# SphereDiar - an efficient speaker diarization system for meeting data

Tuomas Kaseva

**School of Electrical Engineering**

Thesis submitted for examination for the degree of Master of Science in Technology.
Espoo 27.5.2019

**Supervisor**

Prof. Mikko Kurimo

**Advisor**

M.Sc. Aku Rouhe

Aalto University
School of Electrical
Engineering

Aalto University
School of Electrical
Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

**Author** Tuomas Kaseva

**Title** SphereDiar - an efficient speaker diarization system for meeting data

**Degree programme** Computer, Communication and Information Sciences

**Major** Signal, Speech and Language Processing **Code of major** ELEC3031

**Supervisor** Prof. Mikko Kurimo

**Advisor** M.Sc. Aku Rouhe

**Date** 27.5.2019 **Number of pages** 65 **Language** English

**Abstract**

The objective of speaker diarization is to determine who spoke and when in a given audio stream. This information is useful in multiple different speech related tasks such as speech recognition, automatic creation of rich transcriptions and text-to-speech synthesis. Moreover, speaker diarization can also play a central role in the creation and organization of speech-related datasets.

Speaker diarization is made difficult by the immense variability in speakers and recording conditions, and the unpredictable and overlapping speaker turns of spontaneous discussion. Especially diarization of meeting data has been very challenging. Even the most advanced speaker diarization systems still struggle with this type of data.

In this thesis, a novel speaker diarization system, named SphereDiar and designed for the diarization of meeting data, is proposed. This system combines three novel subsystems: the SphereSpeaker neural network for speaker modeling, a segmentation method named Homogeneity Based Segmentation and a clustering algorithm Top Two Silhouettes. The system harnesses up-to-date deep learning approaches for speaker diarization and addresses the problem of overlapping speech in this task.

Experiments are performed on a dataset consisting of over 200 meetings. The experiments have two main outcomes. Firstly, the use of Homogeneity Based Segmentation is not vital for the system. Thus, the configuration of SphereDiar can be simplified by omitting segmentation. Furthermore, SphereDiar is shown to surpass the performance of two different state-of-the-art speaker diarization systems.

**Keywords** speaker diarization, speaker modeling, segmentation, clustering, meeting data

Aalto University
School of Electrical
Engineering

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi
Diplomityön tiivistelmä

**Tekijä** Tuomas Kaseva

**Työn nimi** SphereDiar - tehokas puheen diarisointijärjestelmä kokousäänitteitä varten

**Koulutusohjelma** Computer, Communication and Information Sciences

**Pääaine** Signal, Speech and Language Processing **Pääaineen koodi** ELEC3031

**Työn valvoja** Prof. Mikko Kurimo

**Työn ohjaaja** DI Aku Rouhe

**Päivämäärä** 27.5.2019 **Sivumäärä** 65 **Kieli** Englanti

**Tiivistelmä**

Puheen diarisaatiolla tarkoitetaan automaattista prosessia, joka pyrkii selvittämään kuka puhui ja milloin. Tätä prosessia voidaan hyödyntää monissa puheen käsittelyyn liittyvissä sovelluksissa kuten puheentunnistuksessa, puheen syntetisoinnissa sekä esimerkiksi pöytäkirjojen teossa. Näiden sovellusten lisäksi puheen diarisointia voidaan käyttää myös puheeseen liittyvien datakokoelmien automaattiseen rakennukseen ja organisointiin.

Puheen diarisointi on kuitenkin usein hankalaa, sillä kaikki puhujat ovat erilaisia, ja äänitysten taso ja olosuhteet voivat poiketa huomattavasti toisistaan. Näiden lisäksi spontaanissa puheessa puheenvuorot voivat vaihtua äkillisesti sekä sisältää päälle puhumista. Näin käy usein varsinkin kokousäänitteissä, jotka ovat vielä tänäkin päivänä erityisen hankalia puheen diarisoinnin kannalta.

Tämä diplomityö esittelee uudenlaisen puheen diarisointijärjestelmän, joka on erikoistunut kokousäänitteisiin. Tämä järjestelmä, nimeltään SphereDiar, rakentuu kolmesta osasta: SphereSpeaker-neuroverkosta puhujan mallinnukseen, Homogeneity Based Segmentation-metodista puheen segmentointiin sekä Top Two Silhouettes-algoritmista klusterointiin. SphereDiar hyödyntää uusimpia syväoppimismetodeita, ja on kehitetty huomioimaan varsinkin päälle puhumisen vaikutus puheen diarisaatiossa.

Järjestelmän suorituskykyä on arvioitu kokeissa, joissa käytettiin yli 200 kokousäänitettä. Näissä kokeissa saavutettiin kaksi keskeistä tulosta. Näistä ensimmäinen oli se, että Homogeneity Based Segmentation metodin käyttö ei ollut välttämätöntä järjestelmälle. Siten SphereDiar voitiin yksinkertaistaa jättämällä segmentointi kokonaan pois. SphereDiaria verrattiin myös kahteen alan parhaimpiin kuuluvaan puheen diarisointijärjestelmään ja sen osoitettiin saavan parempia tuloksia näissä vertailuissa.

**Avainsanat** puheen diarisointi, puhujan mallintaminen, segmentointi, klusterointi, kokousäänite

# Preface

Thank you, Mikko Kurimo, Aku Rouhe, Ville Pulkki, Tom Bäckström, Jarno Latva, Reima Karhila, Anja Virkkunen, Stig-Arne Grönroos, parents and friends!

Otaniemi, 27.5.2019

Tuomas Kaseva

# Contents

# Symbols and abbreviations

## Symbols

| | |
|---|---|
| $\mathbf{A}$ | attention matrix |
| $\mathbf{c}$ | cluster center |
| $C$ | set of cluster centers |
| $d$ | average dissimilarity |
| $\delta$ | threshold used in Top2S |
| $\mathbf{f}$ | speaker embedding |
| $F$ | set of speaker embeddings $\mathbf{f}$ |
| $G$ | Gaussian Mixture Model |
| $h$ | binary label created with HBS |
| $H$ | set of binary labels $h$ |
| $H_{\%}$ | homogeneity percentage |
| $H_{\theta\%}$ | homogeneity percentage threshold |
| $\hat{h}$ | output of the HBS neural network |
| $I_o$ | set of time instances which include overlapping speech |
| $I_s$ | set of time instances which include single speaker |
| $l$ | speaker label |
| $L$ | set of speaker labels $l$ |
| $\mathbf{M}$ | supervector |
| $N_{max}$ | maximum number of speakers |
| $O$ | sum of cosine distances, spherical K-means objective |
| $P$ | clustering proposal |
| $R$ | number of cluster initializations |
| $s$ | silhouette score |
| $\mathbf{s}$ | audio frame with 2 second duration |
| $S$ | set of audio frames $\mathbf{s}$ |
| $\sigma$ | binary target for HBS |
| $\Sigma$ | set of targets $\sigma$ |
| $T$ | sequence of transcription labels |
| $\mathbf{T}$ | total variability matrix |
| $\theta$ | threshold |
| $\mathbf{v}$ | i-vector |
| $\mathbf{x}$ | feature sequence |
| $\mathbf{x}^{(i)}$ | $i$th set of features in $\mathbf{x}$ |
| $X$ | set of feature sequences $\mathbf{x}$ |

## Operators

| | |
|---|---|
| $\|.\|$ | number of elements in a given set |
| $\|\|.\|\|_2$ | $L^2$-norm |

## Abbreviations

| | |
|---|---|
| AMI | augmented multi-party interaction |
| $AMI_{eval}$ | evaluation set collected from the AMI corpus |
| ASR | automatic speech recognition |
| $AR$ | adjusted Rand score |
| BIC | Bayesian information criterion |
| $Clust_{dev}$ | development set collected from the ICSI corpus and the AMI corpus |
| DER | diarization error rate |
| EM | expectation maximization |
| GLR | generalized likelihood ratio |
| GMM | gaussian mixture model |
| HAC | hierarchical agglomerative clustering |
| HBS | Homogeneity Based Segmentation |
| HMM | hidden Markov model |
| ICSI | international computer science institute |
| $ICSI_{eval}$ | evaluation set collected from the ICSI corpus |
| ILP | integer linear programming |
| JFA | joint factor analysis |
| LDA | linear discriminant analysis |
| LS | Librispeech corpus |
| $LS_{N_s}$ | Librispeech partition with $N_s$ speakers |
| LSTM | long short-term memory |
| MAP | mean average precision |
| MFCC | mel-frequency cepstral coefficients |
| PLDA | probabilistic linear discriminant analysis |
| SS | SphereSpeaker |
| SVM | support vector machine |
| TopS | Top Silhouette |
| Top2S | Top Two Silhouettes |
| TTS | text-to-speech |
| UBM | universal background model |
| VC | Voxceleb2 datatset |
| $VC_{N_s}$ | Voxceleb2 partition with $N_s$ speakers |
| WCCN | within-class covariance normalization |

# 1 INTRODUCTION

## 1.1 Speaker diarization

Speaker diarization refers to an automatic process which aims to answer the question "who spoke and when" [1]. In this process, the first objective is to determine which parts of a given audio stream contain speech. Next, these parts are segmented into speaker turns which depict intervals including only one or one clearly distinguishable speaker. Finally, each speaker turn is assigned with a suitable speaker identity, and the speaker diarization task is completed. In this task, no visual cues are exploited, making speaker diarization a very challenging speech processing application [1].

When using the phrase "who spoke and when", some care and emphasis must be put on the words "who spoke". Recognizing the exact identities of the speakers and being able to perfectly separate between the speakers may be different things. For example, assuming an audio stream with three different speakers having three equal length speaker turns, an output of a speaker diarization system could be

$$[\textit{Trump, Sanders, Clinton}] \text{ or } [A,\ B,\ C].$$

In the latter, the speakers are separable but not known. Nevertheless, in both cases the diarization system is capable of determining which speaker was speaking at any given time. In the former, the task can easily get prohibitive, as the diarization system has to have prior knowledge of all of the speakers in a given audio stream, but the number of people in the world is in the billions and counting. Furthermore, humans have the ability to distinguish between speakers even when they are not familiar. Thus, a machine should be able to do the same. For these reasons, the latter, also known as an unsupervised speaker diarization task, is more often applied in the speaker diarization literature [1] and the same approach is adopted in this thesis. From here on, speaker diarization refers to unsupervised speaker diarization in this thesis.

## 1.2 Motivation

A task of determining which speaker spoke and when is not something humans can not do. It is well known that humans are exceptionally good at noticing when the speaker changes and whether an audio contains speech or not [2]. However, when the sizes of audio streams in this task grow to, say, hundreds or thousands of hours, the amount of time and manpower needed becomes a limiting factor in manual speaker

diarization. In addition, even if the duration of these streams would be short enough for manual processing, this processing could easily become monotonic and tedious. Most importantly, speaker diarization is a necessary and beneficial subtask in many different speech processing applications.

The first of these applications is the creation of rich transcriptions [1, 3]. In addition to what was spoken, these type of transcriptions include also which person spoke and when [1]. This feature is helpful whenever there is need for summarization of the content of an audio stream. Such streams may include meetings, TV-shows, debates and lectures [1]. Furthermore, rich transcriptions allow extracting parts of the stream which include only one specific speaker. These parts can then be used, for example, to create speech recognition [4] or speaker recognition datasets [5] automatically.

Speaker diarization can also be exploited in speaker adaptation in automatic speech recognition (ASR) [6] and text-to-speech (TTS) synthesis [7]. In these tasks, speaker diarization is used to extract speaker characteristics which can be utilized for training and evaluation of the systems in TTS and ASR. These characteristics are generally compressed to a set of features which are given as an input for these systems [6, 7].

In addition to the speech processing applications, speaker diarization can also contribute to non-speech related tasks. In the section 3, speaker diarization systems are shown to combine different segmentation, speaker verification and clustering techniques. Speaker verification, the task of verifying whether two utterances are either produces by different or by the same speaker, [8], is the speech equivalent of face verification [5, 9, 10]. In recent years, the use of deep learning approaches has moved these two fields even more close to each other [5, 10]. Consequently, advances in either field can benefit the other. Such is also the case with clustering, the unsupervised task of associating similar objects, which has applications for example in document organization [11].

Nevertheless, speaker diarization has proven to be a difficult and complex task that in many occasions fails to provide satisfactory results [1, 12]. This is evident especially with meeting data which can include spontaneous and overlapping speech and challenging recording conditions [1, 13, 14, 15]. Consequently, this thesis develops a novel speaker diarization system which is designed especially for meeting data.

## 1.3 Contributions

This thesis proposes a novel speaker diarization system, SphereDiar, which is composed of three main components:

1. A neural network, named SphereSpeaker (SS), which transforms a short frame of speech into a representation which characterizes the corresponding speaker.

2. A method for determining which partitions of an audio stream contain multiple speakers. This method is named Homogeneity Based Segmentation (HBS). It uses a neural network for categorizing the partitions.

3. An algorithm, named Top Two Silhouettes (Top2S), which is designed for clustering speaker representations.

Each of these components is developed in this thesis. The SS neural network introduces a novel architecture which creates the speaker representations as a byproduct of a speaker classification task. The main feature in this architecture is a $L^2$ normalization layer which forces the speaker representations to lie on a hypersphere. This simple operation is shown to be very effective in this thesis.

HBS introduces a method which performs two important operations in speaker diarization, speaker change and overlapping speech detection, simultaneously. This is made possible by using a novel neural network which is designed for detecting if either one or multiple speakers are vocal. Similar approach has not been published in the speaker diarization literature.

Top2S can be divided to two main operations. In the first, the algorithm creates multiple clustering proposals and in the second, it determines which proposal is the best. The Top2S algorithm is especially designed to be used with the speaker representations extracted from the SS neural network. Moreover, it exploits several heuristic rules which have been discovered in this thesis.

SphereDiar is evaluated with a large meeting dataset which includes over 200 meeting recordings and is trained with various configurations and datasets. In evaluation, the system surpasses previous state-of-the-art scores on two different meeting corpora. Moreover, it is shown that the system can be simplified by omitting the use of HBS. This is not only convenient, but also an interesting discovery since especially overlapping speech detection, which is a subtask in HBS, has been a prominent research direction in speaker diarization [1, 12, 16, 17]. The system is also made available online [1].

[1] `https://github.com/Livefull/SphereDiar`

## 1.4 Outline

The organization of the thesis can be summarized as follows. First, the data is described. Next, the proposed speaker diarization system is introduced and elaborated. In addition, related works concerning different components of the system are discussed and their influence explained. Then, the different experiments conducted on the system are presented and analysed. Finally, conclusions of the thesis are discussed.

# 2 DATA

The data used in this thesis consists of two parts: a meeting corpus and a collection of speaker corpora. This section explains what they are, what are they used for and why they were chosen. In addition, the section introduces two important data related concepts: the input format of the system and a novel concept which is called homogeneity percentage.

## 2.1 Meeting corpus

The meeting corpus composes of the AMI (Augmented Multi-party Interaction) and ICSI (International Computer Science Institute) corpora which in turn consist of audio recordings of different scenario and non-scenario meetings from various sites [18, 19]. In the scenario meetings, participants have predetermined roles and are given specific topics to discuss, whereas the non-scenario meetings are normal meetings which would have taken place regardless of third-party recordings. The scenario meetings occur only in the AMI corpus. All speech is in English but both corpora include non-native English speakers with different English accents [18, 19]. The ICSI corpus includes 75 meetings with 3 to 9 participants per meeting and the AMI corpus has 171 meetings with number of participants varying from 3 to 5.

The main purpose of the meeting corpus is to form an evaluation set for speaker diarization. In addition, the corpus is also used in training the HBS system. The most important reasons for choosing the AMI and ICSI corpora are their availability and the challenge they pose. Both corpora can be downloaded for free and have clear usage instructions. The challenge is based on the number of participants involved, the recording conditions and having both spontaneous and overlapping speech, which occur frequently in both the scenario and non-scenario meetings. Moreover, both corpora have been rather popular in speaker diarization literature [13, 15, 16, 20] making it possible to compare results.

### 2.1.1 Audio format

The meetings in AMI and ICSI are recorded with tabletop microphone arrays, and lapel and headset microphones [18, 19]. As a consequence, both corpora provide audio files in different formats. All formats have a 16 kHz sampling frequency. In this thesis, the chosen audio format is $Headset\ Mix$ in which headset microphone recordings are summed to form a synthetic near-field audio stream [13].

The reasons behind this choice can be summarized as follows. Firstly, the format has the highest audio quality among the formats provided in the AMI and ICSI corpora. In addition, the ICSI corpus does not provide recordings from individual tabletop microphones, which would be the natural alternative to *Headset Mix*. Finally, since the same format has been used previously in related literature [13, 15, 20], the comparison of obtained results is enabled.

### 2.1.2 Transcription labels

The AMI and ICSI corpora provide both manually generated and ASR-based word-level transcripts which describe what the different participants have said and when they have spoken [18, 19]. In practice, both transcripts contain a starting and an ending time of each uttered word of a given speaker in the meeting with an accuracy of 0.1 seconds. In addition, the transcripts include also time boundaries for occurrences of many non-vocal sounds such as laughter and coughing.

In this thesis, these labels, named transcription labels, are generated by combining e manually generated and ASR-based transcripts. This choice is based on preliminary investigations which showed that both transcripts suffered from minor deficiencies. With the manually generated ones, the problem was that short and natural silent segments which might occur between words were labeled as speech. On the other hand, the ASR based transcriptions excluded the silent segments but in some cases assigned undesirable speech sounds, most notably breathing sounds, as vocal activity inadvertently. Fortunately, these deficiencies could be countered to some extent by creating speaker diarization labels based on both transcripts.

The combining of the transcripts is performed with the following procedure. First, for a given meeting audio, two different preliminary transcription label sets are generated using both transcripts. These sets include labels corresponding to either single speaker, overlapping speech, or non-speech. The labels are assigned to every time instance with 0.1 second interval. During this process, filler words such as ”uh”, ”uh-huh” and ”huh” are treated as silence as are all other non-vocal sounds except laughter. The laughing sounds were included as they occur frequently in a natural overlapping speech [16].

Then, time instances which only have a single speaker in both sets are collected to form an instance set $I_s$. A similar operation is also performed with labels describing overlapping speech, giving $I_o$. Finally, the transcription labels are obtained by assigning all time instances in $I_s$ to unique speakers depicted in the manually generated preliminary transcript set, time instances in $I_o$ to overlapping speech and

all other instances to non-speech. Unfortunately, complete ASR based transcriptions were not available for all meetings, which led to exclusions which are depicted in Table 1. After exclusions, the number of included AMI meetings is 163 and the number of ICSI meetings is 74, totalling 237 meetings in the meeting corpus.

**Table 1:** Removed meetings.

| | |
|---|---|
| AMI | EN2001a, EN2001e, EN2002c, EN2003a, EN2006a, EN2006b, IB4005, IS1003b |
| ICSI | Bmr012 |

### 2.1.3 Gender distributions

As additional information, statistics of the number of speakers and gender distributions are provided in the transcriptions. The statistics are presented in Table 2. Interestingly, different numbers of speakers have been reported for the AMI corpus in literature. In [20], the number is 150, but in [13] it is close to 200. In this thesis, the number of speakers in the AMI corpus was determined using unique identifiers in the *meetings.xml* file provided with the transcripts. The number of speakers found this way is 118. This number did not increase even when including all 171 meetings. In the ICSI corpus, the number of speakers is 53, which concurred with [19]. As one of the participants was only recorded by the far-field microphone and only had few vocal segments, they were excluded.

**Table 2:** Gender distributions expressed using speaker counts in the meeting corpus.

| | Female | Male | In total |
|---|---|---|---|
| AMI | 43 | 75 | 118 |
| ICSI | 13 | 39 | 52 |
| Meeting corpus | 56 | 114 | 170 |

### 2.1.4 Framing

As was discussed, the primary purpose of the meeting corpus is to evaluate the proposed speaker diarization designed in this thesis. Ideally, the evaluation could be performed on each meeting by predicting a label value every 0.1 seconds and then comparing the predictions with the transcription labels. Here, the step size of 0.1 seconds is the theoretical maximum accuracy of the transcription labels as mentioned in subsection 2.1.2. This, however, would mean that the system would need to

recognize speakers based on audio chunks with a duration of a less than a tenth of a second: a task that would be extremely difficult even for humans. Therefore, a compromise is made in this thesis by assigning predictions to considerably longer but overlapping audio frames. Consequently, every meeting audio in meeting corpus is transformed into the following form

$$S = \{\mathbf{s_1}, ..., \mathbf{s_N}\}, \quad \mathbf{s_i} \in \mathbb{R}^{32000}, \tag{1}$$

where $S$ depicts a sequence of $N$ frames $\mathbf{s_i}$ with a 2 second duration which are extracted from the given audio with a 1.5 second overlap. The 32000 dimensions results from the sampling frequency of 16 kHz. This configuration, disregarding the overlap duration which can vary, is also the required input format of the proposed speaker diarization system.

The choices of frame and overlap duration are based on several factors. Firstly, it is necessary that a frame is long enough so that proper modeling of the speaker corresponding to the frame is possible. Secondly, the frame has to be a short enough so that spontaneous speaker changes would not go unnoticed. As a result, a duration of 2 seconds was chosen, which has also been used in [21, 22].

Relatively large overlapping in turn is beneficial for the clustering procedure as it enables more samples for forming the clusters. However, an increase in overlap duration also results in a increase in computing time as the number of frames in $S$ increases. In this thesis, preliminary experiments illustrated that an overlap duration of 1.5 seconds would then be a suitable compromise.

Before the creation of $S$, however, all samples labeled as non-speech are excluded from each meeting audio. In other words, a perfect voice activity detection (VAD) is performed. This is necessary as the speaker diarization system proposed in this thesis does not provide a VAD system. This exclusion will be discussed and justified in section 3.

### 2.1.5 Speaker labels

After the creation of $S$ for an arbitrary meeting audio with $N_s$ participants, the next step is to define the corresponding speakers labels

$$L = \{l_1, ..., l_N\}, l \in \{-1, 1, .., N_s\}, \tag{2}$$

for each $\mathbf{s}$ in $S$. In this thesis, these speaker labels also include negative labels which depict overlapping speech. Nevertheless, the negative valued labels will not be

included in the evaluation of the speaker diarization system. The reasons for this choice will be explained in section 4.

However, the label assignment is not straightforward. As a result of the framing operation, a speaker content of a given $\mathbf{s}$ from $S$ can be described with a set

$$T = \{T_{-1}, T_1, ..., T_{n_s}\}, \tag{3}$$

where $T_{i\neq -1}$ is a set of transcription labels corresponding to speaker $i$, $n_s$ the number of speakers in $\mathbf{s}$ and $T_{-1}$ the set of transcription labels corresponding to overlapping speech. In this thesis, the speaker label $l$ of $\mathbf{s}$ is then defined as

$$l = \arg\max_i |T_i|, \tag{4}$$

where $|T_i|$ is the number of labels in $T_i$. However, it is clear from this definition that $l$ itself can not fully describe the speaker content of $\mathbf{s}$. For example, for some of the $\mathbf{s}$, the corresponding $T$ could include only $T_1$ and $T_2$ with $|T_1| = |T_2|$. Furthermore, even though transcription labels are very precise in general, they do contain inaccuracies in speaker change boundaries. For these reasons, an additional measure of the speaker content in $\mathbf{s}$ is needed.

### 2.1.6 Homogeneity percentage

Homogeneity percentage, abbreviated $H_{\%}$, for a given $\mathbf{s}$ is defined as

$$H_{\%} = \frac{max_i|T_{i\neq o}|}{|T|} * 100\%. \tag{5}$$

As the name suggests, this percentage depicts the homogeneity of the speaker content of $\mathbf{s}$. For instance, if $T = \{T_1\}$ implying that $\mathbf{s}$ is uttered by a single speaker, the percentage is 100%. On the other hand, if $T = \{T_{-1}\}$ suggesting that the frame consists solely of overlapping speech, then the percentage is 0%. In general, the percentage is somewhere between these two values, for example as visualized in Figure 1. The percentage can be interpreted as a confidence metric for the label $l$.

Consequently, the percentage can also be used to divide frames into two categories: ones with speaker change boundaries and overlapping speech and to ones which can be considered to only include one dominant speaker. A suitable homogeneity percentage threshold between these two categories was found to be 65% in preliminary experiments. The categorization of the meeting corpus with this threshold is presented in Table 3. Naturally, a gray area where categories are mixed does exist with frames

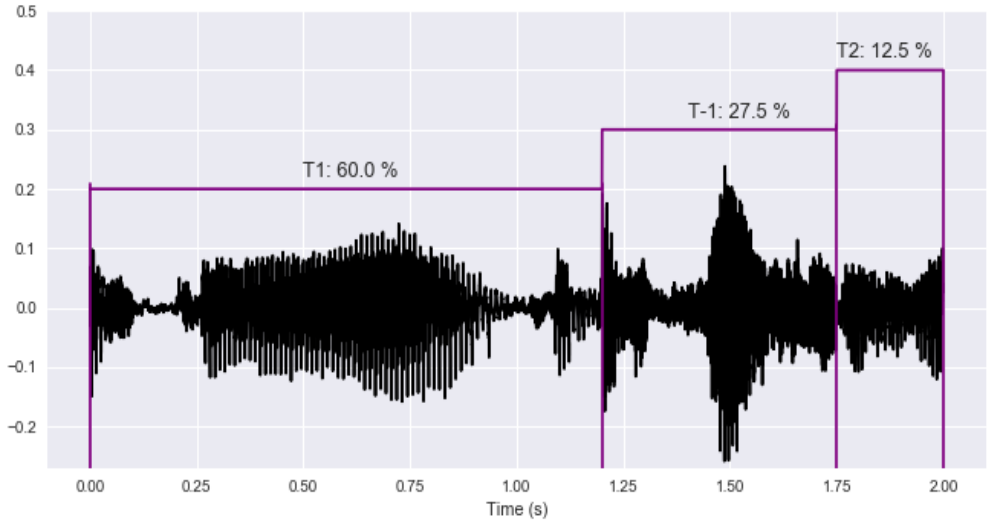


**Figure 1:** A frame with $H_{\%} = 60\%$.

having $H_{\%}$ close to the threshold.

**Table 3:** Frame categorization in the meeting corpus.

| | Percentage (%) | Number of frames |
|---|---|---|
| $H_{>=65\%}$ | 84.2 | 595 086 |
| $H_{<65\%}$ | 15.8 | 111 592 |
| Total | 100 | 706 678 |

## 2.2 Speaker corpora

The speaker corpora include four different speaker datasets which are collected from Librispeech corpus (LS) and Voxceleb2 dataset (VC) [9, 4]. Librispeech consists of recordings of read speech from 2484 different speakers [4]. The recordings are gathered from the LibriVox project audiobooks read by volunteers [4]. All recordings are in English, comprise of 1000 hours of speech and have a 16 kHz sampling rate. The corpus is initially designed for ASR purposes and as a result, the recordings contain virtually no background noise, and the utterances have clear pronunciation [4].

Voxceleb2 includes over a million utterances from over 6000 speakers [9]. These utterances are collected with a semi-automatic procedure from celebrities in Youtube videos with a high variety of recording conditions and background noise [9]. The

dataset comes in both audio and video formats with the former at a 16kHz sampling rate. In this thesis, only the audio format is used. In addition, the dataset includes multiple languages, English being the most common one [9]. The dataset was created for challenging speaker recognition purposes making it extremely relevant for speaker diarization purposes [9, 10].

The speaker corpora are composed of two Librispeech partitions, abbreviated $LS_{1000}$ and $LS_{2000}$, and two Voxceleb2 partitions $VC_{1000}, VC_{2000}$. The index in each partition describes the number of speakers included in the partition. These partitions will be used in the training and evaluation of the SS neural network. The motivation for choosing VC and LS is primarily their contrasting purposes. As discussed, LS was gathered for ASR, whereas VC is specifically developed for speaker recognition. The big question is, does using VC then give noticeable improvements to the speaker diarization system when comparing against LS. Moreover, their size enables creating the partitions with the same number of speakers and the same amount of speech material, making the evaluation relatively fair. Finally, LS and VC are both disjoint in speakers with the meeting corpus meaning that the development of the SS neural network is independent of the meeting corpus.

### 2.2.1 Partition generation

Just as with the meeting corpus, the partitions of the speaker corpora are assembled from frames. The assembling can be divided into four steps as follows. In the first step, 500 males and 500 females with the most speech material are gathered from both Librispeech and Voxceleb2. Then, an additional 1000 speakers are collected from both datasets again based on the amount of speech material, but without any gender quotas. After this step, four different speaker identity sets with 1000 and 2000 speakers are collected, with relatively balanced gender distributions as can be seen in Table 4.

**Table 4:** Gender distribution in speaker corpora.

| | Number of females | Number of males | Number of speakers |
|---|---|---|---|
| $LS_{1000}$ | 500 | 500 | 1000 |
| $LS_{2000}$ | 987 | 1013 | 2000 |
| $VC_{1000}$ | 500 | 500 | 1000 |
| $VC_{2000}$ | 731 | 1269 | 2000 |

In the third step, all corresponding speech data for each speaker is processed with WebRTC VAD [23] with the aim of removing silences and non-vocal sounds. Finally,

the processed speech data for each speaker identity set is framed with the same procedure as with the meeting corpus, with 2 second frame duration but without overlap. The partitions $LS_{1000}, LS_{2000}, VC_{1000}$ and $VS_{2000}$ are then formed from these frames. In order to balance the speaker label distributions with the partitions with the same number of speakers, the maximum number of frames corresponding to a given speaker is limited. The limit for the partitions $LS_{2000}$ and $VC_{2000}$ is assigned as 670 whereas the limit for $LS_{1000}$ and $VC_{1000}$ is 1000. The $LS_{1000}$ partition, however, did not include quite as much speech material as $VC_{1000}$, so the maximum number of frames per speaker is only 764. The final frame compositions of the partitions are summarized in Table 5.

**Table 5:** Frame compositions in speaker corpora.

| | Minimum number of frames per speaker | Maximum number of frames per speaker | Number of frames |
|---|---|---|---|
| $LS_{1000}$ | 382 | 764 | 654 297 |
| $LS_{2000}$ | 341 | 670 | 1 204 967 |
| $VC_{1000}$ | 838 | 1000 | 995 443 |
| $VC_{2000}$ | 577 | 670 | 1 337 601 |

Unlike with the meeting corpus, the frames in the speaker corpora are not assigned homogeneity percentages. Or implicitly, all of the frames are assigned $H_{\%} = 100\%$. With the Librispeech corpus, this procedure is well justified as the audio does not include speaker changes or overlapping speech. With the Voxceleb2 dataset however, this assumption is not completely accurate. The varity of the recordings means that there might be multiple or alternating speakers which are not transcribed correctly. Nevertheless, this is probably rare and if it occurs, a dominant speaker could be still determined with a relative ease [9]. Consequently, each frame in the partitions is assigned an unique speaker label $l$ which can not depict overlapping speech.

# 3 SPHEREDIAR

The main objective of SphereDiar is to transform a sequence of frames $S$ into a sequence of speaker labels $L$. This transformation is obtained with four key procedures: feature extraction, speaker modeling, segmentation and clustering which are visualized in Figure 2. The last three are implemented by the SphereSpeaker neural network, Homogeneity Based Segmentation and the Top Two Silhouettes algorithm, respectively. What this system does not include, however, is voice activity detection (VAD). This is by no means a trivial exclusion since VAD is an essential component in any speaker diarization system [1]. However, when diarization systems are developed, reference VAD labels are often used in order to focus on the actual speaker distinguishing task [13, 16, 17, 20]. Such an approach is also used in this thesis and therefore no methods for VAD will be discussed or proposed.

This section elaborates the functioning of SphereDiar and each component illustrated in Figure 2. Furthermore, the relevant literature is reviewed for each component.

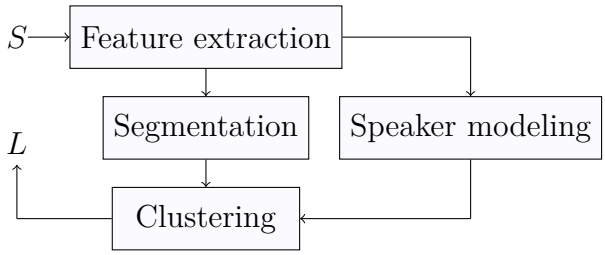


**Figure 2:** Block diagram of SphereDiar.

## 3.1 Feature extraction

A frame **s** consists of a sequence of values which depict air pressure values discretely in time. As an input format, **s** is unsuitable for speaker diarization systems since it is not descriptive in terms of human auditory perception. It has been shown that human hearing is mostly based on different frequency components in sound, their relations and dynamic structure [24]. Moreover, some frequency bands are more relevant than others [25]. However, these frequency components and their importance are difficult to detect automatically simply based on the amplitude sequence. For this reason, it is often necessary to extract more auditorily descriptive features from the frames and use them instead as the input for the systems.

Yet, feature extraction is not indispensable. Recently in speaker diarization

related tasks, such as speaker modeling and segmentation, numerous approaches have been proposed in which feature extraction is left out [5, 9, 10, 26, 27]. In these approaches, deep learning techniques are used to learn the features and the given task jointly from either a raw speech waveform or from a spectrogram. The approaches have been successful and promising, especially in [10]. It is possible that in the near future feature extraction will lose its importance in any speech processing related pipeline. Nevertheless, feature extraction is still currently used in many speaker diarization related articles [21, 28] which have inspired the work presented in this thesis.

One of the most common feature types in speaker diarization, and also in many other speech related applications, are Mel-Frequency Cepstral Coefficients (MFCCs) [1, 12, 15, 21, 29, 30]. In summary, MFCCs are calculated from a given speech frame in four steps [25]. First, the frame is windowed with a chosen window length and overlap. Next, the logarithmic amplitude spectrum based on Fourier transforms of each window is computed. After this, the amplitude spectrum is converted to a Mel spectrum. Finally, MFCCs are produced from the spectrum by taking a discrete cosine transform. The Mel spectrum is a frequency representation which aims to model human auditory perception [25]. A chosen number of the first coefficients of a discrete cosine transform form the MFCC feature vector [25]. The resulting MFCCs have two desirable properties: they are uncorrelated and compress the auditory information in the frame [25].

MFCC-based features, exactly the same as in [28], are used in this thesis. The first 20 MFCCs are extracted, with window size and shift as 32 and 10 milliseconds, respectively. The related Fourier transforms include frequencies up to half of the sampling rate of 16 kHz. In addition, the first and second order derivatives of the MFCCs are calculated. The final features consist of the derivatives and of all the MFCC coefficients except the first coefficient. This coefficient describes the energy of the corresponding speech window, and is discarded as instructed in [28]. The features are also normalized with zero mean and unit variance.

As a result, for a given speech frame sequence $S$

$$S \rightarrow X = \{\mathbf{x}(\mathbf{s_1}), \mathbf{x}(\mathbf{s_2}), ..., \mathbf{x}(\mathbf{s_N})\}\,, \;\; \mathbf{x}(\mathbf{s_i}) \in \mathbb{R}^{201\times 59}, \tag{6}$$

where $\mathbf{x}(\mathbf{s_i})$ is a sequence of features with 201 windows and 59 features. This odd number of windows results from including also half-windows at the start and the end of each frame $\mathbf{s_i}$.

## 3.2 Speaker modeling

Even if the extracted features would be relevant in terms of human auditory perception they might still be incapable of describing differences between speakers. In other words, given two sequences of features $\mathbf{x_k}$ and $\mathbf{x_j}$, the features could be insufficient to determine if the corresponding speech frames are uttered by the same speaker or by two different speakers. In speaker diarization, this pair comparison, also known as speaker verification [31], is a crucial operation [1, 12]. Consequently, it is necessary to project the sequences to a space which is more suitable for the pair comparison. The projection procedure is known as speaker modeling [32, 33].

### 3.2.1 Related work

Traditionally, speaker modeling has been performed with Gaussian Mixture Models (GMM) [32, 34]. For arbitrarily chosen features $\mathbf{x}^{(i)}$ from a given sequence of features $\mathbf{x}$, a GMM can be defined as a sum of multivariate Gaussians as

$$G(\mathbf{x}^{(i)}) = \Sigma_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}^{(i)} | \boldsymbol{\mu_k}, \Sigma_k), \tag{7}$$

with

$$\mathcal{N}(\mathbf{x}^{(i)} | \boldsymbol{\mu_k}, \Sigma_k) = \frac{1}{(2\pi)^{59/2}} \frac{1}{|\Sigma_k|^{1/2}} exp\{-\frac{1}{2}(\mathbf{x}^{(i)} - \boldsymbol{\mu_k})^T \Sigma_k^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu_k})\}, \tag{8}$$

where $K$ denotes number of mixtures, $\boldsymbol{\mu_k} \in \mathbb{R}^{59}$ means of the mixture, $\Sigma_k \in \mathbb{R}^{59 \times 59}$ a covariance matrix and $\pi_k \in \mathbb{R}$ a mixture weight [32]. Turns out that a GMM configuration can be calculated for each $\mathbf{x_i}$ in $X$ and that the $\mu_\mathbf{k}$ parameters of these configurations can then be exploited to describe the corresponding speaker for each $\mathbf{x_i}$. Each $\mathbf{x_i}$ can then be transformed into a supervector

$$\mathbf{M}(\mathbf{x_i}) \in \mathbb{R}^{K*59} = [\boldsymbol{\mu_1}^T, \boldsymbol{\mu_2}^T, ..., \boldsymbol{\mu_K}^T], \tag{9}$$

which is a more relevant representation of speaker characteristics of $\mathbf{x_i}$ than the features alone [34]. The features, however, have an impact on the GMM. Indeed, the GMMs used in speaker recognition tasks are specifically designed to take MFCC based features as input. MFCCs are uncorrelated and enable using diagonal covariance matrices that simplify GMM computation [35].

In practice, so called supervectors are extracted by utilizing the universal background model (UBM) [34]. This model describes a GMM which is fitted with the

expectation maximization (EM) algorithm to a large number of speakers with the aim of finding a general representation of the speaker characteristics [32]. The UBM enables fitting a GMM to a given $\mathbf{x_i}$ with Maximum A Posteriori adaptation by tuning the GMM parameters of the UBM based on the feature content of $\mathbf{x_i}$ [32]. This adaptation procedure results in a more robust GMM configuration when comparing with an approach where the configuration is determined solely based on the features in $\mathbf{x_i}$ [32]. In general, the pair comparison between two supervectors $\mathbf{M_j}$ and $\mathbf{M_k}$ has been performed using support vector machines (SVM) [34].

However, the supervectors extracted in this manner suffer from an unavoidable shortcoming: the MAP adaptation procedure can be easily influenced by the recording conditions. In other words, the MAP adaptation may emphasize channel characteristics of $\mathbf{x_i}$ more than speaker characteristics. Initially, the solution for this problem was Joint Factor Analysis (JFA) [36] which enabled decomposing $\mathbf{M}$ into speaker and channel dependent factors. In [37], this method was developed further with an interesting finding that the channel factors did have some contribution on the speaker modeling after all. In addition, the concept of the i-vector was introduced with a new formula for $\mathbf{M}$ [37]:

$$\mathbf{M} = \mathbf{M}_{UBM} + \mathbf{Tv}, \tag{10}$$

where $\mathbf{M}_{UBM} \in \mathbb{R}^{K*59}$ is a GMM supervector acquired from UBM, $\mathbf{T} \in \mathbb{R}^{K*59 \times D}$ a total variability matrix and $\mathbf{v} \in \mathbb{R}^{D}$ denotes the i-vector. Just as $\mathbf{M}_{UBM}$, $\mathbf{T}$ can be trained externally with a chosen dimensionality $D$. This can be much smaller than the dimension of $\mathbf{M}$ allowing estimating $\mathbf{v}$ for each $\mathbf{x_i}$ [37]. Powerful channel compensation methods such as linear discriminant analysis (LDA) and within-class covariance normalization (WCCN) [2] can then be be used to model the corresponding speaker of $\mathbf{x_i}$ more compactly and accurately than with $\mathbf{M}$. The most popular methods for speaker verification between two i-vectors $\mathbf{v_i}$ and $\mathbf{v_j}$ have been probabilistic linear discriminant analysis (PLDA) and SVMs [2, 38].

Recently, deep learning based methods have provided alternative approaches for i-vectors. The approaches can be divided roughly to two categories: d-vector based approaches and metric learning based approaches. In the former, a neural network configuration is initially trained in a speaker identification task. In this task the network aims to recognize an identity from a given set of speakers, but indirectly

learns d-vectors, also known as neural speaker embeddings,

$$\mathbf{x} \rightarrow \mathbf{f}(\mathbf{x}) \in \mathbb{R}^d. \tag{11}$$

These embeddings, denoted from here on simply as speaker embeddings, can also model differences between speakers which are not in the speaker set [27, 39, 40, 41]. The embeddings are extracted from some inner layer of the neural network, traditionally from activations of the last hidden layer before an output layer. This, in theory, allows the embeddings to characterize more general differences between speakers [27, 39, 40, 41]. The most common methods of assessing similarity between two embeddings $\mathbf{f_j}$ and $\mathbf{f_k}$ have been distance metrics such as Euclidean distance, cosine distance, and also PLDA [27, 39, 40, 41].

However, the d-vector based methods can also be questioned to some extent. When a neural network is assigned the task of speaker identification, the learning process is based solely on that task and might not result in learning general speaker characterising embeddings [5, 21]. The success in the speaker identification task does not guarantee successful speaker modeling [21]. For this reason, a lot of effort has been put into developing another deep learning approach known as metric learning.

In metric learning, instead of learning speaker embeddings as a side product of speaker identification, the embeddings are learned directly. In this approach, the embeddings corresponding to the same speaker are learned to be similar to each other in training whereas embeddings corresponding to different speakers are forced to be distinct [5, 21, 42, 43]. The similarity is evaluated with some chosen distance metric, usually one of the metrics discussed previously with the d-vector [5, 21, 44]. Although more devised for the comparison of speaker embeddings, metric learning based speaker embedding extractors are slower and more difficult to train than d-vector based extractors [45, 46]. Moreover, metric learning based approaches have also been criticized for not necessarily outperforming the d-vector based approaches even if explicitly designed to do so [42].

Nevertheless, both methods have been shown to outperform i-vector based approaches, especially when evaluating on short utterances [5, 2, 40], which is the case in this thesis. Furthermore, in recent years, d-vector based approaches have advanced rapidly, especially in the field of face recognition, and surpassed the performance of the metric learning based methods [46, 47]. In [10], these methods were shown to also work in speaker recognition with a similar increase in performance. For these reasons, in this thesis a d-vector based speaker modeling method is chosen.

### 3.2.2 SphereSpeaker

Speaker modeling in the proposed speaker diarization system is performed by a novel neural network named SphereSpeaker (SS). It is designed to project a set of feature sequences $X$ into speaker embeddings $F$

$$X \rightarrow F = \{\mathbf{f}(\mathbf{x_1}), ..., \mathbf{f}(\mathbf{x_N})\}, \;\; \mathbf{f}(\mathbf{x_i}) \in \mathbb{R}^{1000}, \|\mathbf{f}(\mathbf{x_i})\|_{\mathbf{2}} = 1, \tag{12}$$

which are $L^2$ normalized and consequently lie on a hypersphere. This normalization is the main reason for naming the network SphereSpeaker. Moreover, as the embeddings will play a crucial role in the system, the whole system is named SphereDiar. As discussed, this projection is performed as part of a speaker identification task, resulting in both speaker embeddings $\mathbf{f}(\mathbf{x})$ and a predicted speaker probability distribution $\mathbf{p}(\mathbf{x})$. This feature is visualized in Figure 3 which describes the neural network architecture of SS. The network can be divided to three main components.

In the first component, an input sequence of features $\mathbf{x}$ is processed with three bidirectional recurrent neural network layers with Long Short-Term Memory (LSTM) cells. The layers are designed to extract speaker characteristics based on information from both the features and their temporal behavior [48]. Each of the layers has 250 hidden units and outputs a sequence of hidden states which are created in the recurrent neural network. The dimensions of the sequences are illustrated in Table 6. The combination of bidirectional LSTM layers with skip connections and concatenation is adhered from [28] and is strongly influenced by [21, 49]. The number of hidden units was determined based on preliminary experiments.

In the second component, the embedding layer, the concatenated sequences are compressed into a 1000-dimensional speaker embedding $\mathbf{f}(\mathbf{x})$. The compression is achieved by a combination of a fully connected layer, which has a Rectified Linear Unit (ReLu) nonlinearity [50], and an average pooling layer. In addition, batch normalization is applied before and after these two layers in order to reduce covariate shift occurring in the training of the network [51, 52]. The embedding layer also includes a final layer which performs $L^2$ normalization

$$\mathbf{f}(\mathbf{x}) \rightarrow \frac{\mathbf{f}(\mathbf{x})}{\|\mathbf{f}(\mathbf{x})\|_2}. \tag{13}$$

This ensures that the embedding is spherical. The configuration of this embedding layer was discovered in preliminary experiments.

Finally, the embedding is transformed into the predicted speaker probability

distribution $\mathbf{p}(\mathbf{x}) \in \mathbb{R}^{N_s}$, where $N_s \in \{1000, 2000\}$ denotes the number of speakers in the speaker identification training set. The transform is implemented by a fully connected layer with the softmax non-linearity, which can be described with the following equations

$$S_i = \mathbf{W_i}^T \mathbf{f} + b_i \tag{14}$$

$$p_i = \frac{e^{S_i}}{\sum_{k=1}^{N_s} e^{S_k}}, \tag{15}$$

where $p_i, \mathbf{W_i} \in \mathbb{R}^{1000}$ and $b_i$ denote the $i$-th sample of $\mathbf{p}(\mathbf{x})$, the $i$-th row of the fully connected layer and the $i$-th element of the bias vector of the layer respectively [46]. As the embedding can be extracted before the transformation, the last fully connected layer is only used when the network is trained. Excluding this layer, the network configuration has around 5.2 million parameters.

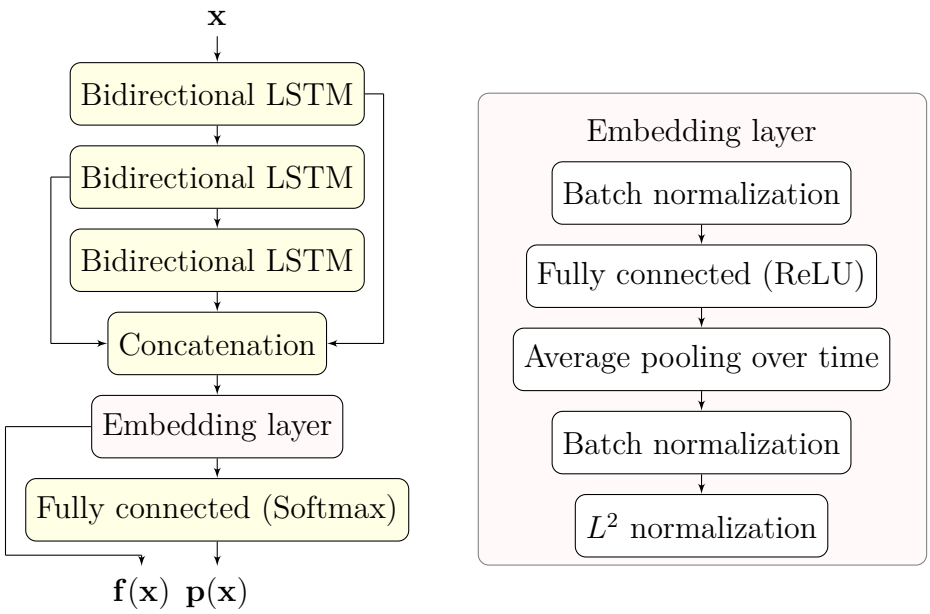


**Figure 3:** The SphereSpeaker neural network.

The embedding layer and especially the use of $L^2$ normalization inside the layer are motivated by the work presented in [10]. In that paper, $p_i$ is formulated slightly differently as

$$p_i = \frac{e^{S_i - \alpha}}{e^{S_i - \alpha} + \sum_{k \neq 1}^{N_s} e^{S_k}}, \tag{16}$$

where $\alpha$ is a constant which is determined before training. In addition, the embedding

**Table 6:** Output dimensions of each layer.

| Layer | Output dimensions |
|---|---|
| Bidirectional $\text{LSTM}_1$ | $201 \times 500$ |
| Bidirectional $\text{LSTM}_2$ | $201 \times 500$ |
| Bidirectional $\text{LSTM}_3$ | $201 \times 500$ |
| Concatenation | $201 \times 1500$ |
| Embedding layer | 1000 |
| Fully connected (Softmax) | $N_s$ |

$\mathbf{f}$ is not $L^2$ normalized before the last fully connected layer but the softmax non-linearity is modified to perform the $L^2$ normalization. In the paper, this modified non-linearity was shown to be superior to not only a normal softmax function but also to metric learning approaches in both speaker identification and verification tasks [10]. A similar configuration was also tested in this thesis but preliminary experiments illustrated that the use of the $\alpha$ term would not yield a considerable performance boost. Furthermore, the use of $\alpha$ also complicated the convergence of the neural network configuration in training. However, the $L^2$ normalization before the non-linearity was found to be very beneficial. This result is discussed in more detail in section 4.

## 3.3 Segmentation

Segmentation means organizing frames $\mathbf{s_i}$ into sequences which are uttered by a unique speaker [1]. It consists of detecting speaker change boundaries and overlapping speech. The main objective of this procedure is to help the clustering, both in excluding overlapping speech from the clustering evaluation and assuring that clustered frames have homogeneous speaker content [1].

### 3.3.1 Related work

The concept of speaker change detection is strongly related to speaker verification but with a few prominent differences. Firstly, instead of comparing two arbitrary chosen sequences of features $\mathbf{x_j}$ and $\mathbf{x_k}$, the comparison is performed with two adjacent sequences $\mathbf{x_i}$ and $\mathbf{x_{i+1}}$. In addition, the utterances from which the sequences are generated are usually shorter in speaker change detection [1, 21]. Moreover, speaker change boundaries can also occur in either of the speech frames from which $\mathbf{x_i}$ and $\mathbf{x_{i+1}}$ [26, 53] have been extracted. As a result, the most popular approaches in speaker change detection differ slightly from the ones introduced in the speaker modeling subsection as will be discussed next.

The earliest approaches have focused on comparing two hypotheses $H_0$ and $H_1$, which assume the sequences $\mathbf{x_i}$ or $\mathbf{x_{i+1}}$ to correspond to either one speaker or two different speakers respectively [1]. The hypotheses have been defined more formally as

$$H_0 : \ \mathbf{x_i}, \mathbf{x_{i+1}} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma) \tag{17}$$

and

$$H_1 : \ \mathbf{x_i} \sim \mathcal{N}(\boldsymbol{\mu_i}, \Sigma_i), \ \ \mathbf{x_{i+1}} \sim \mathcal{N}(\boldsymbol{\mu_{i+1}}, \Sigma_{i+1}), \tag{18}$$

where the sequences are assumed to be generated either by a single Gaussian process or by two different Gaussian processes [1, 54, 55]. In general, the choice of an optimal hypothesis has been made by using Bayesian information criterion (BIC) or generalized likelihood ratio (GLR) [1, 54, 55].

Gaussian modeling has also been exploited in an approach called Gaussian divergence. In this approach, instead of the sequences, static feature vectors $\mathbf{x_s} \in \mathbb{R}^{59}$ are calculated for two adjacent speech frames $\mathbf{s_i}$ and $\mathbf{s_{i+1}}$ [56]. The static vectors for the adjacent speech frames are then modelled as in $H_1$ but the test of hypothesis is based on the value of

$$(\boldsymbol{\mu_{i+1}} - \boldsymbol{\mu_i})^T \Sigma_{i+1} \Sigma_i (\boldsymbol{\mu_{i+1}} - \boldsymbol{\mu_i}), \tag{19}$$

which, naturally, should be small if the speakers are the same and large when speakers are different. In practice, the choice is determined by investigating if the value is below or above a predetermined threshold [56].

Similarly as in speaker modeling, these traditional methods have been recently contested by deep learning approaches. The approaches include both d-vector and metric learning based methods, which, as discussed in the previous subsection, compare speaker embeddings $\mathbf{f_i}$ and $\mathbf{f_{i+1}}$ [21, 57]. The comparison is performed using using either the cosine or the euclidean distance metric [21, 57]. In addition, methods for determining if a given sequence of features $\mathbf{x}$ includes a speaker change boundary have been proposed. In these methods, a neural network transforms the sequence into a real valued prediction

$$\mathbf{x} \rightarrow h(\mathbf{x}) \in \mathbb{R}, \tag{20}$$

which can be interpreted as the probability of a speaker change event [26, 53]. After calculating the probability, the belief of whether the sequence includes a speaker change boundary or not is determined, as in Gaussian divergence, by using a threshold.

In literature, all the aforementioned approaches have been shown to surpass the traditional methods in terms of speaker change detection accuracy [21, 26, 53, 57].

Overlapping speech detection refers to determining if the corresponding frame $\mathbf{s}$ of $\mathbf{x}$ includes multiple speakers speaking simultaneously [1]. In this setting, related work has focused on two approaches: hidden Markov models (HMM) and deep learning methods. In the former, the sequence $\mathbf{x}$ is considered as observations and the $i$th features $\mathbf{x}^{(i)}$ assigned either with a speech, a non-speech or an overlapping speech label, which correspond to three different hidden states in an HMM [15, 16, 58]. Emission probabilities of each hidden state have been modeled with GMMs [15, 16, 58]. Experimenting with a variety of different features has also been a trend in these approaches. In addition to MFCC features, several other acoustic feature sets have been proposed such as linear predictive coding residual values, spectral flatness and harmonic energy ratio [15, 58]. In [16], long-term conversational features describing silence and speaker change statistics have also been shown to be relevant in the overlapping speech detection task.

Deep learning based methods have mostly applied LSTM and bidirectional LSTM based recurrent neural network configurations [17, 59]. In [17], a recurrent neural network with LSTM cells is used with the objective of assigning the last features $\mathbf{x}^{(201)}$ of a given feature sequence $\mathbf{x}$ to one of three classes mentioned previously: non-speech, overlapping speech and speech. In this approach, the central idea has been to use a contextual information of previous features in the sequence $\mathbf{x}$ to influence and justify the prediction for the last features. In their experiments, also several other features mentioned in previous paragraph has been exploited in tandem with MFCCs [17]. Obtained results have been comparable with the results attained by using HMMs [17].

In [59], a bidirectional neural network with LSTM cell was used with the same input $\mathbf{x}$ but with a slightly different objective. Instead of assigning $\mathbf{x}^{(201)}$ to one of the three classes mentioned in previous paragraph, the features are labeled to four different classes. The classes depict how well the features correspond to either male or female voice, overlapping speech or non-vocal sounds [59]. In this approach, artificially generated overlapping speech training data was also experimented with. The results in overlapping speech detection surpassed the ones provided in [17].

However, to the best of the writers knowledge, no approaches which perform overlapping and speaker change detection jointly have been proposed in the literature. In theory, this joint detection detection could be beneficial as overlapping speech and speaker changes have been shown to often occur simultaneously, especially in

spontaneous speech [16]. In this thesis, preliminary experiments concurred with this hypothesis and a method for the joint detection was developed.

### 3.3.2 Homogeneity Based Segmentation

Homogeneity Based Segmentation (HBS) is a novel segmentation technique in which a neural network, depicted in Figure 4, performs a binary classification for a given $X$

$$X \rightarrow H = \{h(\mathbf{x_1}), ..., h(\mathbf{x_N})\}\,, \;\; h(\mathbf{x_i}) \in \{0, 1\}\,, \tag{21}$$

with targets $\Sigma = \{\sigma(\mathbf{x_1}), ..., \sigma(\mathbf{x_N})\}$, where

$$\sigma(\mathbf{x}) = \begin{cases} 0, & \text{if } H_{\%\mathbf{x}} \geq H_{\theta\%} \\ 1, & \text{otherwise} \end{cases} \tag{22}$$

where $H_{\%\mathbf{x}}$ is a homogeneity percentage of the corresponding $\mathbf{s}$ of $\mathbf{x}$ and $H_{\theta\%}$ some threshold percentage. The classification can be equivalently interpreted as a segmentation procedure for the sequence of frames $S$ from which $X$ is extracted, where the segments consist of adjacent speech frames $\mathbf{s}$ with the same $h(\mathbf{x})$ value. In other words, $S$ is segmented based on homogeneity percentages. Consequently, the segments are of two types: single speaker segments, labeled as 0, and segments with multiple speakers, labeled as 1. The motivation behind the segmentation is then to exclude all the frames in the 1-class from the clustering procedure.

The $H_{\%\mathbf{x}}$ percentages, however, are not generally known. Hence, the value of $h$ is in practice defined as

$$h(\mathbf{x}) = \begin{cases} 0, & \text{if } \hat{h}(\mathbf{x}) \leq \theta \\ 1, & \text{otherwise} \end{cases} \tag{23}$$

where $\theta \in [0, 1]$ depicts a given threshold and $\hat{h}(\mathbf{x}) \in [0, 1]$ an output of the neural network architecture which is trained with given target $\{\Sigma_1, ..., \Sigma_M\}$ and feature sets $\{X_1, ..., X_M\}$. The calculation of the output and the structure of the network can be summarized as follows.

First, an input $\mathbf{x}$ is transformed into $\mathbf{y}$ consisting of 201 sequences with 600 elements using a bidirectional recurrent neural network layer with LSTM cell as illustrated in Table 7. The number of elements results from the use of 300 hidden units in the LSTM cell and from the fact that the recurrent neural network is bidirectional. The number of units was determined based on preliminary experiments.

Next, $\mathbf{y}$ is fed to an attention layer in which an attention matrix $\mathbf{A}$

$$\mathbf{y} \rightarrow \mathbf{y}^T \in \mathbb{R}^{600\times 201} \rightarrow \mathbf{p}(\mathbf{y^T}) \in \mathbb{R}^{600\times 201} \rightarrow \mathbf{p}(\mathbf{y^T})^T \in \mathbb{R}^{201\times 600} = \mathbf{A} \qquad (24)$$

is created and multiplied element-wise with $\mathbf{y}$. The output of this multiplication is then processed in an average pooling and a batch normalization layer. Finally, the output is compressed into a single value $\hat{h} \in [0, 1]$ in a fully connected layer with a sigmoid activation function. The class $h(\mathbf{x}) \in \{0, 1\}$ of each $\mathbf{x}$ is determined based on rounding the output of the network $\hat{h}(\mathbf{x}) \in [0, 1]$ to the nearest integer. The network has around one million parameters.

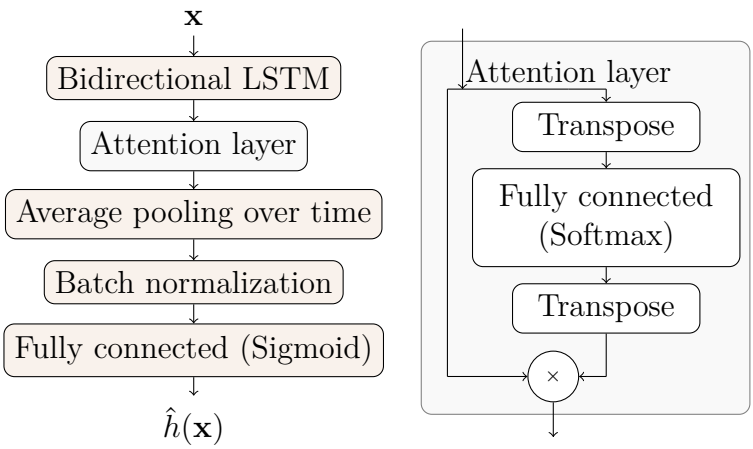


**Figure 4:** The HBS neural network.

**Table 7:** Output dimensions of each layer.

| Layer | Output dimensions |
|---|---|
| Bidirectional $\text{LSTM}_1$ | $201 \times 600$ |
| Attention layer | $201 \times 600$ |
| Average pooling | 600 |
| Fully connected (Sigmoid) | 1 |

The network is most influenced by the work in [17, 53, 59], in which recurrent neural networks with LSTM cells have been used to detect either speaker changes or overlapping speech. However, in these works, instead of a single LSTM layer, multiple layers were used. A similar approach was also investigated in this thesis but the increase in the number of layers did not help in the classification task.

The use of an attention layer is based on an implementation introduced in [60]. This layer is intuitively motivated. For example, when considering a frame which includes a speaker change boundary, only feature sequences close to this boundary

should be relevant for HBS. In other words, some features should be addressed with more attention than others. Similar reasoning can also be applied for overlapping speech detection. The layer was also found to be beneficial in practice as will be shown in section 4.

All other components of the architecture were determined based on preliminary experiments and the SphereSpeaker neural network which showed the benefit of using batch normalization and average pooling. The output layer was also tested with a softmax activation function and with 2-dimensional output, but this choice did not bring any performance improvements.

## 3.4 Clustering

Clustering is a central task in any speaker diarization system [1]. In this task, a given number of observations, which could be for example GMMs or speaker embeddings, are organized into groups. These groups then determine a speaker label for each observation.

### 3.4.1 Related work

One of the earliest and most popular clustering methods in speaker diarization has been hierarchical agglomerative clustering (HAC), also known as bottom-up clustering [1, 61]. In this approach, the first step is to initialize a cluster for each segment in $\mathbf{S} = \{S_1, ..., S_n\}$, which are obtained from the segmentation procedure discussed in the previous subsection. Then, these clusters are merged in an iterative process until an optimal number of clusters, i.e. the number of speakers, is attained [1].

In practice, each cluster $S_i$, has been modeled either by fitting a GMM to the corresponding $X_i$ or by using i-vectors or speaker embeddings [1, 12, 13, 22, 61]. Of these cluster modeling methods, GMMs have been the traditional approach. This approach has been accompanied with HMMs which have been used for modeling the sequence structure of $\mathbf{S}$ [1]. The purpose of HMMs is to represent speakers as states and speaker changes as transitions and the GMMs serve as the emission models [1, 61]. The choice of which clusters are to be merged has been made based on the similarity of their corresponding GMMs. After each merging, a new GMM is fitted for each of the new clusters. In this process, also Viterbi realignment has been used to rearrange segments in $\mathbf{S}$ after each merging iteration [1, 61]. BIC and GLR have been popular stopping criteria for cluster merging [1, 61].

Recently, GMMs have been replaced with i-vectors and speaker embeddings [12, 13, 22] in HAC. These representations have been also assigned to considerably smaller segments $\mathbf{s_i}$, or frames as they are called in this thesis, that has been beneficial for the clustering operation [22, 62]. Otherwise, HAC is used with a similar procedure, by first assigning a cluster for each $\mathbf{s_i}$ in $S$ and then by iteratively merging them based on the similarity of the representations. The similarity has been determined based on cosine distance or PLDA [12, 13, 22, 63]. In addition, PLDA has also been exploited to find a stopping criterion for HAC [22, 62]. The results obtained combining HAC with the representations have been generally very promising and therefore this combination has been considered the state-of-the-art in speaker diarization [12, 13].

However, in addition to approaches using bottom-up clustering, also so called top-down clustering methods have been proposed [1, 61]. In these methods, the objective is to first assign $\mathbf{S}$ to a single cluster and then to break it down into multiple clusters which describe the speakers in $\mathbf{S}$. Again, clustering is done iteratively, using HMM-GMMs and realigning with the Viterbi algorithm [1, 61]. Nevertheless, top-down clustering has not been able to outperfrom its bottom-up counterpart and furthermore, it is not compatible with i-vectors or speaker embeddings [1, 61].

In [64], an Integer Linear Programming (ILP) based clustering method operating on i-vectors is introduced, contrasting the iterative procedures in both bottom-up and top-down clustering approaches. The objective of this method is to minimize

$$z = \sum_{k=1}^{K} y_k + \sum_{k=1}^{K} \sum_{n=1}^{N} D(\mathbf{v_k}, \mathbf{v_n}) x_{kn}, \tag{25}$$

where $K$ denotes a number of clusters, $N$ a number of i-vectors to cluster, $y_k \in \{0, 1\}$, $x_{kn} \in \{0, 1\}$, $\mathbf{v}$ an i-vector and $D$ a distance metric calculated for each i-vector pair. In addition, three conditions are assigned:

$$\sum_{n=1}^{N} x_{kn} = 1, \;\; x_{kn} - y_n \leq 0, \;\; D(\mathbf{v_k}, \mathbf{v_n}) x_{kn} \leq \delta, \tag{26}$$

where $\delta$ depicts a given threshold. Interestingly, the minimization of $z$ can then be interpreted equivalently as an optimal clustering for i-vectors with $x_{kn}$ values depicting a cluster for each $\mathbf{v_n}$ [64]. Moreover, the results obtained with ILP have been comparable or even better than the ones obtained using HAC with i-vector cluster modeling [64].

In addition to the aforementioned methods, also clustering approaches in which the number of speakers must be known beforehand have been experimented in speaker

diarization. These approaches include K-means and spherical K-means algorithms and the use of von Mises-Fisher distributions [65, 66]. Although these algorithms have shown promise, they naturally can not be relied on if the number of speakers in $S$ is unknown [65, 66].

Nonetheless, with many audio streams such as meetings, movies and TV shows, a reasonable estimate of the maximum number of speakers can almost always attained prior to a speaker diarization task. Thus, the aforementioned algorithms can be used in a two phase approach. First, multiple alternative clustering proposals are generated for different numbers of speakers. Second, an optimal proposal is chosen based on some evaluation measure. Yet, it seems that virtually no approaches in speaker diarization related literature exploit the two phase approach.

Consequently, such an approach is developed in this thesis. This approach consists of two essential components: Spherical K-means algorithm for clustering and silhouette coefficients for the evaluation of the clustering proposals. These components are then fused together to form a novel clustering algorithm.

### 3.4.2 Spherical K-means

Spherical K-means is a clustering algorithm which allows grouping given observations into clusters. In this thesis, the observations are speaker embeddings $\mathbf{f}$ and clusters their presumed speaker groups. In this context, spherical K-means finds cluster centers $C = \{\mathbf{c_1}, ...\mathbf{c_K}\}$, with $\mathbf{c_j} \in \mathbb{R}^{1000}, \|\mathbf{c_j}\|_2 = 1$, which maximize a cosine similarity sum objective [67]

$$O = \sum_{\mathbf{f} \in F} \mathbf{f}^T \mathbf{c}_{l(\mathbf{f})}, \tag{27}$$

where $F$ is a set of speaker embeddings and

$$l(\mathbf{f}) = \arg\max_{l} \mathbf{f}^T \mathbf{c_l} \in \{1, ..., K\}, \tag{28}$$

depicts the cluster, i.e. the predicted speaker label of a given $\mathbf{f}$.

In other words, the algorithm locates $K$ cluster centers in a such manner that the sum of cosine distances from the cluster centers to a given set of speaker embeddings $F$ is maximized. This sum consists of smaller summations which are computed from the cosine distances between each cluster center and their nearest speaker embeddings. The speaker labels are then assigned based on the equation (25). With this setting, the number of clusters $K$ must be given as an input for the algorithm.

In practice, the algorithm is calculated iteratively for a given $F$ and $K$ with following steps [67]:

1. Initialize centers $C = \{\mathbf{c_1}, ...\mathbf{c_K}\}$, with $\mathbf{c_j} \in \mathbb{R}^{1000}, \|\mathbf{c_j}\|_2 = 1$.
2. Calculate labels $L = \{l_1, ...l_M\}, l_i \in \{1, ...K\}$ by assigning $l_i = \arg\max_j \mathbf{f_i}^T \mathbf{c_j}$ for each $\mathbf{f_i}$ in $F$.
3. Update values of each $\mathbf{c_j}$ by computing

$$\mathbf{c_j} = \frac{\sum_{\mathbf{f} \in F_j} \mathbf{f}}{\left\| \sum_{\mathbf{f} \in F_j} \mathbf{f} \right\|_2},$$

   where $F_j = \{\mathbf{f_i} \mid l_i = j\}$.
4. If $L$ is left unchanged, return $L$ and $C$, otherwise return to step 2.

However, although the algorithm can always find a set of $\{L, C\}$ which maximize the objective $O$, also other such sets may exist. In other words, the algorithm converges to a local maximum [67]. For this reason, it is necessary to use a proper initialization method for $C$ [68] and run the algorithm multiple times with some validation criterion to find an optimal set of $\{L, C\}$. In this thesis, the initialization is performed with K-means++ [68] and validation with silhouette coefficients. The use of K-means++ has been suggested in [68] whereas the utility of silhouette coefficients is based on preliminary experiments.

### 3.4.3 Silhouette coefficients

Silhouette coefficients arise from the concept of average dissimilarity. Assuming $F$ with labels $L$ computed using the spherical K-means algorithm, the average dissimilarity of a given $\mathbf{f_i}$ to embeddings in $F_k = \{\mathbf{f_j} \mid l_j = k\}$ can be computed as [69]

$$d_k(\mathbf{f_i}) = \frac{1}{M} \sum_{\mathbf{f} \in F_k} \mathbf{f_i}^T \mathbf{f}. \tag{29}$$

Based on this formulation and assuming that $\mathbf{f_i}$ is labeled as $m$, two more specific dissimilarity metrics can be defined:

$$a(\mathbf{f_i}) = d_m(\mathbf{f_i}) \text{ and } b(\mathbf{f_i}) = \max_{k \neq m} d_k(\mathbf{f_i}), \tag{30}$$

where $a(\mathbf{f_i})$ describes the average dissimilarity of $\mathbf{f_i}$ to all speaker embeddings in the same cluster as $\mathbf{f_i}$ and $b(\mathbf{f_i})$ to embeddings in the nearest cluster in terms of cosine similarity. The silhouette coefficient is then calculated as [69]

$$sc(\mathbf{f_i}) = \frac{a(\mathbf{f_i}) - b(\mathbf{f_i})}{\max\{a(\mathbf{f_i}), b(\mathbf{f_i})\}}. \tag{31}$$

Consequently, $-1 \leq sc(\mathbf{f_i}) \leq 1$, with value 1 indicating that $\mathbf{f_i}$ is well clustered and value $-1$ suggesting the opposite. An evaluation score for clustering $F$ is obtained by calculating the average of all $sc(\mathbf{f_i})$ coefficients as

$$s = \frac{1}{M} \sum_{\mathbf{f} \in E} sc(\mathbf{f}), \tag{32}$$

which is named the silhouette score in this thesis. This value has the same bounds as the silhouette coefficients and can be interpreted as an overall clustering score for the speaker embeddings in $F$.

### 3.4.4 Top Two Silhouettes

After the speaker modeling and segmentation, speaker embeddings $F$ and a sequence of HBS labels $H$ have been obtained. As a final step, each $\mathbf{f}$ in $F$ is assigned a speaker label $l$:

$$F \rightarrow L = \{l(\mathbf{f_1}), ..., l(\mathbf{f_N})\}, \;\; l(\mathbf{f_i}) \in \{1, ..., N_s\}, \tag{33}$$

where $N_s$ depicts the predicted number of speakers. This assignment is attained by clustering $E$, a subset of $F$ consisting of embeddings $\mathbf{f_i}$ which have the HBS label $h_i = 0$. The clustering is performed using a novel algorithm which can be divided into two steps: the proposal generation and the optimal proposal determination.

In the first step, $E$ is fitted with multiple different spherical K-means configurations with $K$ ranging from 2 to $N_{max}$, where $N_{max}$ is chosen to be higher than the true number of speakers in $E$. Each configuration is run with $R$ different initializations, from which the final configuration is determined based on the run which yielded the highest silhouette score. The proposals $P_i$, consisting of a set of cluster centers $C_i$ and speaker labels $L_i$, are then created based on these final configurations.

It is worth noticing that the proposals could also be chosen based on their $O$ values in equation (23). This approach, however, was outperformed by the use of silhouette coefficients in preliminary experiments. In these experiments, the use of

coefficients decreased the number of initializations needed but still attained similar or better final silhouette scores.

In the second step, the optimal proposal $P_{opt}$ is chosen. First, the proposals corresponding to the two largest silhouette scores, $P_{top-1}$ and $P_{top-2}$ are recovered. If (i) $P_{top-1}$ has more clusters, or (ii) the silhouette score of $P_{top-2}$ is below a threshold $\delta$, then $P_{opt} = P_{top-1}$. This is a heuristic rule which was found experimentally and can be interpreted as further evidence that $P_{top-1}$ is the optimal proposal.

Otherwise, if both (i) and (ii) are unsatisfied, the algorithm deduces that also $P_{top-2}$ could be chosen. As $P_{top-2}$ has then more clusters than $P_{top-1}$, the algorithm investigates if any of the clusters in $P_{top-1}$ would contain inner clusters. This investigation is performed in a similar fashion as in the first step but for each cluster in $P_{top-1}$. The assignment $P_{opt} = P_{top-2}$ is then obtained if for any initialization or cluster, both the maximum silhouette value is above $\delta$ and the corresponding $K \in \{2, 3\}$. In this condition, the maximum number of inner clusters is restricted to 3 since a higher number would be highly improbable. However, if this condition is not satisfied the algorithm chooses $P_{opt} = P_{top-1}$.

Top Two Silhouettes is described more formally in Algorithm 1. In this description, spherical K-means is denoted with $\phi$ and the calculation of the silhouette score with a variable $\upsilon$. Moreover, instead of two steps, the description consists of three main steps consisting of the calculation of the silhouette scores, the evaluation of the conditions (i) and (ii) and the possible inner cluster search.

The motivation behind the inner cluster search in the second step is based on the preliminary experiments. In these experiments, the speaker embeddings and their predicted clusters were visualized with t-SNE [70] in 2D. The visualizations illustrated that in many cases, $P_{top-1}$ would result in an underestimate of the number of clusters. An example of this is shown in Figure 5. This Figure describes two clustering proposals $P_{top-1}$ and $P_{top-2}$ projected to 2D. The proposals are created for an example meeting taken from the ICSI corpus with 5 participants. As can be seen from this Figure, all clusters are distinguishable but $P_{top-1}$ has merged two of these clusters. However, $P_{top-2}$ has assigned all the clusters correctly. In this Figure, it is clear that such an inner cluster search as described in the second step is beneficial. In general, however, this inner cluster search needs to be regularized in some manner since some meeting recordings in the meeting corpus do have $P_{top-1}$ as the best choice. The heuristic rules in both the first and second steps were developed for this reason.

Finally, the labels $L$ for $F$ are generated using the associated cluster centers $C_{opt}$

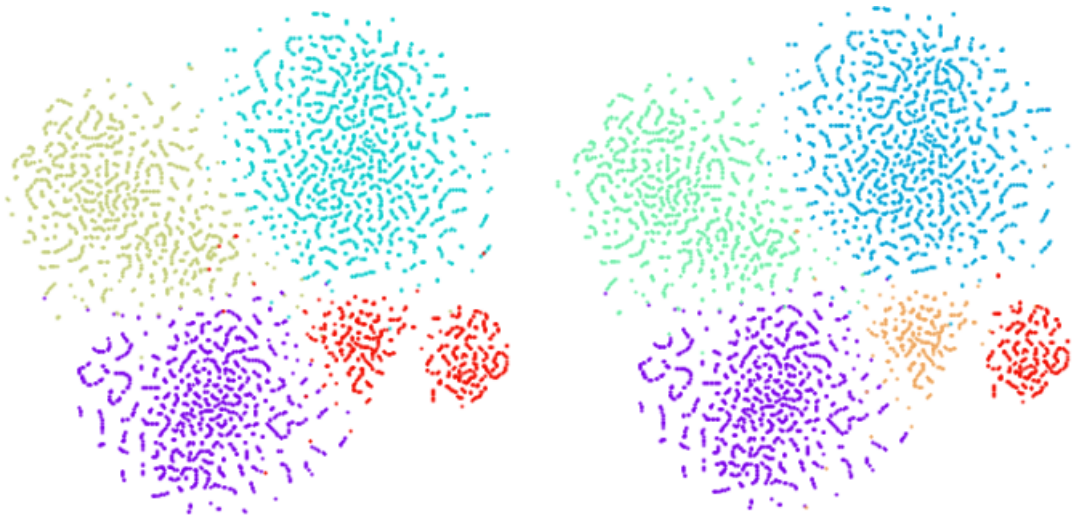

**Figure 5:** Example clustering proposals in 2D, $P_{top-1}$ in the left and $P_{top-2}$ in the right.

of $P_{opt}$. As the proposals corresponding to the two largest silhouette scores are central for the algorithm, it is named Top Two Silhouettes. The validity of this algorithm is demonstrated in the experiments section where it is compared with Top Silhouette (TopS) which is essentially the same as Top2S but always assigns $P_{opt} = P_{top-1}$.

As a final remark, it must be mentioned that Top2S is rather ill suited for situations were some of the clusters contain significantly less speakers than other clusters. For instance, if one speaker would only speak for 2 seconds, this utterance would be described by a single speaker embedding. Naturally, spherical K-means will be unable to cluster this speaker correctly and this also applies to Top2S. Preliminary experiments suggested that this problem was still present in situations were the total amount of speech material for speaker would be close to 10 seconds. This is due to the fact that Top2S does not utilize the time varying structure of a given set of speaker embeddings in any meaningful way. That is, the algorithm treats the embeddings as they would be independent from each others. This statement is not completely true since in some cases the differences between adjacent speaker embeddings may imply speaker change boundaries. This was discussed in subsection 3.3.1. The imbalance is present in the ICSI corpus to some extent but not in the AMI corpus.

**Algorithm 1:** Top Two Silhouettes
**Input:** Speaker embeddings $E$, a number of initializations $R$, a maximum number of centers $N_{max}$ and a threshold $\delta$
**Output:** Proposal $P = \{L, C\}$.
**Steps:**

1. Initialize $K = \{2, ..., N_{max}\}$ and
   $s = \{0, ..., 0\}, |s| = |K|$

2. **for** $r = 1$ to $R$ **do**
   **for** $i = 1$ to $|K|$ **do**
   $\phi(K_i, E) \rightarrow L_i \rightarrow \upsilon(L_i, E) \rightarrow \hat{s}_i$
   **if** $(\hat{s}_i > s_i) \rightarrow s_i = \hat{s}_i$.

3. Find largest and second largest silhouette values $s_{top-1}$ and $s_{top-2}$, respectively.
   **If not** top-2 > top-1 $\wedge s_{top-2} > \delta$)
   $\rightarrow$ **return** $L_{top-1}, C_{top-1}$

4. Repeat step 2 for each $E_j \in E = \{\mathbf{f_i} \mid l_i = k \in L_{top-1}\}$ In the process, for any $j, r$:
   **If** $(\max_i \upsilon(L_{ij}, E_j) > \delta \wedge K_i \in \{2, 3\})$
   $\rightarrow$ **return** $L_{top-2}, C_{top-2}$

5. **return** $L_{top-1}, C_{top-1}$.

# 4 EXPERIMENTS

This section presents the conducted experiments and discusses the obtained results. It consists of four subsections of which the first introduces the evaluation metrics. The following three investigate the performance of the proposed segmentation and speaker modeling methods separately and finally illustrate the results obtained with SphereDiar. These results also include comparisons with two other state-of-the-art speaker diarization systems.

## 4.1 Evaluation metrics

### 4.1.1 Adjusted Rand score

Let us assume that a set of speaker embeddings $F$ can be partitioned to either $F_L = \{F_{L1}, ..., F_{Lp}\}$ or $F_{\hat{L}} = \left\{F_{\hat{L}1}, ..., F_{\hat{L}q}\right\}$ based on a reference label set $L$ and a predicted label set $\hat{L}$, respectively. Equivalently, $F_{Li} = \{\mathbf{f_j} \in F|\ l_i = i\}$ and $F_{\hat{L}i} = \left\{\mathbf{f_j} \in F|\ \hat{l}_i = i\right\}$. In this case, it is assumed that speaker labels $l, \hat{l}$ do not depict overlapping speech labels. In addition, let us define three variables:

$$\begin{aligned} n_{ij} =& |F_{Li} \cap F_{\hat{L}j}|, \\ a_i =& \sum_{j=1}^{q} |F_{Li} \cap F_{\hat{L}j}|, \\ b_j =& \sum_{i=1}^{p} |F_{Li} \cap F_{\hat{L}j}|. \end{aligned} \tag{34}$$

Then, adjusted Rand score $AR$ can be written as [71]

$$AR = \frac{\sum_{i,j} \binom{n_{ij}}{2} - \sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} / \binom{N}{2}}{\frac{1}{2}(\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}) - \sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} / \binom{N}{2}}, \tag{35}$$

which is bounded in $[-1, 1]$ and can be interpreted as a clustering performance measure with the value 1 representing optimal clustering and the value $-1$ the opposite. In this thesis, the score is used in speaker modeling evaluation. The use of this score is motivated based on its computational simplicity and success as a clustering evaluation measure [71].

### 4.1.2 Mean average precision

In this thesis, mean average precision (MAP) is used to evaluate HBS models. In this setting, the idea is to validate a binary classification task performed on $X = \{\mathbf{x_1}, ..., \mathbf{x_N}\}$ with target labels $\Sigma = \{\sigma_1, ..., \sigma_N\}$ and predicted labels $H = \{h_1, ..., h_N\}$ when $\theta$ which is utilized in the creation of $H$ is varied. The validation and computation of MAP is based on precision and recall which are defined for each $\theta$ as [72]

$$Pr_\theta = \frac{|\{\mathbf{x_i}|\ \sigma_i = 1\} \cap \{\mathbf{x_i}|\ h_i = 1\}|}{|\{\mathbf{x_i}|\ h_i = 1\}|} \tag{36}$$

$$Re_\theta = \frac{|\{\mathbf{x_i}|\ \sigma_i = 1\} \cap \{\mathbf{x_i}|\ h_i = 1\}|}{|\{\mathbf{x_i}|\ \sigma_i = 1\} \cap \{\mathbf{x_i}|\ h_i = 1\}| + |\{\mathbf{x_i}|\ h_i = 0\} \setminus \{\mathbf{x_i}|\ \sigma_i = 0\}|}. \tag{37}$$

Assuming $T$ different threshold values $\theta_i$ with $\theta_i > \theta_{i-1}$, MAP can then be approximated as

$$\text{MAP} = \sum_{i=1}^{T} (Re_{\theta_i} - Re_{\theta_{i-1}}) Pr_{\theta_i}, \tag{38}$$

where $\theta_0 = 0$ and $Re_{\theta_0} = 1$. With this definition, MAP can be understood as the area under the precision-recall curve.

In summary, MAP is targeted for a binary classification evaluation when the classification is based on some given threshold. However, this task could also be addressed by using area under the receiver operating characteristics (ROC) curve [73]. This measure was also investigated in preliminary experiments but it was found to be overoptimistic. The deficiency was due to a rather high class imbalance in the meeting corpus. Even when the imbalance was addressed in many ways, as will be discussed later, the measure was found to be suboptimal compared to MAP.

### 4.1.3 Diarization error rate

Similarly as previously, let us assume $F$ with predicted and reference speaker labels $\hat{L}$ and $L$. Diarization error rate (DER) is then calculated between $L$ and $\hat{L}$ with the following steps. First, all speaker embeddings with corresponding $H_{\%}$ smaller than a given $H_{\theta\%}$ threshold are excluded. This means that all frames with overlapping speech and also some frames which include multiple speakers are not considered in the evaluation process. As a result, a speaker embedding subset $E$ and speaker label subsets $L_s$ and $\hat{L}_s$ are generated.

Secondly, the partitions $E_{L_s}$ and $E_{\hat{L}_s}$ are composed based on $L_s$ and $\hat{L}_s$ similarly as discussed with adjusted Rand index. In this setting, the variable $n_{ij}$ is defined as

$$n_{ij} = |E_{L_s i} \cap E_{\hat{L}_s j}|, \tag{39}$$

with a so called confusion matrix, which is formulated as

$$\mathbf{C} \in \mathbb{N}^{N_s \times N_s} = \{n_{ij} |\ i, j \in \{1, ..., N_s\}\}, \tag{40}$$

where $N_s$ is the number of unique labels in $L_s$. Finally, DER is computed as

$$DER = 1 - \frac{\sum_{k=1}^{N_s} A_k}{|L_s|}, \tag{41}$$

where $\{A_1, ..., A_{N_s}\}$ describe $n_{ij}$ values which are the solution to a linear assignment which minimizes the confusion in $\mathbf{C}$. These values are obtained using the Hungarian algorithm [74]. In this formulation, the DER calculation in this thesis is essentially the same as generally performed in the speaker diarization literature [1, 74].

## 4.2 Speaker modeling

Experiments conducted on speaker modeling concentrate on evaluating the SS neural network based on the identification accuracy and the relevance of speaker embeddings when

1. the $L^2$ normalization layer is either used or not
2. the training and evaluation sets are varied

In practice, 8 neural network configurations are trained in total with model architectures SS and SS$^*$, where the $L^2$ normalization layer is excluded. Training and evaluation are performed with all four partitions of the speaker corpora.

### 4.2.1 Training procedure

The training and evaluation sets for each partition are generated by randomly choosing 45 frames from each speaker for testing and leaving the rest for training. As a result, evaluation sets for $LS_{1000}$ and $VC_{1000}$ consist of 45000 frames and the sets for $LS_{2000}$ and $VC_{2000}$ of 90000 frames. Model training is performed with the configuration depicted in Table 8. In this process, the model candidate is saved after each epoch

and the final model is chosen based on the candidate with the lowest categorical cross-entropy value on an evaluation set.

**Table 8:** Training configuration

| | |
|---|---|
| GPU | Quadro P5000 |
| Optimizer | Adam [75] |
| Loss function | Categorical cross-entropy |
| Batch size | 256 |
| Epochs | 45 |

### 4.2.2 Results

The results can be divided into two categories: identification accuracy scores, depicted in Table 9, and speaker embedding validation scores in Table 10. The accuracy scores are calculated based on the identification predictions on the evaluation sets of each partition. The validation scores are adjusted Rand scores, which are determined based on the following procedure. First, all frames in a given partition test set are transformed into speaker embeddings. Next, these embeddings are clustered using spherical K-means with $K$ being the number of speakers in the partition. Then, the obtained predicted labels are compared with the reference and the adjusted Rand score is computed. This procedure allows investigating both the intra-class and the inter-class variation of the created embeddings.

The identification accuracy of each model with the corresponding partition evaluation set is presented in Table 9. The implications of these scores can be categorized as follows. Firstly, the identification task on $LS$ partitions is easier than with $VC$ partitions for both model architectures. This result is as expected, since the Librispeech corpus includes a lot less variation in recording conditions and noise backgrounds. However, the results with the $VC$ partitions are also fairly promising. For instance, the identification accuracy with Voxceleb, a predecessor of Voxceleb2, containing around 1200 speakers, has been reported to be around 80% in [5].

Secondly, the number of speakers does not seem to affect the identification accuracy drastically. A decrease in the accuracy is only seen with the $VC$ partitions and even then it is not very significant considering a substantial increase in the number of speakers. One hypothesis for this result is the variation in the training sets. Even though the difficulty of the identification task increases when more speakers are involved, so does the number of different speakers which the neural network processes in training. Thus, the model needs to learn more specific ways to discriminate

between speakers in training which in turn benefits the identification accuracy on an evaluation set. This hypothesis, however, can be reasonable only when the number of frames for each speaker is high enough and balanced as is the case with the $LS$ and $VC$ partitions.

Most importantly, the scores illustrate that $L^2$ normalization is beneficial in terms of identification accuracy with each model. This increase is not substantial when considering $LS$ but on the $VC$ partitions the benefit of $L^2$ normalization is noticeable. A similar result has also been obtained in [10] on the previously mentioned Voxceleb dataset.

**Table 9:** Identification accuracies (%)

| | SS* | SS |
|---|---|---|
| $LS_{1000}$ | 99.2 | **99.8** |
| $LS_{2000}$ | 99.4 | **99.8** |
| $VC_{1000}$ | 88.8 | **90.2** |
| $VC_{2000}$ | 87.1 | **88.7** |

The results in Table 10 in turn indicate that the use of an $L^2$ normalization layer is also advantageous in terms of speaker embedding quality. With the $LS$ partitions the relative improvement is around 25%, which, given the magnitudes of the adjusted Rand scores, is a considerable upgrade. However, with the $VC$ partitions, the improvement exceeds 100% for both $VC_{1000}$ and $VC_{2000}$.

Moreover, the adjusted Rand scores on the $LS$ partitions suggest that the SS architecture does succeed in creating relevant speaker embeddings. Even when the numbers of clusters are in the thousands, the scores are very high. The scores attained with SS* are also promising, implying that the general model architecture of $SS$ is suitable for speaker embedding generation.

With the $VC$ partitions and SS, however, the adjusted Rand scores are only about half as good as they are with the $LS$ partitions. With SS*, these scores are less than 30%. The reasons for this performance drop are essentially the same as the ones discussed with identification accuracy results but lead to a question: have the neural networks trained on the $VC$ partitions actually learned relevant ways to discriminate between different speakers? If the answer were only based on the adjusted Rand scores in Table 10, it would most certainly be no, especially when considering SS*. However, the clustering objective from which the adjusted Rand scores are obtained is not easy to reach, especially considering the challenging recordings of $VC$. Furthermore, the speaker modeling ability of the neural networks

also needs to be tested with speakers which are not in the training set. Thus, the validation of the proposed speaker embedding extraction approach continues in the SphereDiar subsection.

**Table 10:** Adjusted Rand scores

| | SS* | SS |
|---|---|---|
| $LS_{1000}$ | 0.75 | **0.94** |
| $LS_{2000}$ | 0.71 | **0.89** |
| $VC_{1000}$ | 0.20 | **0.51** |
| $VC_{2000}$ | 0.18 | **0.42** |

## 4.3 Segmentation

Experiments in segmentation are rather similar to the ones previously discussed with speaker modeling. Namely, the experiments investigate

1. The effect of an attention layer and dropout on the HBS neural network
2. The performance of the HBS neural network with different training and evaluation sets

The training and evaluation sets are extracted from the meeting corpus. 6 different neural network configurations are trained and compared in total.

### 4.3.1 Training procedure

Two different training and evaluation set configurations are collected from the meeting corpus. In the first, the same evaluation set as in [13] is used, here named $AMI_{eval}$, and all other meetings are used as the training set. In this division, the speakers in $AMI_{eval}$ are disjoint from the speakers in the training set. In the second configuration, 9 meetings from the ICSI corpus are collected to form an evaluation set, $ICSI_{eval}$, leaving the rest of the ICSI meetings as training data. The choice of meetings included in $ICSI_{eval}$, depicted in Table 11, is based on maximizing the number of speakers in the evaluation set which are not in the training set. This is a compromise since a completely disjoint evaluation set in terms of speakers can not be formed if the evaluation set consists of ICSI meetings.

**Table 11:** $ICSI_{eval}$

| |
|---|
| Bed017, Bmr014, Bed009, Bro017 |
| Bsr001, Buw001, Bmr003, Bro024, Bns002 |

In addition, only speech frames from either $H_{100\%}$ or $H_{\leq 65\%}$ are used, both in training and evaluation, with the targets being

$$\sigma(\mathbf{x}) = \begin{cases} 1, & \text{if } H_{\%\mathbf{x}} \leq H_{65\%} \\ 0, & \text{if } H_{\%\mathbf{x}} = H_{100\%} \end{cases}$$

This choice is based on preliminary experiments which showed that the corresponding binary classification task would be more difficult if the speech frames from both targets could have a similar $H_{\%}$. Furthermore, in training, only every second speech frame with $H_{100\%}$ is used in order to balance the target distributions and to remove redundancy caused by the large overlap duration used in the meeting corpus.

Otherwise, the training configuration is the same as depicted in Table 8 but with two differences. Firstly, the loss function is binary cross-entropy. Secondly, as a final counter measure against class imbalance, the targets of the 1-class are weighted twice as much as the 0-targets in the training. The best neural network is again chosen in a similar fashion as in the speaker modeling experiments.

### 4.3.2 Results

The results obtained in the HBS binary classification task are illustrated in Table 12. In this table, (B) refers to a neural network which is the same as the HBS neural network but without the attention layer. Consequently, (B+A) adds the attention layer to the original HBS neural network. The architecture titled (B+A+D) adds both regular and recurrent dropout [76] in the bidirectional LSTM layer. Dropout rates are set to 0.2. This value was found in preliminary experiments.

**Table 12:** mAP scores

| | $AMI_{eval}$ | $ICSI_{eval}$ |
|---|---|---|
| B | 0.867 | 0.884 |
| B + A | 0.927 | 0.902 |
| B + A + D | **0.953** | **0.935** |

Based on the results in Table 12, it is clear that both the attention layer and the use of dropout are beneficial. The intuition behind the attention layer was explained

in the previous section, and here this intuition is further confirmed. The success of dropout is also not surprising. Although the classification task includes only two classes it is still difficult since in many cases the labels themself are inaccurate. This is very natural because transcribing human conversations with a high precision is extremely hard. Many frames labeled as 1 could easily belong to the opposite class as the speaker change or overlapping speech would be hard to detect in these frames even for humans. Some of these frames were actually listened in this thesis and the remark was confirmed. The wrongly labeled frames could then lead to an overfitting which dropout tries to prevent. On top of that, overfitting could occur simply because of a relatively small training set.

Enlarging the training set artificially was also tested. In this approach, frames with $H_{\%} \leq 65\%$ were created artificially using the Librispeech corpus. In practice, the frames were generated by cutting and adding frames together with either a small overlap or none. The frames sounded fairly natural. Multiple different sized artificial data inclusions were experimented with, but none of them resulted in a performance improvement. This was a rather disappointing result in general when considering HBS. It implied that the real data would be needed for improving the HBS neural network. This real data, at least for now, is still quite difficult to acquire.

Table 12 shows that the MAP scores are higher on $AMI_{test}$ than on $ICSI_{test}$ with all the different neural network configurations. This result is as presumed since the ICSI corpus can be considered more challenging than AMI for several reasons. Firstly, the meetings in ICSI corpus have generally more participants and in many cases some of the participants speak significantly less than others. This problem was addressed briefly in the Top Two Silhouettes subsection. Furthermore, the meetings in the ICSI corpus have lower quality. The quality difference is not only based on differences in the microphone configurations but also on the training of the participants [18, 19]. In the ICSI corpus, the participants were prone to produce very loud breathing sounds near the headset microphones [19]. In the AMI corpus, the participants were advised to avoid this behavior [18].

Despite the difficulties, the best MAP scores are overall promising. Nevertheless, HBS is yet to be tested as part of the speaker diarization system. Similarly as with speaker modeling, the evaluation of HBS continues in the next subsection which illustrates the experiments with SphereDiar. From now on, all references to the HBS neural network will assume that this network uses the previously discussed dropout configuration (B+A+D).

## 4.4 SphereDiar

In the final experiments, the performance of the speaker diarization system proposed in this thesis is investigated. These investigations focus on three main aspects:

1. Experiments with SphereDiar using multiple different configurations but without using HBS.
2. Experiments with SphereDiar with the best configuration and with HBS.
3. Comparing the best SphereDiar configuration with two other state-of-the-art speaker diarization systems.

The investigations are a natural continuation to the previous subsections since they evaluate SS and HBS as a part of the speaker diarization system. The configurations will include all trained SS and SS* neural networks and the best HBS neural networks discussed in the previous subsection.

### 4.4.1 Parameters of Top Two Silhouettes

The parameters of Top Two Silhouettes and also Top Silhouette consist of a number of initializations $R$, a maximum number of speakers $N_{max}$ and a threshold $\delta$ as depicted in Algorithm 1 in subsection 3.4.3. In all experiments, $R$ is 50 and $N_{max} = 11$. $R$ is set this high since, as discussed, spherical K-means has a tendency to converge to a local maximum [67]. The value of $N_{max}$ is selected to exceed the highest possible number of participants, 9, of all the meetings in the meeting corpus. In addition, $\delta$ is assigned as 0.1. This value was attained by conducting a grid search on a clustering development set $Clust_{\text{dev}}$, consisting of 12 meetings, which are listed in Table 13. This set is disjoint from both $AMI_{test}$ and $ICSI_{test}$. In the grid search, each threshold was evaluated using DER, the use of HBS was omitted and the speaker modeling was performed with SphereSpeaker trained on $VC_{1000}$.

**Table 13:** $Clust_{dev}$

| |
|---|
| IS1003d, ES2016b, ES2010a, ES2006a, TS3012b, TS3008b, TS3011d, Bro008, Bed013, Bed016, Bmr022, Btr002 |

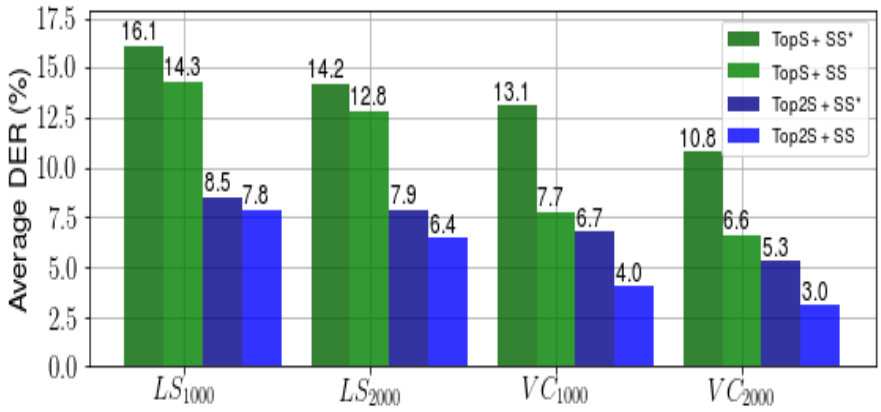


**Figure 6:** Average DER over 225 meetings from the meeting corpus with different SphereDiar configurations which omit HBS.

### 4.4.2 Results

In Figure 6, the speaker diarization results with 225 meetings from the meeting corpus are visualized. The meetings consist of all of the meetings in the meeting corpus excluding $Clust_{dev}$. These results are obtained using all possible SphereDiar configurations introduced in this thesis but without using HBS ($h_i = 0, \forall i$) as most of the meetings have been used in HBS training. The results illustrate that the SS network outperforms the SS* network, especially when these neural networks are trained with Voxceleb2 partitions. This result concurs with the Tables 9 and 10 which were provided in the previous speaker modeling experiments subsection. Moreover, the results show that both the increase in the number of training speakers and the use of Voxceleb2 partitions over Librispeech partitions are preferable in speaker modeling training. Clearly, the low adjusted Rand scores which were discussed in the speaker modeling subsection are not problematic.

Top2S performs markedly better than TopS with all configurations. Thus, the results provide some proof that their main difference, the inner cluster search, is reasonable and effective. The best configuration is attained by combining SS trained with $VC_{2000}$ and Top2S and it achieves a 3% average DER over the 225 meetings.

The results in Table 14 show that HBS fails to benefit the speaker diarization task. However, the results also show that even when using an oracle HBS, which assigns $h_i$ based on the reference HBS labels, no significant improvement for the task is attained. This result is especially clear when the evaluation set consists of all of the 225 meetings. Interestingly, these results imply that SphereDiar is not heavily dependent on overlapping speech detection or speaker change detection

**Table 14:** Average DER (%) over different evaluation sets and HBS setups with the best SphereDiar configuration.

| Segmentation | $AMI_{eval}$ | $ICSI_{eval}$ | 225 meetings |
|---|---|---|---|
| - | 2.4 | 2.9 | 3.0 |
| HBS | 3.5 | 4.8 | - |
| Optimal HBS | 2.0 | 2.5 | 2.8 |

which have been previously shown to be important factors in speaker diarization [1, 12]. This outcome might be due to two reasons: good generalization ability of the speaker embeddings and a relatively low significance of HBS for the Top2S algorithm. Especially the latter can be emphasized, since the HBS labels are only utilized to exclude some of the embeddings from the clustering procedure but not in any other manner. For example, the labels could have also been used in the initialization of the spherical K-means algorithm. Nevertheless, based on the results in Table 3, it is clear that SphereDiar achieves good results even without HBS.

**Table 15:** Average DER (%) comparison.

| Test set | Previous best | SphereDiar ($H_{\theta\%} = 55\%$) |
|---|---|---|
| $AMI_{eval}$ | 4.8 [13] | 3.6 |
| ICSI subset | 13.1 [14] | 4.5 |

In Table 15, a comparison between the best SphereDiar configuration and two other speaker diarization systems which have obtained top scores on AMI and ICSI subsets in the literature is provided. These systems include a state-of-the-art i-vector based speaker diarization system [13] and the ICSI RT07s speaker diarization system, which uses both MFCCs and deep learning based features [14, 77]. The average DER for both systems has been calculated from the segments which do not include overlapping speech and by using a forgiveness collar around speaker change boundaries [13, 14]. With [14], this collar is $\pm 0.25$ seconds, whereas [13] uses the collar of $\pm 0.5$ seconds.

The computation of DER for SphereDiar is based on using the frames which have homogeneity percentages above the threshold $H_{\theta\%} = 55\%$. Due to the formulation of the percentage, this means that virtually all overlapping speech is removed from the DER calculation. Furthermore, decreasing the $H_{\theta\%}$ from 65%, which was used previously, to 55%, can be interpreted as shrinking the collar around speaker change boundaries. This decrease allows the average DER comparison to be as fair as possible since any further decrease in the value of $H_{\theta\%}$ results in severe difficulties

of labeling the frames accurately. Consider, for instance, the example given in subsection 2.1.5. If a frame would have $H_{\%} = 50\%$, and would contain two speakers without any overlapping speech, then, the speaker label of this frame could not be determined.

In addition, [14] does not specify which meetings were included in the ICSI subset, they only inform that it consists of 55 meetings. For this reason, the ICSI subset used in this thesis consists of all 69 ICSI meetings from the 225 meeting subset.

The results illustrate that SphereDiar is able to outperform the systems in [13, 14]. In particular, the average DER of SphereDiar is better compared to [14] but it must be mentioned that SphereDiar was trained with Voxceleb2, which was not available at the time for [14]. However, the system in [13] was trained with very similar data as in this thesis, using Voxceleb [5] and other relevant datasets, but the result obtained with SphereDiar is still better. Furthermore, as HBS is not used in the comparison, the domain adaptation is only based on 12 meetings in $Clust_{dev}$. This is significantly less than used in either [13] or [14] and further emphasizes the generality of SphereDiar.

# 5 CONCLUSIONS

This thesis proposed a novel speaker diarization system named SphereDiar. This system was constructed from three main components: the SphereSpeaker neural network, the Homogeneity Based Segmentation neural network and the Top Two Silhouettes clustering algorithm. The first of these components is used in speaker modeling, whereas the last two perform segmentation and clustering, respectively. It was illustrated that these three separate tasks can be fused successfully into one speaker diarization task. SphereDiar was evaluated with over 200 meeting recordings and the average DER over these meetings was only 3%. Furthermore, the system was compared with two other state-of-the-art speaker diarization systems which both had larger average DER than SphereDiar over two meeting recording subsets. Interestingly, the conducted experiments also revealed that the use of HBS would not be crucial for the system. As a result, SphereDiar could be simplified by omitting segmentation. The diagram of this simplified system configuration is visualized in Figure 7.

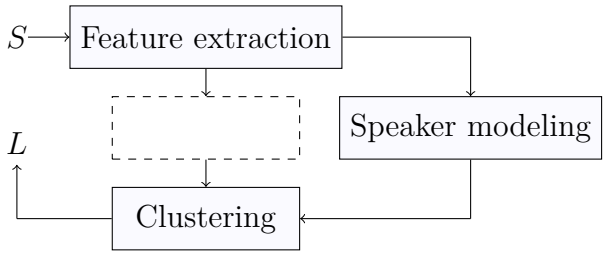


**Figure 7:** Updated block diagram of SphereDiar, which excludes segmentation (recall Figure 2 in section 3).

The remainder of this section discusses the operation, motivation, performance and future work suggestions for each individual component of the SphereDiar speaker diarization system. Finally, some recommendations for the usage of SphereDiar are given.

## 5.1 The SphereSpeaker neural network

The SS neural network was designed to transform a short utterance to a representation which describes the speaker of the utterance. The main idea behind this transformation adheres to the so called d-vector method. In this method, a neural network is devised to classify a given utterance to a speaker identity. In the process, the d-vector, i.e. a neural speaker embedding, is extracted from the last hidden layer

of the network, and this vector is used as a speaker representation. In recent years, the d-vector based methods have been proven both efficient and successful in speaker modeling.

In this thesis, the d-vector method was developed further by studying the effect of $L^2$ normalization on the speaker embedding. The experiments showed that this relatively simple operation has a significant positive impact both on the performance of the SS and the whole speaker diarization system. With this normalization, the speaker embeddings are spherical. This was the main reason for naming the neural network SphereSpeaker and, consequently, naming the speaker diarization system SphereDiar.

Furthermore, the experiments conducted in this thesis emphasized the effect of a training dataset. The SS neural network was trained both using Librispeech and Voxceleb2 datasets, of which Voxceleb2 was shown to be the better choice. This result was not surprising since Voxceleb2 had a lot more variety in both recording conditions and noise backgrounds. Moreover, Voxceleb2 has been an extremely successful training set in the speaker verification literature. In this thesis, the importance of Voxceleb2 was also verified in speaker diarization.

The SS neural network can be considered the greatest individual contribution of this thesis. The experiments revealed that the speaker embeddings extracted from this neural network are suitable for clustering even in the presence of spontaneous and overlapping speech. Yet, the network architecture of SS is still far from optimal. That is, the embedding dimension could and should be smaller and the network could use an attention mechanism and dropout. Furthermore, the network could be also trained with a significantly larger dataset. For instance, there are still around 4000 speakers more in Voxceleb2 which were not used in this thesis. In future work, all these possible improvement suggestions will be investigated.

## 5.2 Homogeneity Based Segmentation

The purpose of the HBS neural network was to determine if a 2 second length audio sample contains one dominant or multiple speakers. In this approach, overlapping speech detection and speaker change detection are performed jointly. To the best of the writer's knowledge, this thesis was the first to investigate such an approach. The experiments illustrated that the HBS neural network is successful in the classification task but not as a part of the speaker diarization system. However, the experiments also pointed out that even the use of perfect oracle HBS segmentation improved

DER only 0.2% compared to diarization without any segmentation information. This discovery was especially surprising since it differs from the general consensus in the speaker diarization literature regarding the importance of overlapping speech and speaker change detection.

The result might still be due to the fact that the exploitation of HBS was rather limited in this thesis. In practise, HBS was used to simply select speaker embeddings for the clustering procedure. It remains future work to utilize HBS also in the initialization of the clusters. However, it is important to consider also the limitations of HBS. While acquiring labeled overlapping speech data is difficult, experiments indicated that adding artificially overlapped training data does not improve the performance of HBS. Moreover, the use of HBS does add one additional component to the speaker diarization system, which increases complexity. Nevertheless, HBS could be used in other speech related applications such as ones considering privacy.

## 5.3 Top Two Silhouettes

This thesis introduced also a novel clustering algorithm Top2S. The main idea behind this algorithm was to use a two stage approach which consists of generating different clustering proposals and searching for the optimal one. The proposals were created using spherical K-means whereas the search was based on using silhouette scores and heuristic rules. The proposals with the two largest silhouette scores were central to the algorithm and the main reason for its name. The algorithm differed from the mainstream approaches in speaker diarization but achieved promising results in the conducted experiments. It outperformed a similar clustering algorithm TopS with a wide margin and also surpassed the results of two state-of-the-art speaker diarization systems.

But on the other hand, the Top2S algorithm also had deficiencies. Firstly, it was not optimal for speaker diarization tasks which include speakers speaking only in few utterances. Secondly, the clustering algorithm could not be used for online speaker diarization, and it did not exploit the time varying structure of clustered embeddings. Moreover, the proposal generation related to Top2S was not the most efficient since it also included the computation of unnecessary clusterings. Future work will focus especially on developing Top2S for online speaker diarization and for detecting briefly appearing speakers.

## 5.4 For users

When using the SphereDiar speaker diarization system, the first thing to address is that SphereDiar does not include any VAD system. This means that the VAD needs to be acquired or to be developed externally for tasks which require complete speaker diarization. Fortunately, free and user friendly VAD systems do exist, with one example being the webRTC VAD [23]. Furthermore, this thesis has not conducted any experiments in which the effect of possible errors of a non-optimal VAD would have been taken into account. As a result, the performance of SphereDiar may suffer when it is used alongside an external VAD.

Secondly, the Top2S algorithm is not well suited for speaker diarization tasks involving speakers which are vocally active only briefly (in the ballpark of less than 10 seconds in total). For further information, revise the end of subsection 3.4.3. The problem can be countered to some extent by using a large overlap in the frame extraction procedure, but this operation may not result in sufficient performance improvements.

In addition, the input format of the system should not be altered. In section 2, this format was defined as a sequence of frames with 2 second duration and sampled at 16 kHz frequency. However, it is not impossible to use a different frame duration or sampling frequency. Nevertheless, the alteration of these two is not advisable since the SS and the HBS neural networks have been trained based on the standard input format. More specifically, the input of these networks have been MFCC based feature sequences which have been extracted from the 2 second duration frames. Keeping the dimensions of the sequences the same, which would be required, and modifying the frame duration could lead to more rapid changes in the time structure of the sequences. There is no guarantees how the networks would react to this change. Moreover, the change in sampling frequency would alter the frequency bands which the MFCCs have aimed to describe in the training of the networks.

The system can be acquired from the repository[2] introduced in section 1. This repository includes a demo in which SphereDiar is used to diarize one of the meetings in the ICSI corpus. It is highly recommended to take a look at this demo before using the system. Hopefully, the repository provides a good baseline speaker diarization system which is easy to use.

[2] `https://github.com/Livefull/SphereDiar`

# REFERENCES

[1] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356–370, 2012.

[2] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal processing magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[3] X. Anguera Miró, *Robust speaker diarization for meetings*. Universitat Politècnica de Catalunya, 2006.

[4] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pp. 5206–5210, IEEE, 2015.

[5] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *Interspeech*, 2017.

[6] N. Tomashenko and Y. Estève, "Evaluation of feature-space speaker adaptation for end-to-end acoustic models," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018.

[7] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, Y. Wu, *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in Neural Information Processing Systems*, pp. 4480–4490, 2018.

[8] D. A. Reynolds, "An overview of automatic speaker recognition technology," in *ICASSP*, vol. 4, pp. 4072–4075, 2002.

[9] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Interspeech*, 2018.

[10] M. Hajibabaei and D. Dai, "Unified hypersphere embedding for speaker recognition," *arXiv preprint arXiv:1807.08312*, 2018.

[11] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, "Clustering on the unit hypersphere using von Mises-Fisher distributions," *Journal of Machine Learning Research*, vol. 6, no. Sep, pp. 1345–1382, 2005.

[12] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, *et al.*, "Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge," in *Proc. Interspeech*, pp. 2808–2812, 2018.

[13] M. Maciejewski, D. Snyder, V. Manohar, N. Dehak, and S. Khudanpur, "Characterizing performance of speaker diarization systems on far-field speech using standard methods," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5244–5248, IEEE, 2018.

[14] S. H. Yella and A. Stolcke, "A comparison of neural network feature transforms for speaker diarization," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[15] K. Boakye, B. Trueba-Hornero, O. Vinyals, and G. Friedland, "Overlapped speech detection for improved speaker diarization in multiparty meetings," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 4353–4356, IEEE, 2008.

[16] S. H. Yella and H. Bourlard, "Overlapping speech detection using long-term conversational features for speaker diarization in meeting room conversations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1688–1700, 2014.

[17] J. T. Geiger, F. Eyben, B. Schuller, and G. Rigoll, "Detecting overlapping speech with long short-term memory recurrent neural networks," in *Proceedings Interspeech 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France*, 2013.

[18] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, *et al.*, "The AMI meeting corpus," in *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, vol. 88, p. 100, 2005.

[19] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, *et al.*, "The ICSI meeting corpus," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 1, pp. I–I, IEEE, 2003.

[20] S. H. Yella, A. Stolcke, and M. Slaney, "Artificial neural network features for speaker diarization," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pp. 402–406, IEEE, 2014.

[21] H. Bredin, "Tristounet: triplet loss for speaker turn embedding," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5430–5434, IEEE, 2017.

[22] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4930–4934, IEEE, 2017.

[23] A. Johnston, J. Yoakum, and K. Singh, "Taking on WebRTC in an enterprise," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 48–54, 2013.

[24] M. Portnoff, "Short-time Fourier analysis of sampled speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 3, pp. 364–373, 1981.

[25] B. Logan *et al.*, "Mel frequency cepstral coefficients for music modeling.," in *ISMIR*, vol. 270, pp. 1–11, 2000.

[26] M. Hrúz and Z. Zajíc, "Convolutional neural network for speaker change detection in telephone speaker diarization system," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pp. 4945–4949, IEEE, 2017.

[27] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with SincNet," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 1021–1028, IEEE, 2018.

[28] G. Wisniewksi, H. Bredin, G. Gelly, and C. Barras, "Combining speaker turn embedding and incremental structure prediction for low-latency speaker diarization," in *Proc. Interspeech*, 2017.

[29] S. Meignier and T. Merlin, "LIUM SpkDiarization: an open source toolkit for diarization," in *CMU SPUD Workshop*, 2010.

[30] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[31] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker and session variability in GMM-based speaker verification," *IEEE Transactions on Audio Speech and Language Processing*, vol. 15, no. 4, p. 1448, 2007.

[32] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[33] H. Beigi, *Fundamentals of speaker recognition*. Springer Science & Business Media, 2011.

[34] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 308–311, 2006.

[35] D. A. Reynolds, R. C. Rose, *et al.*, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE transactions on speech and audio processing*, vol. 3, no. 1, pp. 72–83, 1995.

[36] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 980–988, 2008.

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

[38] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, "PLDA for speaker verification with utterances of arbitrary duration," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 7649–7653, IEEE, 2013.

[39] E. Variani, X. Lei, E. McDermott, I. Lopez-Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification.," in *ICASSP*, vol. 14, pp. 4052–4056, Citeseer, 2014.

[40] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, pp. 999–1003, 2017.

[41] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333, IEEE, 2018.

[42] S. Horiguchi, D. Ikami, and K. Aizawa, "Significance of softmax-based features in comparison to distance metric learning-based features," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[43] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[44] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*, pp. 165–170, IEEE, 2016.

[45] O. Rippel, M. Paluri, P. Dollar, and L. Bourdev, "Metric learning with adaptive density discrimination," *ICLR*, 2016.

[46] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[47] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, p. 1, 2017.

[48] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[49] G. Gelly and J. Gauvain, "Spoken language identification using LSTM-based angular proximity," in *Proc. Interspeech*, 2017.

[50] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[51] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *ICML*, 2015.

[52] C. Laurent, G. Pereyra, P. Brakel, Y. Zhang, and Y. Bengio, "Batch normalized recurrent neural networks," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2657–2661, IEEE, 2016.

[53] R. Yin, H. Bredin, and C. Barras, "Speaker change detection in broadcast tv using bidirectional long short-term memory networks," in *Interspeech 2017*, ISCA, 2017.

[54] S. Chen, P. Gopalakrishnan, *et al.*, "Speaker, environment and channel change detection and clustering via the Bayesian information criterion," in *Proc. darpa broadcast news transcription and understanding workshop*, vol. 8, pp. 127–132, Virginia, USA, 1998.

[55] P. Delacourt and C. J. Wellekens, "Distbic: A speaker-based segmentation for audio data indexing," *Speech communication*, vol. 32, no. 1-2, pp. 111–126, 2000.

[56] C. Barras, X. Zhu, S. Meignier, and J.-L. Gauvain, "Multistage speaker diarization of broadcast news," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1505–1512, 2006.

[57] R. Wang, M. Gu, L. Li, M. Xu, and T. F. Zheng, "Speaker segmentation using deep speaker vectors for fast speaker change scenarios," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pp. 5420–5424, IEEE, 2017.

[58] K. Boakye, O. Vinyals, and G. Friedland, "Two's a crowd: Improving speaker diarization by automatically identifying and excluding overlapped speech," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[59] G. Hagerer, V. Pandit, F. Eyben, and B. Schuller, "Enhancing LSTM RNN-based speech overlap detection by artificially mixed data," in *Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio*, Audio Engineering Society, 2017.

[60] P. Rémy, "Keras Attention Mechanism." `https://github.com/philipperemy/keras-attention-mechanism`, 2017.

[61] N. Evans, S. Bozonnet, D. Wang, C. Fredouille, and R. Troncy, "A comparative study of bottom-up and top-down approaches to speaker diarization," *IEEE*

*Transactions on Audio, speech, and language processing*, vol. 20, no. 2, pp. 382–392, 2012.

[62] G. Sell and D. Garcia-Romero, "Speaker diarization with PLDA i-vector scoring and unsupervised calibration," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, pp. 413–417, IEEE, 2014.

[63] E. Khoury, L. El Shafey, M. Ferras, and S. Marcel, "Hierarchical speaker clustering methods for the NIST i-vector challenge," in *Odyssey: The Speaker and Language Recognition Workshop*, pp. 254–259, Citeseer, 2014.

[64] M. Rouvier and S. Meignier, "A global optimization framework for speaker diarization," in *Odyssey 2012*, 2012.

[65] H. Dubey, A. Sangwan, and J. Hansen, "Robust speaker clustering using mixtures of von Mises-Fisher distributions for naturalistic audio streams," in *Interspeech*, pp. 3603–3607, 2018.

[66] S. Shum, N. Dehak, E. Chuangsuwanich, D. Reynolds, and J. Glass, "Exploiting intra-conversation variability for speaker diarization," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[67] S. Zhong, "Efficient online spherical K-means clustering," in *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, vol. 5, pp. 3180–3185, IEEE, 2005.

[68] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, Society for Industrial and Applied Mathematics, 2007.

[69] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[70] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[71] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[72] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240, ACM, 2006.

[73] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve.," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[74] O. Galibert, "Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech.," in *Interspeech*, pp. 1131–1134, 2013.

[75] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2015.

[76] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in neural information processing systems*, pp. 1019–1027, 2016.

[77] C. Wooters and M. Huijbregts, "The ICSI RT07s speaker diarization system," in *Multimodal Technologies for Perception of Humans*, pp. 509–519, Springer, 2007.