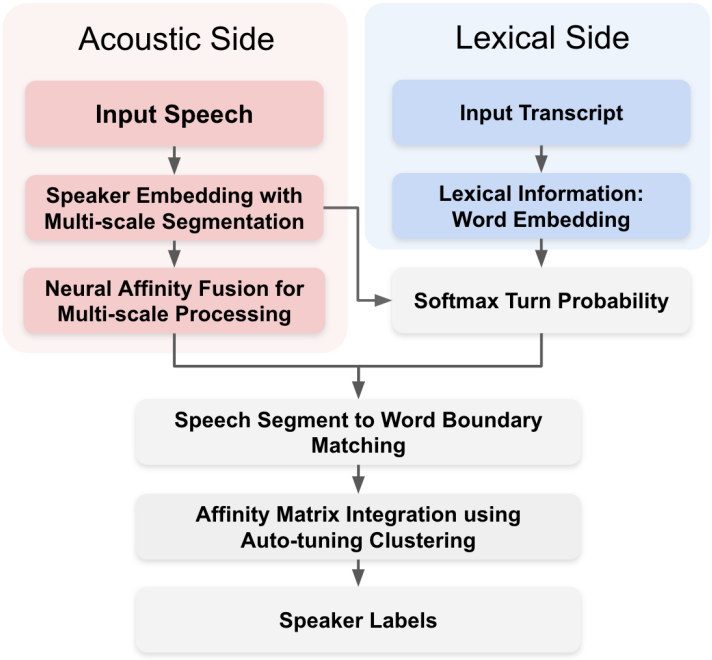


Figure 5.1: Overall structure of the proposed multimodal speaker diarization system.

groups that are very likely to be spoken by a single speaker and represent this information in a lexical information matrix. Finally, we integrate the speaker representation affinity matrix and lexical information matrix to create a final affinity matrix that can provide graph information for spectral clustering. After the clustering process, we get the labels from the cluster labels. In the following sections, the modules we mentioned above will be described in detail. The Acoustic information side of the proposed speaker diarization system is almost identical to the speaker diarization system we described in the previous work [77]. However, there is a distinct difference in terms of segmentation and affinity value processing since we employ the multi-scale speaker diarization system [80] in our proposed system.

5.1 Multi-scale Speaker Diarization

Extracting an authentic speaker representation from a short audio segment is a very challenging task. In general, speaker representations of short speech segments (less than 0.5 second) are known

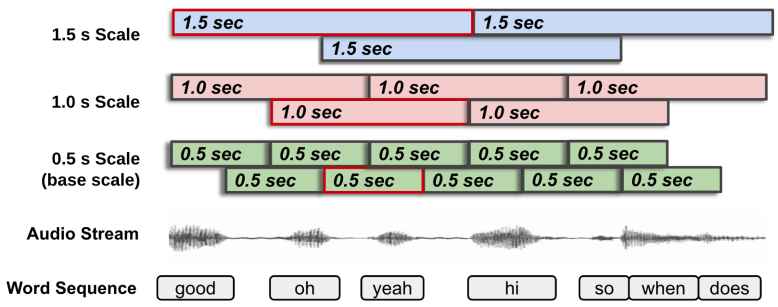


Figure 5.2: Three different segment scales aligned with an audio stream and a word sequence.

to be very unreliable and show a poor performance on speaker recognition tasks. In the previous chapter, we introduced the novel method [80] that deals with the trade-off between temporal resolution and the quality of the speaker representations. The proposed multi-scale diarization approach in [80] employs multiple discrete segment lengths and proposes a method to integrate the given multiple scales. The multi-scale speaker diarization system employs Neural Affinity Score Fusion (NASF) model that can attentively weigh the affinity from the multiple scales depending on the domain and characteristics of the given speech signal. Fig. 5.2 shows the dataflow of the multi-scale speaker diarization system. For the multi-scale processing, we need to generate multiple streams of information for each scale. We employ three scales, 1.5 s, 1.0 s and 0.5 s as in [80]. As shown in the experiment section in [80], multi-scale approach outperforms a single-scale diarization system while having the same dataset and speaker embedding extractor. It is worth emphasizing that we employ the multi-scale approach for having a better resolution for matching the speech segments to word boundaries, as we will state in the following section.

5.2 Segment-word Matching

The shorter segment scale has an integral benefit for matching word boundaries to the existing speech segments. Fig. 5.2 shows the three scales we use and the input waveform displayed with the word boundaries. As shown in the diagram, 1.5 s scale can contain up to 4 words in a single segment. On the other hand, 0.5 s segment has relatively lower chance to contain multiple

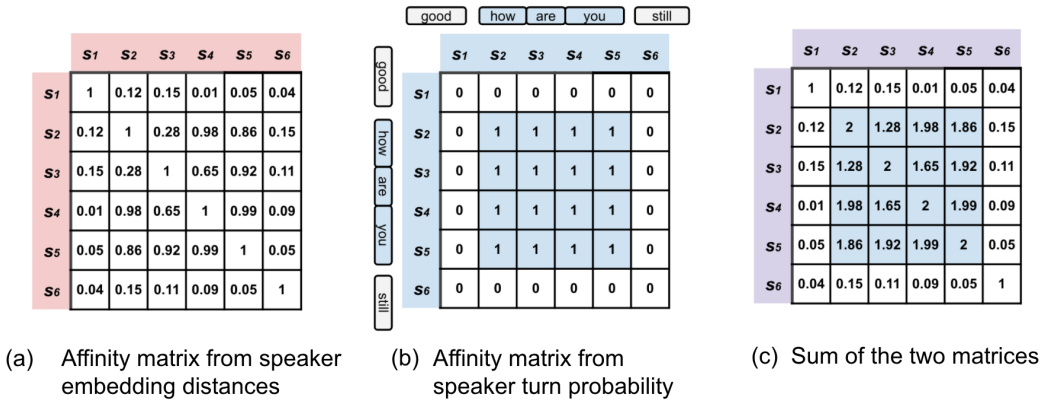


Figure 5.3: Example diagram: (a) Affinity matrix $\mathbf{P}$ from the similarities between speaker embeddings. (b) Affinity matrix $\mathbf{Q}$ from speaker turn probabilities.

words in a single segment. Since the temporal resolution of the multi-scale diarization system is determined by the finest scale, in this case 0.5 s, having shorter scales will eventually help the speaker diarization system to more precisely match the word boundaries with the given audio segments. We will show this benefit in the experimental section.

5.3 Matrix Integration Using NME-SC Method

In this section, we describe how the two affinity matrices are integrated by the NME-SC method instead of using max operation we used in our previous work [75]. We calculate the sum of two affinity matrices: Affinity matrix $\mathbf{P}$ form Eq. 3.9 and affinity matrix $\mathbf{Q}$ that is calculated by speaker turn probabilities. Thus, the sum of matrix is $\mathbf{A}$ as follows:

$$\mathbf{A} = \mathbf{P} + \mathbf{Q}. \tag{5.1}$$

However, adding two matrices is not enough since using the raw affinity values without processing it leads to a poor clustering performance. Thus, we apply NME-SC clustering using the affinity

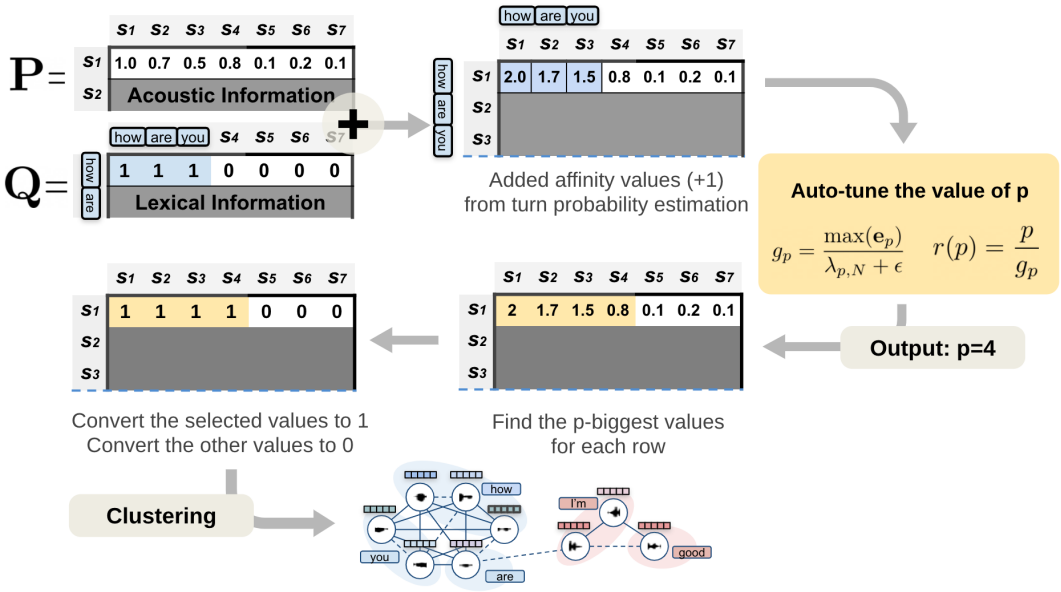


Figure 5.4: Example diagram of Acoustic affinity matrix, lexical information matrix and fused affinity matrix. The word sequence is matched with the speech segments using the time stamps.

matrix $\mathbf{A}$. To obtain eigenvalues and eigenvectors, we calculate a Laplacian matrix for every p value of the affinity matrix $\mathbf{A}$ as follows:

$$\begin{aligned} d_i &= \sum_{k=1}^{N} a_{ik} \\ \mathbf{D}_p &= \mathrm{diag}\{d_1, d_2, ..., d_N\} \\ \mathbf{L}_p &= \mathbf{D}_p - \bar{\mathbf{A}}_p. \end{aligned} \tag{5.2}$$

We calculate eigenvalues from $\mathbf{L_p}$ and then build an eigengap vector $\mathbf{e_p}$:

$$\mathbf{e_c} = [\lambda_2 - \lambda_1, \lambda_3 - \lambda_2, \cdots, \lambda_N - \lambda_{N-1}] \tag{5.3}$$

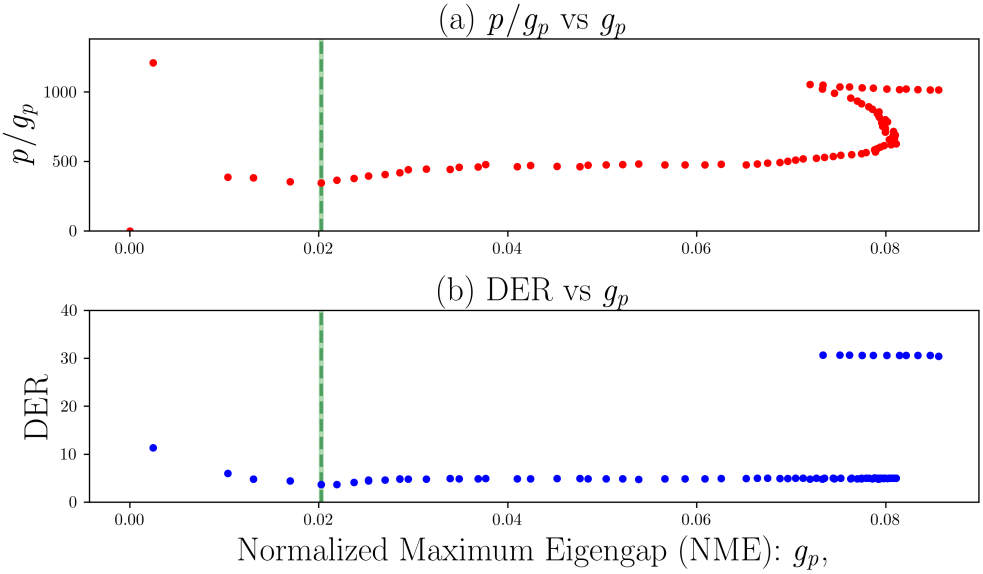


Figure 5.5: Example plot showing the trend between (a) p/g_p versus g_p plot and (b) DER versus g_p plot. The green dotted line shows the g_p value that returns the minimum value for both p/g_p and DER.

where λ_1 is the smallest eigenvalue and λ_M is the largest eigenvalue. The number of clusters, which is assumed to be number of speakers, is estimated by finding the arg max value of the eigengap vector $\mathbf{e_p}$ as follows:

$$\widehat{n_s} = \arg\max_n(\mathbf{e_p}) \tag{5.4}$$

where $\widehat{n_s}$ represents the estimated number of speakers. Using the eigengap values in $\mathbf{e_p}$ To compare the size of the maximum eigengap $\max(\mathbf{e}_p)$ with the range of different p values, we normalize $\max(\mathbf{e}_p)$ with the maximum eignenvalue $\lambda_{p,N}$ to calculate the normalized size of $\max(\mathbf{e}_p)$. Therefore, the NME value g_p for the given binarization parameter p is defined as follows:

$$g_p = \frac{\max(\mathbf{e}_p)}{\lambda_{p,N} + \epsilon}, \tag{5.5}$$

where $\lambda_{p,N} = \max(\mathbf{\Lambda}_p)$ and ϵ is an extremely small value (ϵ=1×10^{-10}) for avoiding instability. We obtain the ratio $r(p)$ between the binarization threshold p and the NME value g_p as follows:

$$r(p) = \frac{p}{g_p}. \tag{5.6}$$

The $r(p)$ value of an example session is plotted against g_p in Fig.5.5-(a) in red dots. The green dotted line Fig.5.5-(a) represents the minimum of $\frac{p}{g_p}$ value. In Fig.5.5-(b), DER value is plotted against g_p. The minimum DER value is also marked by the green dotted line. We can notice that the overall trend is similar with each other and the g_p value that makes the minimum of $r(p)$ also makes the minimum of DER value. In this way, we find the most desirable p value and then perform spectral clustering on $\mathbf{A_p}$ as in [77]. The process mentioned above is depicted in Fig.5.4 with an example value p=4. After we cluster the speaker representation with NME-SC method, the rest of the speaker assigning and evaluation process follow our previous work [77, 75] and the experimental result will be discussed in the following chapter.

Chapter 6

Experiments and Results

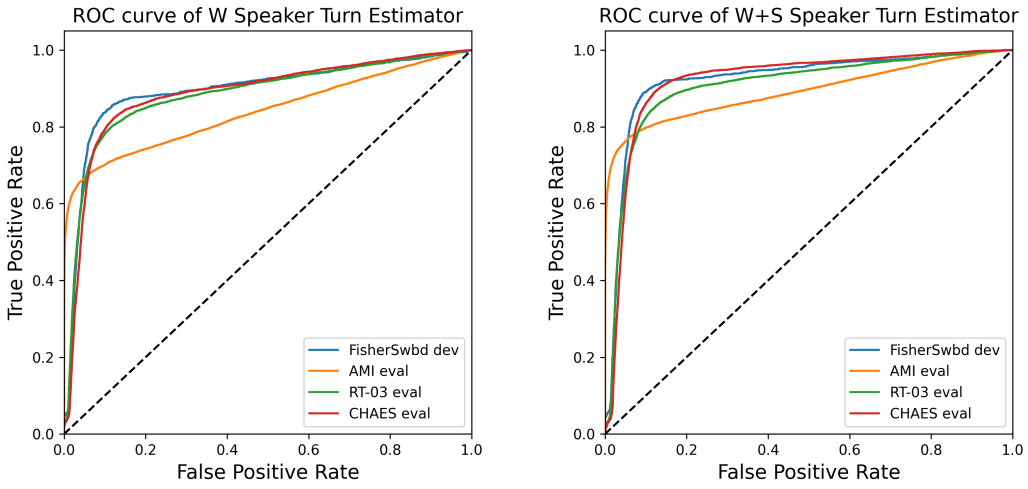
6.1 Datasets

6.1.1 Training Data

We confine the target language to English and train our system on English corpora. We combine the following training datasets to create an integrated word set for our proposed speaker turn estimator. In addition, for speaker turn estimation, the speaker embeddings were also extracted from the same corpora along with word sequences. Henceforth, we refer to the training dataset as *FisherSwbd* for brevity.

- **Fisher English Training Speech** (LDC2004S13, LDC2005S13) Fisher corpus contains telephonic speech of the conversations between two speakers. We use a total of 11699 conversions from Part 1 and Part 2 of Fisher English Training Speech corpora. Fisher corpus contains 2742 hours of speech.

- **Switchboard-1 Release 2** Switchboard [35] dataset contains telephonic speech with a total of 2434 conversations while there are two speakers per conversation. In total, the Switchboard dataset contains 260 hours of speech and 543 speakers.



(a) model W

(b) model W+S

Figure 6.1: ROC curves of speaker turn estimation for each dataset and method.

Table 6.1: Turn probability estimation accuracy based on the ground-truth transcript.

	model W		**model W+S**	
	EER	AUC	EER	AUC
Fisher-Swbd Dev	13.51	0.894	10.49	0.923
CHAES-eval	15.55	0.885	11.89	0.919
RT03	16.56	0.880	13.76	0.902
AMI	24.67	0.839	18.04	0.890

6.1.2 Evaluation Data

The evaluation splits from RT03, CHAES and AMI are used for evaluation of our proposed method. We use the development set splits of the following datasets to tune the parameters of our proposed system.

6.2 Speaker Turn Estimation Results

Speaker turn estimation is the most essential part of our proposed system since it provides the speaker turn probability that contains the information we can extract from the lexical side. Thus,

evaluating the performance of speaker turn estimation is very crucial to analyze the system and figure out which part is contributing to the overall performance. However, if we employ ASR results to evaluate the speaker turn estimation accuracy, it cannot accurately show the performance of the speaker turn estimator since ASR systems are bound to have errors such as missed words and insertions. Thus, we use ground truth transcription to assess the actual contribution of the speaker turn probability estimator and compare that with the ASR based speaker turn estimator.

Fig. 6.1 shows Receiver Operating Characteristic (ROC) curve for model W and model W+S tested on FisherSwbd dev-set, AMI-eval, RT03-eval and CHAES-eval. The two-speaker dataset FisherSwbd dev, RT-03 and CHAES-eval show relatively similar performance compared to AMI-eval. We conjecture that this is due to the overlaps caused by the number of speakers where AMI corpus has more speakers (3 5 speakers) per session and it creates significantly more overlaps among speakers. This can be intuitively observed in Fig. 6.1: For the same true positive rate, AMI-eval shows significantly higher false positive rate. This indicates that the trained speaker turn estimator model overestimates the probability of the speaker turns especially in AMI-eval split.

More importantly, we can check the performance gap between model W and model W+S in Table 6.1. The accuracy of the turn probability estimation is shown in equal error rate (EER) and area under cover (AUC) values that are obtained from Table 6.1. On average, model W+S shows 22.4% of relative EER improvement. The performance gap in AMI-eval showed the biggest improvement from adding speaker embedding to the speaker turn estimator. This shows that the speaker turns in the AMI-eval set is more challenging to be estimated by the word-only system, model W.

Table 6.2: DER (%) based on ground-truth transcription.

Scale	Turn Est. Model	**CHAES**	**RT03**	**AMI**
SS	**No Lex.**	2.48	2.21	3.37
MS	**No Lex.**	2.04	2.17	3.32
SS	**model W**	2.48	2.11	4.02
SS	**model W+S**	2.32	2.11	3.89
MS	**model W**	1.64	1.76	2.94
MS	**model W+S**	**1.47**	**1.68**	**2.78**
MS	**Oracle Lex.**	1.05	1.22	2.07

Table 6.3: WER (%) of evaluation set.

	CHAES-eval	**RT03**	**AMI**
WER (%)	23.27	25.61	34.49

6.3 Diarization Evaluations

The estimated speaker turn probability values are used to obtain the actual speaker diarization output. For calculating diarization error rate (DER), we employ the evaluation schemes and software that appeared in [26]. This evaluation method involves 0.25 second of collar region.

Table 6.2 shows the DER of the diarization systems based on ground truth transcriptions. In the leftmost column, the segmentation types, multi-scale (MS) and single-scale (SS) with 1.5 s segments, is notified. The 1.5 s scale is selected since the 1.5 s segment performs the best for all the datasets and methods when it comes to the single-scale speaker diarization. Oracle Lex. represents the performance we get by providing the ground truth turn probability to the clustering system. Even if we provide the ground truth turn information, errors can be generated since the speaker turn probability only provides what speech segments should be tied together. Therefore, there is a chance that the several speech segments that are grouped by the lexical information can still be incorrectly clustered. Therefore, the performance of Oracle Lex. in Table 6.2 can be regarded as a system that has oracle ASR (ground truth transcript) and a perfect speaker turn estimator that has 0% EER and 1.0 of AUC. No Lex. system represents speaker diarization system without any lexical information. This is identical to the speaker diarization system appeared in

Table 6.4: DER (%) based on ASR outputs.

Scale	Turn Est. Model	**CHAES**	**RT03**	**AMI**
MS	**No Lex.**	22.04	2.17	3.32
MS	**model W**	1.87	2.01	3.25
MS	**model W+S**	1.71	1.86	3.25

[80] where the multi-scale diarization approach is applied with auto-tuning clustering [77]. We use No Lex. as a baseline to compare the performance of our proposed system.

The most consistent trend we notice in the Table. 6.2 is the performance improvement from SS to MS. There are two different factors in the improvement from SS to MS. First, the multi-scale approach itself brings about a certain degree of DER improvement. However, the performance degradation of SS-model-W and SS-model-W+S are explained by the poor temporal resolution of 1.5 s segment length. As we explained in the Fig. 5.2, 1.5 s segments are bound to create more errors than shorter segments. This becomes evident as we evaluate the model with 1.5 s single-scale and compare it to multi-scale based models. Thus, we are able to check that multi-scale approach helps matching the temporal alignment between word boundaries and speech segments and the finer base scale length reduces the error.

More importantly, Model-W and Model-W+S show a good margin of improvement for CHAES-eval and RT-03 dataset. However, even if we use ground truth transcript, the improvement made for AMI-eval was relatively minor. This result is conjectured to be originating from the poor speaker turn probability in Table 6.1. Thus, we can observe that the speaker turn estimation accuracy is affecting the improvement of the model W and model W+S from No Lex. system.
The DER results with ASR output are shown in Table 6.4. Since the quality of ASR largely affects the turn probability estimation, we provide word error rate (WER) of ASR in Table 6.3. We use the trained model from the ASR system in the Kaldi ASpIRE recipe[1] [83] that is openly accessible to the public. In Table 6.4, AMI-eval performance did not show significant improvement from the No Lex. system. This improvement is worse than the ground truth transcription based result

[1]http://kaldi-asr.org/models/m1

(Table 6.2) and this is conjectured to be caused by poor ASR result. The evaluation results from transcription and ASR suggest that the conversations with more speakers are more challenging to benefit from lexical information. However, it must be noted that the result shown in Table 6.4 is heavily dependent on the accuracy of the ASR system. In addition, it is highly likely that the improvement in ASR makes the diarization performance higher since the DER with ground truth transcript is a lot lower than the ASR based DER results. This leaves room for improvement and makes us expect the further improvement of our proposed method as the ASR technologies constantly improve and get more and more accurate.

Chapter 7

Future Work

In the previous chapters, we covered our research endeavors regarding multimodal diarization and self-guided clustering methods. In the future, we plan to add two more research topics. First, we will employ a pre-trained language model that helps the speaker turn-probability estimator capture more sophisticated patterns. Because our proposed speaker turn-probability estimator is trained on a relatively small number of datasets, we plan to employ a language model that is trained with a significant number of text data, thereby benefiting from out-of-domain data. Second, we plan to extend the speaker diarization system to a multi-task system that conducts not only speaker diarization tasks but also audio scene detection or topic classification. The benefit of the additional performance improvement will also be investigated through a multi-task approach.

7.1 Pre-trained Language Model for Multimodal Speaker Diarization

Extending from the multimodality related studies introduced in the previous chapters, we plan to employ a pre-trained language model that can boost the turn probability estimation performance. A proposed language model called Bidirectional Encoder Representations from Transformers (BERT) [23] has demonstrated state-of-the-art performance on most natural language processing (NLP) tasks. BERT is based on a sequence transducer called a transformer [122]. BERT showed that a language model that is trained on a huge amount of text data including Wikipedia and published books. We plan to integrate the encoder part of the BERT language model into our speaker turn-probability estimator network.

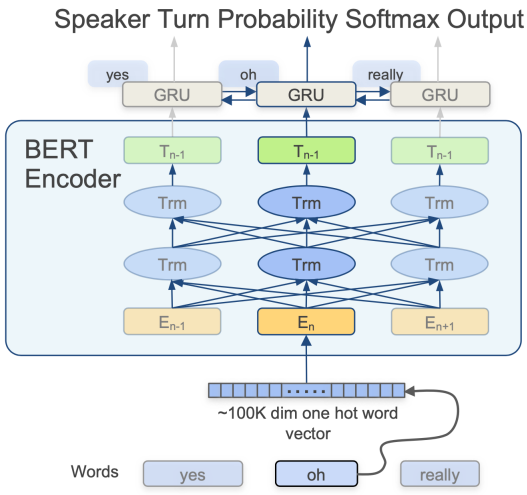


Figure 7.1: Decoder side of the proposed sequence-to-sequence model.

Fig. 7.1 shows the simplified structure of the speaker turn-probability estimator with a pre-trained language model. The multi-layer GRU units are replaced with transformer modules that return encoded embedding to the GRU layer (the final layer). The GRU layer outputs the speaker turn-probability similar to the GRU-based speaker turn-probability estimator. However, there are some expected problems with the employment of the pre-trained language model. First,

BERT is not trained on a conversational speech dataset. Conversational speech datasets usually contain different vocabularies and sentence lengths. Thus, the speaker turn-probability estimation model should be fine-tuned on a conversational dataset and tested with the conversational speech datasets. Second, because BERT has a bidirectional structure, it can be difficult to transform into an online model, whereas RNN models can simply employ a one-directional RNN to conduct the prediction task in an online manner. These problems should be investigated further in a future study.

7.2 End-to-end Speaker Diarization System

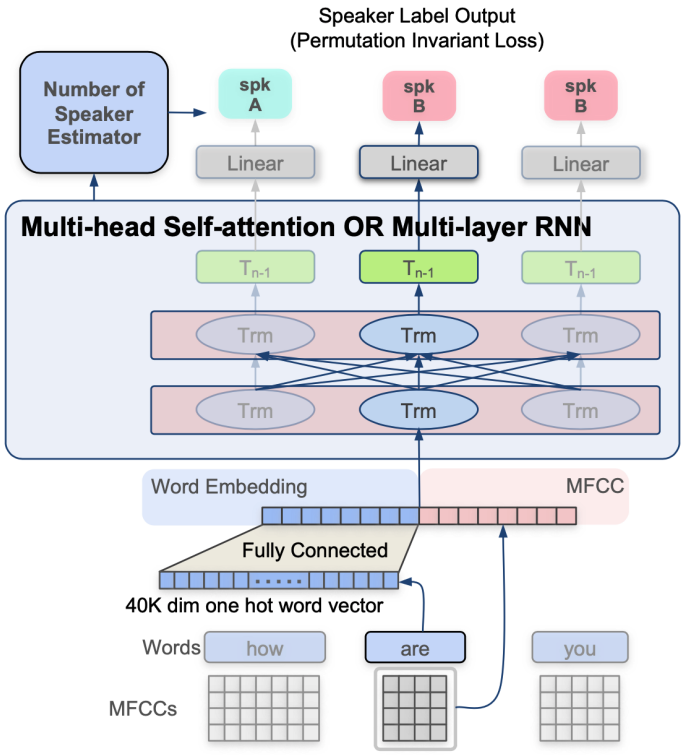


Figure 7.2: End-to-end speaker diarization system with multimodal input

An end-to-end neural network based system has also recently been employed in a diarization system [28]. The benefit of an end-to-end model includes a straight-forward training process and

an easier implementation in a production system. Despite these strengths, end-to-end systems have certain weaknesses. First, an end-to-end system cannot be trained module-by-module and needs to be trained on a dataset with sufficient variability in acoustic conditions, as well as a significant number of vocabularies and speakers. In addition, it is difficult to validate each function (e.g., speaker representation or speaker clustering). Moreover, the accuracy of an end-to-end diarization system has lagged behind that of a modular speaker diarization system [28]. Although there are several shortcomings to an end-to-end approach, we still recognize some benefits for speaker diarization tasks. First, an end-to-end speaker diarization system does not require a

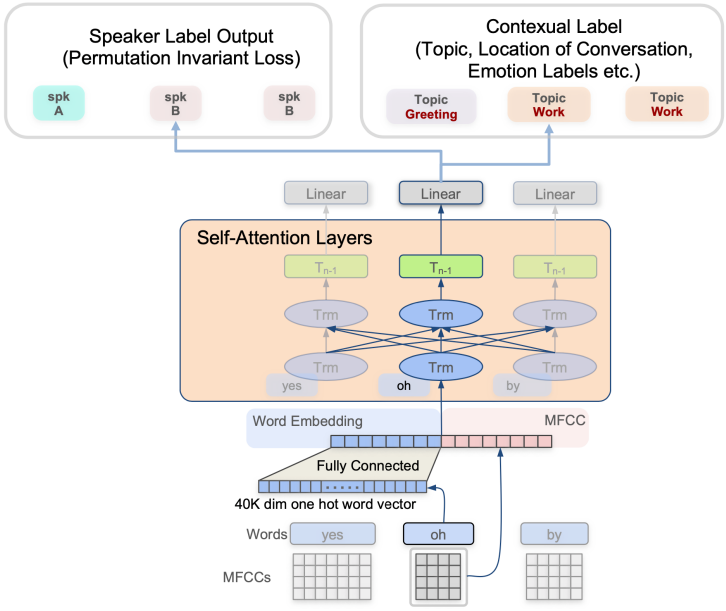


Figure 7.3: End-to-end speaker diarization system with lexical information input

special process for multimodal clustering, as we described in the previous chapter. Instead of employing a relatively complicated segment mapping and matrix fusion, the layers from a speaker representation can be integrated with a lexical layer through a simple concatenation of the layers. Fig.7.2 describes the end-to-end multimodal speaker diarization system. Note that we need a module that estimates the number of speakers of a given utterance because such a number is never given in a real-life scenario. Second, by employing an end-to-end approach, we are able to

train our multimodal speaker diarization system with multiple labels. For example, we can train the speaker diarization system with topic labels in [20], as indicated in Fig.7.3.

The contextual labels in Fig. 7.3 not only operate as an additional loss for training but also as an estimation system for estimating the topic of the conversation. The contextual labels can be emotion or location labels (e.g., in an office, outdoors, or on a telephone call). Thus, we plan to broaden the potential applications of an end-to-end multimodal speaker diarization system such that the multi-task system can also perform tasks such as an audio scene analysis and topic classification.

Chapter 8

Conclusions

In this dissertation, we described our effort to build a speaker diarization system that leverages self-guided clustering methods and multimodal approaches. We proposed novel self-guided clustering approaches. With the NME-SC method, we showed that the performance of the speaker diarization can be significantly improved if we can successfully estimate the parameter for each session rather than using the same parameter for multiple sessions by optimizing the parameters through the development set. Thus, the NME-SC method not only improves the performance but makes a more generalizable speaker diarization system that can cope with variability of the input data. Moreover, we proposed a multi-scale speaker diarization system that determines the weights between multiple scales such that the multi-scale speaker diarization system can improve the temporal resolution while achieving a superior fidelity of the speaker representation. In addition, we showed that the multi-scale speaker diarization method can help match the words to the speech segments by applying a finer temporal resolution.

On the multimodal side, we proposed a multimodal speaker diarization system that can integrate lexical information into a speaker diarization system. Unlike previous studies, our proposed multimodal speaker diarization system involves the lexical information directly in the form of word-by-word speaker turn-probability. Our proposed method is fundamentally different from previous studies in which the lexical information is employed to refine the segmentation result

or to find the phrases that can identify the potential speakers. Moreover, our proposed lexical information integration scheme captures the aspects of a conversation that can never be captured by the traditional acoustic feature based speaker diarization systems. Through this study we discovered the following key findings. First, we showed that the lexical cues can be captured by the speaker-turn probability for a speaker diarization task. The discrepancy between speech segments and word boundaries from ASR has been one of the most frequent problems in speech processing. Second, we showed that a multi-scale approach can help the speaker diarization system achieve a finer base-scale, leading to an improved temporal resolution. We showed that the improved temporal resolution can help the matching between speech segments and word boundaries. Third, we showed that the integration of a lexical information matrix can be achieved through an auto-tuning clustering method. This removes the burden of additional parameter tuning for the clustering process and successfully replaces the maximum operation that we used in a previous study.

However, there are a few limitations to our proposed method. First, the overall system is overly complicated compared to other speaker diarization systems. We need to employ three different neural network models: a speaker embedding extractor, speaker turn estimator, and neural affinity fusion network, each of which needs to be optimized separately and integrated with a parameter tuning. Second, the proposed method does not handle overlapping speech, which is one of the most difficult problems in speaker diarization and the ASR field. In this study, overlapping speech also causes a significant degradation because the turn-probability cannot be accurately estimated in the overlapping regions.

Therefore, further studies should include the following aspects. To reduce the effort for optimizing the parameters or separately training each module, we can investigate an end-to-end model that accepts speech and returns a speaker label. Thus, such an end-to-end model can be a type of jointly modeled system that is simultaneously optimized for the ASR criterion and speaker diarization criterion. However, the end-to-end approach for ASR and speaker diarization requires

a sizable dataset and securing a high-quality dataset to obtain a competitive performance could be a challenge. In addition to an end-to-end system, source separation techniques or a personalized SAD system can be integrated to deal with the overlapping speech problem.

We believe that the future of speaker diarization systems will focus more on the contextual aspect of a conversation. However, even to date, state-of-the-art speaker diarization systems do not consider contextual information and only apply acoustic information. This is largely because the technology for incorporating contextual information has yet to be fully developed and matured to improve the speaker diarization performance. We strongly believe that our study fills in the gap between traditional speaker diarization methods and the contextual speaker diarization systems that will likely appear in the future. Thus, we hope that our research can be a cornerstone for building a contextual speaker diarization system in the future.

Reference List

[1] J. Ajmera, G. Lathoud, and L. McCowan. Clustering and segmenting speakers and their locations in meetings. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 605–608, 2004.

[2] J. Ajmera and C. Wooters. A robust speaker clustering algorithm. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 411–416, 2003.

[3] AMI. Ami consortium. [Online]. Available: `https://www.amiproject.org/index.html`, [Accessed Feb. 3, 2021].

[4] X. Anguera, C. Wooters, and J. Hernando. Purity algorithms for speaker diarization of meetings data. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 1025–1028, 2006.

[5] 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, 2007.

[6] X. Anguera, C. Wooters, B. Peskin, and M. Aguilo. Robust speaker segmentation for meetings: The ICSI-SRI Spring 2005 diarization system. In *Proceedings of Machine Learning for Multimodal Interaction Workshop*, pages 402–414, 2005.

[7] Xavier Anguera, Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals. Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):356–370, 2012.

[8] Ashish Arora, Desh Raj, Aswin Shanmugam Subramanian, Ke Li, Bar Ben-Yair, Matthew Maciejewski, Piotr Żelasko, Paola Garcia, Shinji Watanabe, and Sanjeev Khudanpur. The JHU multi-microphone multi-speaker asr system for the CHiME-6 challenge. *arXiv preprint arXiv:2006.07898*, 2020.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[10] C. Barras, Xuan Zhu, S. Meignier, and Jean-Luc Gauvain. Multistage speaker diarization of broadcast news. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1505–1512, 2006.

[11] Homayoon S.M. Beigi and Stéphane H. Maes. Speaker, channel and environment change detection. In *Proceedings of World Congress of Automation*, 1998.

[12] Paul E. Black. Hungarian algorithm. https://xlinux.nist.gov/dads/HTML/HungarianAlgorithm.html, [Accessed Feb. 3, 2021].

[13] Simon Bozonnet, Nicholas Evans, Xavier Anguera, Oriol Vinyals, Gerald Friedland, and Corinne Fredouille. System output combination for improved speaker diarization. In *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 2642–2645. ISCA, 2010.

[14] Hervé Bredin. Tristounet: Triplet loss for speaker turn embedding. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5430–5434. IEEE, 2017.

[15] Leonardo Canseco-Rodriguez, Lori Lamel, and Jean-Luc Gauvain. Speaker diarization from speech transcripts. In *Proceedings of the International Conference on Spoken Language Processing*, volume 4, pages 3–7, 2004.

[16] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4960–4964. IEEE, 2016.

[17] Scott Chen, Ponani Gopalakrishnan, et al. Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *Proceedings DARPA broadcast news transcription and understanding workshop*, volume 8, pages 127–132. Virginia, USA, 1998.

[18] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[19] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[20] Christopher Cieri, David Miller, and Kevin Walker. Fisher english training speech parts 1 and 2. *Philadelphia: Linguistic Data Consortium*, 2004.

[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), 2011.

[22] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

[23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[24] G. Dupuy, M. Rouvier, S. Meignier, and Y. Esteve. i-Vectors and ILP clustering adapted to cross-show speaker diarization. In *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 2174–2177, 2012.

[25] Yannick Esteve, Sylvain Meignier, Paul Deléglise, and Julie Mauclair. Extracting true speaker identities from transcriptions. In *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[26] Jonathan G Fiscus, Jerome Ajot, Martial Michel, and John S Garofolo. The rich transcription 2006 spring meeting recognition evaluation. In *Proceedings of International Workshop on Machine Learning and Multimodal Interaction*, pages 309–322, May 2006.

[27] Nikolaos Flemotomos, Panayiotis Georgiou, and Shrikanth Narayanan. Linguistically aided speaker diarization using speaker role information. *arXiv*, pages arXiv–1911, 2019.

[28] Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Kenji Nagamatsu, and Shinji Watanabe. End-to-end neural speaker diarization with permutation-free objectives. *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 4300–4304, 2019.

[29] Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Yawen Xue, Kenji Nagamatsu, and Shinji Watanabe. End-to-end neural speaker diarization with self-attention. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 296–303. IEEE, 2019.

[30] Daniel Garcia-Romero, David Snyder, Gregory Sell, Daniel Povey, and Alan McCree. Speaker diarization using deep neural network embeddings. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4930–4934, Mar. 2017.

[31] J.-L. Gauvain, G. Adda, L. Lamel, and M. Adda-Decker. Transcription of broadcast news: The LIMSI Nov 96 Hub4 system. In *Proceedings of ARPA Speech Recognition Workshop*, pages 56–63, 1997.

[32] J.-L. Gauvain, L. Lamel, and G. Adda. Partitioning and transcription of broadcast news data. In *Proceedings of the International Conference on Spoken Language Processing*, pages 1335–1338, 1998.

[33] J.-L. Gauvain, L. Lamel, and G. Adda. The LIMSI 1997 Hub-4E transcription system. In *Proceedings of DARPA News Transcription and Understanding Workshop*, pages 75–79, 1998.

[34] H. Gish, M. . 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*, pages 873–876, 1991.

[35] John J Godfrey and Edward Holliman. Switchboard-1 release 2. *Linguistic Data Consortium, Philadelphia*, 926:927, 1997.

[36] Kyu J Han and Shrikanth S Narayanan. A robust stopping criterion for agglomerative hierarchical clustering in a speaker diarization system. In *Proceedings of the Annual Conference of the International Speech Communication Association*, 2007.

[37] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer. End-to-end text-dependent speaker verification. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5115–5119, 2016.

[38] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[39] Jing Huang, Etienne Marcheret, Karthik Visweswariah, and Gerasimos Potamianos. The ibm rt07 evaluation systems for speaker diarization on lecture meetings. In *Multimodal Technologies for Perception of Humans*, pages 497–508. Springer, 2007.

[40] MAH Huijbregts, DA van Leeuwen, and FM Jong. The majority wins: a method for combining speaker diarization systems. In *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 924–927. ISCA, 2009.

[41] Miquel Àngel India Massana, José Adrián Rodríguez Fonollosa, and Francisco Javier Hernando Pericás. Lstm neural network-based speaker segmentation using acoustic and language modelling. In *INTERSPEECH 2017: 20-24 August 2017: Stockholm*, pages 2834–2838. International Speech Communication Association (ISCA), 2017.

[42] Sergey Ioffe. Probabilistic linear discriminant analysis. In *Proceedings of European Conference on Computer Vision*, pages 531–542, May 2006.

[43] D. Istrate, C. Fredouille, S. Meignier, L. Besacier, and J.-F. Bonastre. NIST RT05S evaluation: Pre-processing techniques and speaker diarization on multiple microphone meetings. In *Proceedings of Machine Learning for Multimodal Interaction Workshop*, 2006.

[44] Q. Jin, K. Laskowski, T. Schultz, and A. Waibel. Speaker segmentation and clustering in meetings. In *Proceedings of the International Conference on Spoken Language Processing*, pages 597–600, 2004.

[45] Youngmoon Jung, Seongmin Kye, Yeunju Choi, Myunghun Jung, and Hoirin Kim. Multi-scale aggregation using feature pyramid module for text-independent speaker verification. *arXiv preprint arXiv:2004.03194*, 2020.

[46] P. Kenny, D. Reynolds, and F. Castaldo. Diarization of telephone conversations using factor analysis. *IEEE Journal of Selected Topics in Signal Processing*, 4(6):1059–1070, 2010.

[47] Patrick Kenny, Gilles Boulianne, and Pierre Dumouchel. Eigenvoice modeling with sparse training data. *IEEE Transactions on Speech and Audio Processing*, 13(3):345–354, 2005.

[48] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel. Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1435–1447, 2007.

[49] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel. Speaker and session variability in gmm-based speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1448–1460, 2007.

[50] Patrick Kenny, Pierre Ouellet, Najim Dehak, Vishwa Gupta, and Pierre Dumouchel. A study of interspeaker variability in speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(5):980–988, 2008.

[51] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *Proceedings of the Workshop on Deep Learning in International Conference on Machine Learning, ICML*, 2015.

[52] Federico Landini, Shuai Wang, Mireia Diez, Lukáš Burget, Pavel Matějka, Kateřina Žmolíková, Ladislav Mošner, Oldřich Plchot, Ondřej Novotnỳ, Hossein Zeinali, et al. But system description for dihard speech diarization challenge 2019. *arXiv preprint arXiv:1910.08847*, 2019.

[53] D. A. V. Leeuwen and M. Konecny. Progress in the AMIDA speaker diarization system for meeting data. In *Proceedings of International Evaluation Workshops CLEAR 2007 and RT 2007*, pages 475–483, 2007.

[54] Qingjian Lin, Ruiqing Yin, Ming Li, Hervé Bredin, and Claude Barras. Lstm based similarity measurement with spectral clustering for speaker diarization. In *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 366–370, 2019.

[55] Qingjian Lin, Ruiqing Yin, Ming Li, Hervé Bredin, and Claude Barras. LSTM based similarity measurement with spectral clustering for speaker diarization. In *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 366–370, Sep. 2019.

[56] D. Liu and F. Kubala. Fast speaker change detection for broadcast news transcription and indexing. In *Proceedings of the International Conference on Spoken Language Processing*, pages 1031–1034, 1999.

[57] D. Liu and F. Kubala. A cross-channel modeling approach for automatic segmentation of conversational telephone speech. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 333–338, 2003.

[58] Stuart Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[59] Bo Long, Zhongfei Mark Zhang, Xiaoyun Wu, and Philip S Yu. Spectral clustering for multi-type relational data. In *Proceedings ofInternational Conference on Machine Learning*, pages 585–592, Jun. 2006.

[60] Robert G Lorenz and Stephen P Boyd. Robust minimum variance beamforming. *IEEE Transactions on Signal Processing*, 53(5):1684–1696, 2005.

[61] J. Luque and J. Hernando. On the use of agglomerative and spectral clustering in speaker diarization of meetings. In *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, pages 130–137, Jun. 2012.

[62] James Lyons. Python speech features. `https://github.com/jameslyons/python-speech-features`, 2017. Accessed: 2018-03-23.

[63] S. Meignier, D. Moraru, C. Fredouille, J.-F. Bonastre, and L. Besacier. Step-by-step and integrated approaches in broadcast news speaker diarization. *Computer, Speech & Language*, 20(2-3):303–330, 2006.

[64] 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*, pages 1017–1020, 2006.

[65] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

[66] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14:849–856, 2001.

[67] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings of Advances in Neural Information Processing Systems*, pages 849–856, Dec. 2002.

[68] Huazhong Ning, Ming Liu, Hao Tang, and Thomas S Huang. A spectral clustering approach to speaker diarization. In *Proceedings of the International Conference on Spoken Language Processing*, pages 2178–2181, 2006.

[69] NIST. Rich transcription evaluation. [Online]. Available: `https://www.nist.gov/itl/iad/mig/rich-transcription-evaluation`, [Accessed Feb. 3, 2021].

[70] Sergey Novoselov, Aleksei Gusev, Artem Ivanov, Timur Pekhovsky, Andrey Shulipa, Anastasia Avdeeva, Artem Gorlanov, and Alexandr Kozlov. Speaker diarization with deep speaker embeddings for dihard challenge ii. In *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 1003–1007, 2019.

[71] Monisankha Pal, Manoj Kumar, Raghuveer Peri, Tae Jin Park, So Hyun Kim, Catherine Lord, Somer Bishop, and Shrikanth Narayanan. Speaker diarization using latent space clustering in generative adversarial network. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6504–6508. IEEE, 2020.

[72] Jose Pardo, Xavier Anguera, and Chuck Wooters. Speaker diarization for multiple-distant-microphone meetings using several sources of information. *IEEE Transactions on Computers*, 56(9):1212–1224, 2007.

[73] Tae Jin Park and Panayiotis Georgiou. Multimodal speaker segmentation and diarization using lexical and acoustic cues via sequence to sequence neural networks. *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 1373–1377, 2018.

[74] Tae Jin Park and Panayiotis Georgy. Multistream diarization fusion using the minimum variance bayesian information criterion. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5224–5228. IEEE, 2018.

[75] Tae Jin Park, Kyu J Han, Jing Huang, Xiaodong He, Bowen Zhou, Panayiotis Georgiou, and Shrikanth Narayanan. Speaker diarization with lexical information. *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 391–395, 2019.

[76] Tae Jin Park, Kyu J Han, Jing Huang, Xiaodong He, Bowen Zhou, Panayiotis Georgiou, and Shrikanth Narayanan. Speaker diarization with lexical information. *arXiv preprint arXiv:2004.06756*, 2020.

[77] Tae Jin Park, Kyu J Han, Manoj Kumar, and Shrikanth Narayanan. Auto-tuning spectral clustering for speaker diarization using normalized maximum eigengap. *IEEE Signal Processing Letters*, 27:381–385, 2019.

[78] Tae Jin Park, Naoyuki Kanda, Dimitrios Dimitriadis, Kyu J Han, Shinji Watanabe, and Shrikanth Narayanan. A review of speaker diarization: Recent advances with deep learning. *arXiv preprint arXiv:2101.09624*, 2021.

[79] Tae Jin Park, Manoj Kumar, Nikolaos Flemotomos, Monisankha Pal, Raghuveer Peri, Rimita Lahiri, Panayiotis Georgiou, and Shrikanth Narayanan. The second dihard challenge: System description for usc-sail team. *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 998–1002, 2019.

[80] Tae Jin Park, Manoj Kumar, and Shrikanth Narayanan. Multi-scale speaker diarization with neural affinity score fusion. *arXiv preprint arXiv:2011.10527*, 2020.

[81] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[82] T. Pfau, D. Ellis, and A. Stolcke. Multispeaker speech activity detection for the ICSI meeting recorder. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 107–110, 2001.

[83] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The Kaldi speech recognition toolkit. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, Dec. 2011.

[84] Simon JD Prince and James H Elder. Probabilistic linear discriminant analysis for inferences about identity. In *IEEE International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[85] D. A. Reynolds, P. Kenny, and F. Castaldo. A study of new approaches to speaker diarization. In *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 1047–1050, 2009.

[86] D. A. Reynolds and P. Torres-Carrasquillo. The MIT Lincoln Laboratory RT-04F diarization systems: Applications to broadcast audio and telephone conversations. In *Proceedings of Fall 2004 Rich Transcription Workshop*, 2004.

[87] D. A. Reynolds and P. Torres-Carrasquillo. Approaches and applications of audio diarization. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 953–956, 2005.

[88] Douglas A Reynolds. Speaker identification and verification using gaussian mixture speaker models. *Speech communication*, 17(1-2):91–108, 1995.

[89] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.

[90] J. R. Rohlicek, D. Ayuso, M. Bates, R. Bobrow, A. Boulanger, H. Gish, P. Jeanrenaud, M. Meteer, and M. Siu. Gisting conversational speech. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 113–116, 1992.

[91] A. E. Rosenberg, A. Gorin, Z. Liu, and P. Parthasarathy. Unsupervised speaker segmentation of telephone conversations. In *Proceedings of the International Conference on Spoken Language Processing*, pages 565–568, 2002.

[92] Jamal E Rougui, Mohammed Rziza, Driss Aboutajdine, Marc Gelgon, and José Martinez. Fast incremental clustering of gaussian mixture speaker models for scaling up retrieval in on-line broadcast. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages V–V. IEEE, 2006.

[93] Mickael Rouvier, Grégor Dupuy, Paul Gay, Elie Khoury, Teva Merlin, and Sylvain Meignier. An open-source state-of-the-art toolbox for broadcast news diarization. In *Interspeech*, 2013.

[94] Bernd Scherer. A note on the returns from minimum variance investing. *Journal of Empirical Finance*, 18(4):652–660, 2011.

[95] Gregory Sell and Daniel Garcia-Romero. Speaker diarization with plda i-vector scoring and unsupervised calibration. In *Proceedings of IEEE Spoken Language Technology Workshop*, pages 413–417. IEEE, 2014.

[96] Gregory Sell, David Snyder, Alan McCree, Daniel Garcia-Romero, Jesús Villalba, Matthew Maciejewski, Vimal Manohar, Najim Dehak, Daniel Povey, Shinji Watanabe, et al. Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge. In *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 2808–2812, Sep. 2018.

[97] Mohammed Senoussaoui, Patrick Kenny, Themos Stafylakis, and Pierre Dumouchel. A study of the cosine distance-based mean shift for telephone speech diarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1):217–227, 2013.

[98] Laurent El Shafey, Hagen Soltau, and Izhak Shafran. Joint Speech Recognition and Speaker Diarization via Sequence Transduction. In *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 396–400. ISCA, 2019.

[99] S. H. Shum, N. Dehak, R. Dehak, and J. R. Glass. Unsupervised methods for speaker diarization: An integrated and iterative approach. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(10), Oct. 2013.

[100] Stephen Shum, Najim Dehak, and James Glass. On the use of spectral and iterative methods for speaker diarization. In *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 482–485, Sep. 2012.

[101] Stephen H Shum, Najim Dehak, Réda Dehak, and James R Glass. Unsupervised methods for speaker diarization: An integrated and iterative approach. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(10):2015–2028, May 2013.

[102] Matthew A Siegler, Uday Jain, Bhiksha Raj, and Richard M Stern. Automatic segmentation, classification and clustering of broadcast news audio. In *Proceedings of DARPA Speech Recognition Workshop*, pages 97–99, 1997.

[103] Jan Silovsky, Jindrich Zdansky, Jan Nouza, Petr Cerva, and Jan Prazak. Incorporation of the asr output in speaker segmentation and clustering within the task of speaker diarization of broadcast streams. In *International Workshop on Multimedia Signal Processing*, pages 118–123. IEEE, 2012.

[104] David Snyder. Callhome diarization recipe using x-vectors. Github, May 4, 2018. [Online]. Available: `https://david-ryan-snyder.github.io/2018/05/04/model_callhome_diarization_v2.html`, [Accessed Oct. 9, 2019].

[105] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur. Deep neural network embeddings for text-independent speaker verification. In *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 999–1003, 2017.

[106] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust DNN embeddings for speaker recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5329–5333, Apr. 2018.

[107] G. W. Stewart and J. Sun. *Matrix Perturbation Theory*. Boston, MA, USA: Academic Press, 1990.

[108] Andreas Stolcke and Takuya Yoshioka. DOVER: A method for combining diarization outputs. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 757–763. IEEE, 2019.

[109] M. Sugiyama, J. Murakami, and H. Watanabe. Speech segmentation and clustering based on speaker features. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 395–398, 1993.

[110] Guangzhi Sun, Chao Zhang, and Philip C Woodland. Speaker diarisation using 2d self-attentive combination of embeddings. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5801–5805. IEEE, 2019.

[111] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014.

[112] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[113] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[114] Yun Tang, Guohong Ding, Jing Huang, Xiaodong He, and Bowen Zhou. Deep speaker embedding learning with multi-level pooling for text-independent speaker verification. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6116–6120. IEEE, 2019.

[115] S. E. Tranter and D. A. Reynolds. Speaker diarisation for broadcast news. In *Proceedings of Odyssey Speaker and Language Recognition Workshop*, pages 337–344, 2004.

[116] S. E. Tranter, K. Yu, G. Evermann, and P. C. Woodland. Generating and evaluating for automatic speech recognition of conversational telephone speech. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 753–756, 2004.

[117] S. E. Tranter, K Yu, D. A. Reynolds, G Evermann, D. Y. Kim, and P. C. Woodland. An investigation into the the interactions between speaker diarisation systems and automatic speech transcription. *CUED/F-INFENG/TR-464*, 2003.

[118] Wei-Ho Tsai, Shih-Sian Cheng, and Hsin-Min Wang. Speaker clustering of speech utterances using a voice characteristic reference space. In *Proceedings of the International Conference on Spoken Language Processing*, 2004.

[119] F. Valente, P. Motlicek, and D. Vijayasenan. Variational Bayesian speaker diarization of meeting recordings. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4954–4957, 2010.

[120] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. G-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4052–4056, 2014.

[121] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4052–4056. IEEE, 2014.

[122] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[123] 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(7):1382–1393, 2009.

[124] Jesús Villalba, Nanxin Chen, David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Jonas Borgstrom, Fred Richardson, Suwon Shon, François Grondin, et al. State-of-the-art speaker recognition for telephone and video speech: The JHU-MIT submission for NIST SRE18. In *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 1488–1492, 2019.

[125] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statist. and Comput.*, 17(4):395–416, 2007.

[126] Quan Wang, Carlton Downey, Li Wan, Philip Andrew Mansfield, and Ignacio Lopez Moreno. Speaker diarization with lstm. *arXiv preprint arXiv:1710.10468*, 2017.

[127] Quan Wang, Carlton Downey, Li Wan, Philip Andrew Mansfield, and Ignacio Lopz Moreno. Speaker diarization with LSTM. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5239–5243, Apr. 2018.

[128] C. Wooters, J. Fung, B. Peskin, and X. Anguera. Toward robust speaker segmentation: The ICSI-SRI Fall 2004 diarization system. In *Proceedings of Fall 2004 Rich Transcription Workshop*, pages 402–414, 2004.

[129] Sree Harsha Yella and Andreas Stolcke. A comparison of neural network feature transforms for speaker diarization. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[130] Bing Yin, Jun Du, Lei Sun, Xueyang Zhang, Shan He, Zhenhua Ling, Guoping Hu, and Wu Guo. An analysis of speaker diarization fusion methods for the first dihard challenge. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1473–1477. IEEE, 2018.

[131] Zbyněk Zajíc, Marek Hrúz, and Luděk Müller. Speaker diarization using convolutional neural network for statistics accumulation refinement. In *Proc. Interspeech 2017*, pages 3562–3566, 2017.

[132] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Proceedings of Advances in Neural Information Processing Systems*, pages 1601–1608, Dec. 2005.

[133] Aonan Zhang, Quan Wang, Zhenyao Zhu, John Paisley, and Chong Wang. Fully supervised speaker diarization. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6301–6305, 2019.