FACULTY OF APPLIED SCIENCES
UNIVERSITY
OF WEST BOHEMIA

University of West Bohemia
Faculty of Applied Sciences
Department of Cybernetics

Speaker Diarization

DISSERTATION THESIS

submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the field of **Cybernetics**

Ing. Marie Kunešová

Advisor: Doc. Dr. Ing. Vlasta Radová
Plzeň, 2021

FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra Kybernetiky

Diarizace řečníků

DISERTAČNÍ PRÁCE

k získání akademického titulu doktor
v oboru **Kybernetika**

Ing. Marie Kunešová

Školitel: Doc. Dr. Ing. Vlasta Radová
Plzeň, 2021

Acknowledgments

I would like to thank my advisor Doc. Dr. Ing. Vlasta Radová for her guidance during my Ph.D. study.

I would also like to thank my colleagues, both from our university and elsewhere, with whom I've had many helpful discussions. I am particularly thankful to Ing. Zbyněk Zajíc Ph.D., Ing. Marek Hrúz Ph.D., Ing. Jan Vaněk Ph.D. and Ing. Pavel Campr Ph.D., for collaboration on various projects and papers and for all the useful advice they offered me during different stages of my study.

Finally, I am grateful to my family for supporting me during these long years.

Abstract

The thesis focuses on the topic of speaker diarization, a speech processing task that is commonly characterized as the question “Who speaks when?”. It also addresses the related task of overlapping speech detection, which is very relevant for diarization.

The theoretical part of the thesis provides an overview of existing diarization approaches, both offline and online, and discusses some of the problematic areas which were identified in early stages of the author’s research. The thesis also includes an extensive comparison of existing diarization systems, with focus on their reported performance. One chapter is also dedicated to the topic of overlapping speech and the methods of its detection.

The experimental part of the thesis then presents the work which has been done on speaker diarization, which was focused mostly on a GMM-based online diarization system and an i-vector based system with both offline and online variants. The final section also details a newly proposed approach for detecting overlapping speech using a convolutional neural network.

Keywords

speaker diarization, overlapping speech detection, speech processing

Abstrakt

Disertační práce se zaměřuje na téma diarizace řečníků, což je úloha zpracování řeči typicky charakterizovaná otázkou „Kdo kdy mluví?“. Práce se také zabývá související úlohou detekce překrývající se řeči, která je velmi relevantní pro diarizaci.

Teoretická část práce poskytuje přehled existujících metod diarizace řečníků, a to jak těch offline, tak online, a přibližuje několik problematických oblastí, které byly identifikovány v rané fázi autorčina výzkumu. V práci je také předloženo rozsáhlé srovnání existujících systémů se zaměřením na jejich uváděné výsledky. Jedna kapitola se také zaměřuje na téma překrývající se řeči a na metody její detekce.

Experimentální část práce předkládá praktické výstupy, kterých bylo dosaženo. Experimenty s diarizací se zaměřovaly zejména na online systém založený na GMM a na i-vektorový systém, který měl offline i online varianty. Závěrečná sekce experimentů také přibližuje nově navrženou metodu pro detekci překrývající se řeči, která je založena na konvoluční neuronové síti.

Klíčová slova

diarizace řečníků, detekce překrývající se řeči, zpracování mluvené řeči

Declaration

I hereby declare that this thesis is my own work and to my best knowledge it does not contain any previously published materials except for the ones acknowledged in the text.

.....
Ing. Marie Kunešová

Contents

List of Figures

List of Tables

List of Abbreviations

AHC	Agglomerative hierarchical clustering
ASR	Automatic speech recognition
BiLSTM	Bi-directional long short-term memory network
BIC	Bayesian Information Criterion
CLR	Cross-Likelihood Ratio
CMN	Cepstral Mean Normalization
CNN	Convolutional Neural Network
DER	Diarization Error Rate
DNN	Deep Neural Network
EER	Equal Error Rate
EM	Expectation-Maximization (algorithm)
FA	Factor Analysis
FFT	Fast Fourier Transform
GLR	Generalized Likelihood Ratio
GMM	Gaussian Mixture Model
HCI	Human-computer interaction
HMM	Hidden Markov Model
HRI	Human-robot interaction
IB	Information Bottleneck
ILP	Integer Linear Programming
JER	Jaccard Error Rate
JFA	Joint Factor Analysis
KL	Kullback-Leibler divergence
KL2	Symmetric Kullback-Leibler divergence
LDA	Linear discriminant analysis
LFCC	Linear Frequency Cepstral Coefficients
LPC	Linear Predictive Coding
LSTM	Long Short-Term Memory (neural network)
MAP	Maximum a posteriori probability
MI	Mutual Information
MDM	Multiple distant microphones
MFCC	Mel Frequency Cepstral Coefficients

NCLR	Normalized Cross-Likelihood Ratio
NIST	National Institute of Standards and Technology
ODE	Overlap Detection Error
PCA	Principal Component Analysis
PLDA	Probabilistic linear discriminant analysis
PLP	Perceptual Linear Prediction
RNN	Recurrent Neural Network
SAD	Speech activity detection, also known as voice activity detection (VAD)
SCD	Speaker change detection
SER	Speaker error (one of the components of DER), also "(speaker) confusion error"
SDM	Single distant microphone
SVM	Support vector machines
TDNN	Time Delay Neural Network
TDOA	Time difference of arrival (also "time delay of arrival")
UBM	Universal background model
VAD	Voice activity detection, also known as speech activity detection (SAD)
VB	Variational Bayes
VB-HMM	(Variational) Bayesian Hidden Markov Model
WCCN	Within-class covariance normalization

Chapter 1

Introduction

In the current times, an increasing amount of audio data is being recorded and stored. This leads to a need for automated methods which can process the large volume of data and extract relevant information, sparking increased interest in various areas of automated speech processing. Among the main topics are speech recognition, speaker recognition and also speaker diarization.

Speaker diarization is the task of determining "Who speaks when?" within a recorded conversation of several speakers. In contrast to the related task of speaker recognition, speaker diarization does not aim at identifying the actual identities of the speakers and is typically performed without any prior knowledge about the number of speakers or their identities.

In other words, a diarization system is presented with an audio recording which contains the speech of several unknown speakers. The goal of the system is then to find the intervals of speech within the recording, divide them into speaker homogeneous segments and label these segments such that the intervals of speech of the same speaker are assigned an identical label. This task is complicated by the fact that some of the speakers may be talking simultaneously. An example of such labeling is illustrated in Figure 1.1.

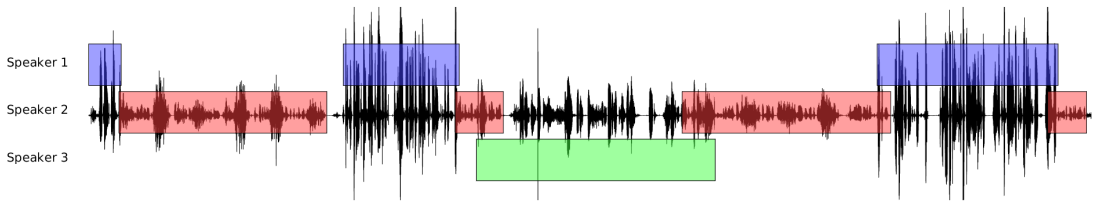

Figure 1.1: An illustration of the final result of speaker diarization, with three speakers who sometimes talk simultaneously.

The main use for speaker diarization is in situations where it is necessary to annotate the speech signal based on the individual speakers without the need to know their identities. The most common examples are audio indexing and rich transcription. Other applications may include voice assistants and similar forms of human-computer interaction (HCI) with multiple simultaneous users (such as shown by Minotto et al., 2015) and human-robot interaction (HRI).

Additionally, detecting speaker changes can be beneficial in improving the performance of speech recognition systems, by allowing such system to switch between different acoustic models based on the current speaker or to reset language models when a change is detected.

Diarization systems generally fall into one of two distinct categories: *offline* and *online*, depending on the manner in which they operate. An offline system processes a complete audio recording after it has already been fully recorded, allowing the system to base all of its decisions on the entirety of the data.

By contrast, online systems are used to process an incoming audio stream sequentially, while it is being received. All decisions are based only on the data seen up to the current point. This allows such systems to be used for real-time diarization of live data, at the cost of lower performance, as less information is available to the system at the time of decision.

Diarization tasks have traditionally been divided into three distinct domains with different characteristics, based on the type of the audio recording they are intended for. These groups are *telephone speech*, *broadcast news* and *conference meetings* (Tranter and Reynolds, 2006). These domains differ in several characteristics such as the typical number of speakers, length of speaker turns, background speech, etc., leading to different advantages and challenges.

In *telephone speech* applications, the number of different speakers is typically very small. In many circumstances, it can be expected that only two speakers are present in a recorded call. Such knowledge allows for simpler approaches and for this reason, some of the systems found in literature work under this assumption (e.g. Sell and Garcia-Romero, 2014).

A less convenient property of telephone speech arises from the spontaneous nature of most conversations. One can generally expect very short speaker turns and frequent instances of overlapping speech, both of which pose additional challenge to the diarization process. In addition, the audio quality of telephone recordings is generally poorer than that of the other two audio types and background noise may be present.

Broadcast news, by contrast, tend to involve a larger, unknown number of speakers with generally longer speaker turns and significantly less overlapping speech. However, this type of audio may also contain different types of noise and non-speech sound sources, such as music, which often have to be detected as well, in order not to influence the diarization process. Additionally, it is also necessary take into account that a single speaker may be encountered in more than one situation or environment – for instance, they may sometimes speak from the studio and sometimes during a report in the field, such as on a busy street. This will influence the acoustic characteristics of the speech signal in different ways.

A defining characteristic of *conference meetings* diarization is the frequent utilization of multiple sound sources. While both telephone speech and broadcast news typically involve speech on a single channel, conference meetings often employ multiple microphones, which are either worn by the participants, or placed on various points around the meeting room. This allows for the additional speech sources to be used to enhance the acoustic signal or to obtain spatial information about the speakers (e.g. Pardo et al., 2006).

While speaker diarization is mainly used for the processing of audio information, there has also been significant effort put into the research of multimodal diarization techniques, which combine audio and visual information in order to

improve the performance of diarization systems in situations where the visual information is available (e.g. Noulas et al., 2012). However, this thesis will focus primarily on the audio domain. Multimodal approaches will only be briefly acknowledged in the interest of completeness.

In the time since the work on this thesis was started, many things have rapidly changed in the field of speech processing. In the early 2010s, the most common approaches involved traditional and time-tested concepts like MFCC features and GMMs (and later the then-new i-vectors), and the majority of speaker diarization systems followed a framework of several clearly separated steps, each with their own specialized algorithms. Yet continued interest in deep learning has meant that over the past few years, more and more of this process is being substituted by neural networks, with some of the very newest cutting-edge research now focusing on end-to-end systems which aim to encompass the entire process in a single neural architecture.

In order to maintain continuity in research, most of the contents of this thesis still focus on the classic approaches. However, references to newer methods have been added where appropriate.

1.1 Outline and Aims of the Thesis

The structure of this thesis reflects the initial aims which were established based on the preceding thesis report (Kunešová, 2017):

- Survey existing speaker diarization methods – both offline and online
- Identify some of the main challenges and obstacles in speaker diarization
- Create an overview of previous results reported in literature, comparing the performance of individual systems
- Implement some of the described methods in a new diarization system and propose new methods or improvements
- Address one or more of the previously identified challenges

These points mainly served as the initial framework of the research. The specific goals and objectives of the thesis will be presented and discussed in more detail in chapter 6, after the theoretical part of the thesis which provides the necessary background.

The thesis is divided into 10 chapters. The first third of the text presents the theoretical background, basic approaches and challenges of speaker diarization. This is followed by an extensive overview of recent state-of-the-art results in literature, and a large penultimate chapter which details the newly proposed techniques, experiments and results.

Chapter 2 introduces certain theoretical concepts which are referred to multiple times throughout the later chapters.

Chapter 3 focuses on offline diarization. It presents the typical framework of offline systems and describes the most common methods used for each of the individual subtasks. Multimodal diarization is also briefly touched upon in this chapter.

In chapter 4, this is followed by a presentation of online diarization approaches, of which many stem from offline diarization or share some of the same methods. This chapter includes the description of the additional challenges and limitations presented in the task of online diarization.

Chapter 5 introduces several important obstacles frequently encountered in speaker diarization – including overlapping speech, which is more closely addressed in chapter 8.

The specific goals of the thesis are presented in chapter 6.

Chapter 7 introduces the methods used for the evaluation of diarization systems and presents a comparison of many recent state-of-the-art systems and their reported results.

Chapter 8 focuses on overlapping speech as one of the aforementioned important obstacles in speaker diarization and provides an overview of existing research into overlap detection.

Chapter 9 then contains the experimental part of this thesis. It presents the work which has been done on speaker diarization, which was focused mostly on a GMM-based online diarization system and an i-vector based system with both offline and online variants. The final section also details a newly proposed approach for detecting overlapping speech.

Finally, chapter 10 summarizes the thesis and its contributions.

Some of the text in this thesis has previously appeared in the preceding thesis report (Kunešová, 2017). In the experimental part of the thesis, certain passages and tables have also been adapted from the author's previous publications, generally indicated at the start of the relevant sections.

Chapter 2

Theoretical Background

Before we can focus on the topic of speaker diarization itself, it is necessary to introduce certain important concepts which play a significant role in this process and which will be referred to multiple times throughout the subsequent chapters.

Section 2.1 presents a brief overview of the standard ways for representing speaker information. Namely, the traditional GMMs, the more recent and still widely popular i-vectors, and the newly emerging DNN-based approaches.

Section 2.2 then contains an overview of several commonly used distance metrics, which serve for the comparison of sound sources in different intervals of speech as well as between the representations of individual speakers.

2.1 Representation of Speakers

As speaker diarization involves differentiating between individual speakers in an audio stream and finding parts of speech spoken by the same speaker, it is important for a system to be able to represent and store relevant speaker-dependent information in some way.

Over the last decade, the state of the art has gradually shifted from the use of Gaussian Mixture Models (GMMs) to i-vectors and, more recently, to approaches based on Deep Neural Networks.

2.1.1 GMM-based Speaker Representation

Until relatively recently, the traditional approach was to use *GMMs* to represent individual speakers. While their use has declined over the past years, GMMs have long been popular in both speaker diarization and speaker recognition contexts due to their robustness and ability to estimate the underlying distribution from given data. A description of their use for speaker representation can be found in e.g. (Reynolds, 1995).

Some diarization approaches also employ *Hidden Markov Models* (*HMMs*), modeling the transitions between speakers as a Markov process. Typically, each speaker is represented by a single HMM state and GMM-based models are utilized for this purpose, such as in (Fredouille and Evans, 2008).

Besides the individual speaker models, many GMM-based diarization systems also employ a so-called *universal background model* (*UBM*), which is a model

trained to represent the voice of a generic speaker. A UBM is generally obtained by training a model on a large amount of data from a wide array of different speakers, so as to suppress any speaker-specific characteristics. A UBM can then serve as the basis from which the models of specific speakers are adapted.

One more important related concept is the *supervector*. In the context of speech processing, this term refers to a high-dimensional vector, obtained by concatenating the statistics of a GMM – typically the mean vectors of individual GMM components. Supervectors are of a fixed dimension, and thus can serve as a simple representation of speech segments of variable length for use in classification (e.g. in W. M. Campbell et al., 2006).

Supervectors and UBM are also used as the basis for obtaining i-vectors, a more recent alternative to GMM-based models which became widely popular for both speaker diarization and speaker recognition for a time. Because of their widespread use, i-vectors will be examined in more detail in the next section.

Finally, the less commonly used *binary key* modeling technique, proposed by Anguera and Bonastre (2010) and more recently used by e.g. Patino et al. (2018a), shows some similarities to i-vectors, in that it also uses a GMM-UBM approach to obtain a vector representation of speaker information. However, the approaches differ in the statistical modeling and in the case of binary keys, the vectors consist of only binary values.

2.1.2 i-Vectors

For a long time, GMMs were the most common models used in speaker diarization. However, they were eventually replaced in this role by *i-vectors*.

First introduced by Dehak et al. (2011) for speaker verification, i-vector representation of individual speakers' utterances has since been successfully applied to speaker recognition tasks (e.g. Garcia-Romero and Espy-Wilson, 2011; Machlica, 2012) and has now seen wide-spread use for speaker diarization as well.

Although i-vectors are arguably better suited for the extraction of speaker information from longer utterances (preferably tens of seconds of speech, as observed by e.g. Hasan et al., 2013), they can also be used for representing shorter speech segments in speaker diarization.

The theory behind i-vectors stems from factor analysis and the Joint Factor Analysis (JFA) approach (Kenny, 2005). By defining a new low-dimensional space called the *total variability space*, the i-vector approach aims at representing each speaker's utterance by a single vector of a fixed length.

Following is a brief summary of the i-vector extraction process. A more detailed description can be found in the original paper by Dehak et al. (2011).

The first step of the process consists of extracting a supervector (see previous section) from a speaker's utterance. This is done with the use of a large GMM-based UBM:

The general approach consists of performing maximum a posteriori probability (MAP) adaptation to obtain a new GMM for the utterance, based on the UBM.

The concatenated mean vectors of the adapted GMM then form the supervector. This process is illustrated in Figure 2.1.

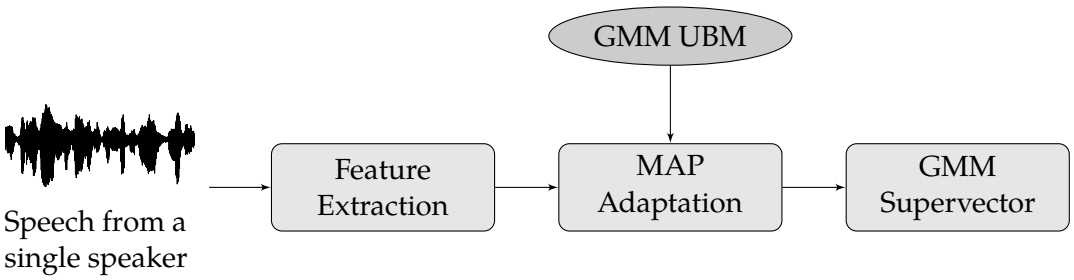

Figure 2.1: Supervector extraction process.

The i-vector extraction itself is then based on the decomposition of the speaker- and channel-dependent supervector M via factor analysis into the following components:

$$M = m + Tw \,, \tag{2.1}$$

where m is the speaker-independent mean supervector of the UBM, T is a low-rank rectangular matrix called the *total variability matrix*, which defines the total variability space, and w is a vector with standard Gaussian distribution $w \sim N(0, I)$, referred to as an i-vector.

The resulting i-vector w has a lower dimension than the original supervector or a GMM, while containing most of the important information. In some situations, the dimension can be decreased even further with the use of methods such as the Principal Component Analysis (PCA) (e.g. Sell and Garcia-Romero, 2014).

Because the MAP adaptation process does not adapt only the speaker-dependent characteristics of the speech, but also information about the channel and background noise, all of these factors are also present within the resulting i-vector. To resolve this, one may additionally perform channel compensation using approaches such as linear discriminant analysis (LDA) and within-class covariance normalization (WCCN) (Dehak et al., 2011). Alternatively, the use of PLDA as a distance metric (see section 2.2.2) can also serve a similar purpose.

Though i-vectors remain an important concept, they are now being replaced in turn by x-vectors and other DNN-based approaches.

2.1.3 DNN-based Speaker Embeddings, x-vectors

Following the success of Deep Neural Networks (DNNs) in many areas of machine learning, people have naturally started investigating their potential for extracting speaker-specific information. Initially, this mainly focused on obtaining short-term bottleneck features (see section 3.3.2), but over the last few years, there have been several proposed methods which aim to replace i-vectors as the state-of-the-art speaker representation.

The simplest approach to obtaining such embeddings is very similar to the aforementioned bottleneck features – training a DNN to discriminate between

Table 2.1: DNN architecture in the Kaldi implementation of x-vectors, adapted from (Snyder et al., 2018)

Layer	Layer context	Total context	Input x output
frame1	$[t-2, t+2]$	5	120x512
frame2	$t-2, t, t+2$	9	1536x512
frame3	$t-3, t, t+3$	15	1536x512
frame4	t	15	512x512
frame5	t	15	512x1500
stats pooling	$[0, T)$	T	1500Tx3000
segment6	0	T	3000x512
segment7	0	T	512x512
softmax	0	T	512xN

speakers and then removing the last one or more layers. The output of the remaining layers then can be used to obtain vector representation.

The most notable example of this is x-vectors (Garcia-Romero et al., 2017; Snyder et al., 2018), which are extracted using a feed-forward neural network trained to classify speakers. They are briefly described in the subsection below.

Other notable examples include the LSTM-based d-vectors (Wan et al., 2018; Wang et al., 2018) and the work of Song et al. (2018), which combines DNN embeddings with triplet loss metric learning.

x-vectors

x-vectors (Garcia-Romero et al., 2017; Snyder et al., 2018) are DNN-based speaker embeddings which were recently proposed as a direct replacement for i-vectors for both speaker recognition and speaker diarization. They have been shown to achieve better results, particularly when dealing with short speech segments (Snyder et al., 2017). In the few years since their original introduction, x-vectors have become the de facto successors to i-vectors as the most popular state of the art approach.

The x-vector extraction process uses a feed-forward DNN trained for speaker classification in speech segments of variable length, and in the original version, the input is given in the form of a sequence of MFCC feature vectors.

The network consists of three blocks of layers: a frame-level time-delay (TDNN) architecture (Peddinti et al., 2015), followed by a temporal-pooling layer which obtains long-term speaker characteristics, and a final set of segment-level layers from which the speaker embeddings can be extracted. The specific number and size of the layers can vary between implementations. Table 2.1 shows the architecture proposed in (Snyder et al., 2018).

A more detailed description of x-vectors can be found in (Snyder et al., 2017).

An open-source implementation of x-vector extraction (using the architecture

described in Table 2.1) is available in the Kaldi[1] speech recognition toolkit. This was also used in the diarization system described in section 9.4 of this thesis.

2.2 Distance Calculation

Among the common diarization approaches which will be introduced in later chapters, many involve the comparison of a pair of speech segments or their representations, in order to decide whether the two segments contain the speech of the same speaker. Additionally, some other methods similarly compare pairs of speaker models, most commonly GMMs, which potentially represent the same speaker, and all of their associated data.

In both cases, the usual approach is to calculate the distance between the pair of speech segments or models using one of many distance measures which are suitable for the purpose. This section provides an overview of some of the most popular metrics which can be used with the different forms of speaker representation which were discussed in section 2.1: section 2.2.1 shows metrics which have been historically used in GMM-based systems, while section 2.2.2 presents several options which are more suitable for use with i-vectors or various other vector-based speaker embeddings.

2.2.1 GMM-based Distance Metrics

The metrics listed here are mostly used with single Gaussian or GMM representations of the data. These metrics follow one of two basic principles: they are either *statistics-based* distances, which compare the statistics of two sets of data, and *likelihood-based*, which evaluate the likelihood of the data according to models representing it (Anguera, 2007). As GMMs are being replaced by more modern approaches, these metrics are becoming obsolete. However, they are still relevant for some of the earlier work in this thesis.

In all subsequent equations, X_i and X_j will refer to two speech segments consisting of N_i and N_j feature vectors, respectively. $X = X_i \cup X_j$ is the segment of length $N = N_i + N_j$, obtained by joining X_i and X_j. $L(X, M)$ denotes the likelihood of X given model M.

Bayesian Information Criterion (BIC)

For a long time, the arguably most wide-spread distance metric in speaker diarization was based on the calculation of the Bayesian Information Criterion (BIC).

BIC is a likelihood-based model selection criterion, penalized by model complexity. The value of the criterion indicates how well a specific model fits a given set of data. In order to prevent over-fitting, there is an added penalty term which is dependent on the number of free parameters in the model.

[1] https://kaldi-asr.org/

For speaker diarization, BIC is usually defined as

$$\mathrm{BIC}(M) = \log L(X, M) - \lambda \frac{1}{2} \#(M) \log N \,, \tag{2.2}$$

where $\#(M)$ is the number of parameters of model M and λ is a data-dependent penalty weight.

When comparing two speech segments X_i and X_j in speaker diarization, we usually calculate the *difference* Δ BIC in total BIC value, which would result from representing $X = X_i \cup X_j$ with two different models, as opposed to a single one. If Δ BIC is less than 0, single model representation is preferable (Chen and Gopalakrishnan, 1998).[2]

BIC has historically been very widely used in speaker diarization systems, due to its relative simplicity and good performance. However, it was shown by Chen and Gopalakrishnan (1998) that in the speaker change detection task (see section 3.5.1), BIC has problems with the detection of very short speaker turns, especially those under 2 seconds of length. As such, it may not be as suitable for applications where short speaker turns occur frequently, such as in spontaneous telephone conversations.

BIC was not used for any of the experimental work in this thesis. However, it is frequently referenced in the overview of existing diarization systems in chapter 7, as many of them have used BIC in the past, most commonly for segmentation.

Generalized Likelihood Ratio (GLR)

Generalized Likelihood Ratio (GLR) is another likelihood-based metric. Similarly to ΔBIC, its purpose is to express whether a given pair of speech segments is better represented by a single model or two different ones. This is achieved by computing the ratio between two hypotheses: Hypothesis H_0 says that both segments X_i and X_j contain the speech of the same speaker and as such, a single model M represents the data best. Conversely, hypothesis H_1 says that the segments belong to different speakers and are best represented by two different models, M_i and M_j.

GLR is then defined as

$$\mathrm{GLR}(X_i, X_j) = \frac{H_0}{H_1} = \frac{L(X_i \cup X_j | M)}{L(X_i | M_i) \cdot L(X_j | M_j)} \,. \tag{2.3}$$

In the above expression, high values of $\mathrm{GLR}(X_i, X_j)$ indicate the similarity of the two speech segments. In order to obtain a distance, the negative logarithm of the GLR is typically used:

$$d(X_i, X_j) = -\log \mathrm{GLR}(X_i, X_j) \,. \tag{2.4}$$

In this thesis, GLR is used mainly in the segmentation experiments in section 9.3.3.

[2] There is some inconsistency in the definition of ΔBIC in literature. Some authors, such as Delacourt and Wellekens (2000), swap the order of comparison of the two alternatives. Values less than 0 then indicate different speakers.

Kullback-Leibler Divergence (KL, KL2)

Kullback-Leibler (KL) divergence, also called *relative entropy*, is a statistics-based distance measure used to calculate the difference between two probability distributions.

For two random variables Y_i and Y_j with distributions P_i and P_j, KL divergence is defined as

$$\mathrm{KL}(Y_i, Y_j) = E_i(\log P_i - \log P_j) \ , \tag{2.5}$$

where $E_i()$ signifies expectation computed with respect to P_i (Siegler et al., 1997).

In the context of speaker diarization, P_i and P_j can be understood as the underlying distributions of the feature vectors belonging to speech segments X_i and X_j, respectively.

Since KL divergence is asymmetrical, a symmetrical variant known as KL2 may be used, defined by Siegler et al. (1997) as

$$\mathrm{KL2}(Y_i, Y_j) = \mathrm{KL}(Y_i, Y_j) + \mathrm{KL}(Y_j, Y_i) \ . \tag{2.6}$$

Kullback-Leibler divergence does not appear in the experimental part of this thesis, but it is referenced in the overview of literature.

Cross-Likelihood Ratio (CLR)

Cross-Likelihood Ratio (CLR) is a distance metric most commonly used to express the similarity between two different speaker models which have been obtained by adapting the same UBM using different sets of data. As the name implies, it is a likelihood-based metric.

It is defined as (Reynolds et al., 1998)

$$\mathrm{CLR}(M_i, M_j) = \frac{1}{N_i} \cdot \log \frac{L(X_i|M_{\mathrm{UBM}})}{L(X_i|M_j)} + \frac{1}{N_j} \cdot \log \frac{L(X_j|M_{\mathrm{UBM}})}{L(X_j|M_i)} \ . \tag{2.7}$$

Here, M_i and M_j are a pair of speaker models, which were both obtained by adapting the same UBM, M_{UBM}, with data X_i and X_j, respectively.

If we consider only the expression

$$\frac{L(X_i|M_{\mathrm{UBM}})}{L(X_i|M_j)} \ , \tag{2.8}$$

we are comparing the likelihood of the data X_i given the model M_j with the likelihood of the same data given the original UBM and the following can be observed:

If data X_i and X_j corresponds to the same speaker, the adapted model M_j will also represent the other set of data, X_i, better than the non-adapted UBM. Therefore, the resulting value of this expression will be low.

On the other hand, a high value of this expression would indicate that X_i is better represented by the original UBM than the adapted model M_j, indicating that there are two different speakers.

Note: The weak point of this method is the UBM. In order to achieve reliable results, the UBM needs to represent all speakers in the recording sufficiently well. If the UBM fits the data X_i very badly, it is possible that *any* adapted model M_j will always be better than the UBM, regardless of whether the speakers match. This would result in a misleadingly low value of the CLR. This is especially likely to occur if we attempt to use a diarization system with significantly different data than it was developed for, such as applying a UBM trained purely on broadcast news data to a telephone speech task.

In this thesis, CLR was used in the GMM-based online diarization system in section 9.2.1. It or the related NCLR metric (see below) were also used by several of the systems listed in the literature overview.

Normalized Cross-Likelihood Ratio (NCLR)

Normalized Cross-Likelihood Ratio (NCLR) is a likelihood-based metric closely related to the previously described CLR. It was first used by Reynolds (1995) for speaker verification.

The only difference between CLR and NCLR is that compared to Equation 2.7, $L(X_i|M_{\text{UBM}})$ is replaced by $L(X_i|M_i)$ (and similarly for X_j). In other words, we are not comparing the two models in question with a speaker-independent UBM, but rather directly with each other.

The NCLR distance is then obtained as

$$\text{NCLR}(M_i, M_j) = \frac{1}{N_i} \cdot \log \frac{L(X_i|M_i)}{L(X_i|M_j)} + \frac{1}{N_j} \cdot \log \frac{L(X_j|M_j)}{L(X_j|M_i)} \,, \tag{2.9}$$

removing the dependence on the UBM.

2.2.2 Distances Between i-Vectors or Speaker Embeddings

One of the advantages shared by i-vectors and various DNN-based approaches is that as simple vectors of a fixed dimension, they allow the use of very basic methods for speaker comparison. In contrast to the various likelihood-based distance metrics which are used with GMMs, one of the most common options here is the simple *cosine distance*. This is given by

$$d_C(\mathbf{w}_1, \mathbf{w}_2) = 1 - \frac{\mathbf{w}_1^T \mathbf{w}_2}{\|\mathbf{w}_1\| \cdot \|\mathbf{w}_2\|} \,, \tag{2.10}$$

where $\mathbf{w}_1$ and $\mathbf{w}_2$ are vectors representing two speakers or speech segments. Examples of its use with i-vectors include (Dehak et al., 2011) or (Senoussaoui et al., 2014).

Much less commonly, *Mahalanobis distance* is also used for the same purpose. It is defined as

$$d_M(\mathbf{w}_1, \mathbf{w}_2) = (\mathbf{w}_1 - \mathbf{w}_2)^T W^{-1} (\mathbf{w}_1 - \mathbf{w}_2) \,. \tag{2.11}$$

where W is a covariance matrix of the underlying distribution. In (Larcher et al., 2012), Mahalanobis distance is used for comparisons between normalized i-vectors for speaker verification, with W being the within-class covariance matrix obtained on the development data.

In addition to these simpler metrics, approaches based on probabilistic linear discriminant analysis (PLDA) have also been proposed for i-vectors (e.g. Garcia-Romero and Espy-Wilson, 2011). These rely on a further decomposition of the individual i-vectors into separate speaker-dependent and channel-dependent components. While more computationally demanding, this has been shown to outperform the more traditional cosine distance on the i-vector clustering task (e.g. Sell and Garcia-Romero, 2014; Salmun et al., 2016) and has since been used for x-vectors as well. PLDA is described in more detail below.

Finally, there are also metrics based on neural networks, such as the triplet loss (Hoffer and Ailon, 2015) approaches used by Le Lan et al. (2017) and Song et al. (2018) to score i-vectors and DNN embeddings, respectively.

Probabilistic Linear Discriminant Analysis

Probabilistic Linear Discriminant Analysis (PLDA) is a generative probability model originally proposed by Prince and Elder (2007) and Ioffe (2006) for face recognition.

In the context of speech processing, it is commonly applied to i-vectors or a similar form of speaker representation. PLDA then can serves two purposes: it further decomposes the i-vectors into separate speaker-dependent and channel-dependent components and at the same time it also provides a way to measure the distances between the resulting representations, in the form of a PLDA score matrix.

There are multiple different variants of PLDA used in speech processing. In the most commonly used variant, the i-vector w is decomposed into a speaker factor y and a channel factor x as

$$w = \mu + Vy + Ux + \varepsilon \tag{2.12}$$

where μ is the mean of all the i-vectors in the dataset, ϵ is noise (assumed to be Gaussian, with $P(\varepsilon) = N(0, \Sigma)$), and V and U are matrices which represent the between-speaker and within-speaker variability, respectively. These parameters are obtained on training data.

In the above equation, $\mu + Vy$ represents the speaker dependent part of the i-vector, and $Ux + \varepsilon$ is channel dependent. Vectors x and y are assumed to be generated by a random distribution, most commonly Gaussian.

Finally, the PLDA scores for each pair of i-vectors are obtained as a log likelihood ratio comparing two hypotheses:

$$\text{PLDA score}(w_i, w_j) = \log \frac{p(w_i, w_j | H_d)}{p(w_i | H_s) p(w_j | H_s)} \tag{2.13}$$

where w_i and w_j is a pair of i-vectors, H_d is the hypothesis that w_i and w_j belong to different speakers, and H_s is the hypothesis that they belong to the same speaker.

A more detailed exploration of i-vectors and PLDA can be found in e.g. the work of Silovský (2011).

Chapter 3

Offline Speaker Diarization

In the introductory chapter, several of the most common uses for speaker diarization were listed, with the two most important being audio indexing and rich transcription. In most such applications, the relevant information does not have to be obtained in real-time. Rather, the audio data in question can be processed retroactively, once the entire recorded conversation is available. Therefore it is not surprising that the majority of the world's research in speaker diarization has so far been focused on systems which perform this task offline.

As a consequence, there is a large number of different offline approaches in existence. Online diarization systems, by contrast, are fewer in number and can generally be considered a special, limited variant of the task. Also, many of the common approaches used in offline systems cannot be applied to the online task, although a large portion of the underlying principles can be adapted with some changes.

For these reasons, it is best to examine these two variants of the diarization task separately. This chapter will focus solely on the offline approaches. This will then also serve as the foundation for an overview of online approaches, which will follow in chapter 4.

This chapter describes the main methods used for offline speaker diarization. First, the general framework of a typical offline diarization system is introduced, followed by a more detailed exploration of the individual steps and common methods. The final section presents a short overview of multimodal diarization approaches.

3.1 General Framework of Offline Diarization Systems

A large number of different offline diarization systems can be found in literature. While they employ a wide range of different methods and approaches, the majority of these systems share the same general framework, consisting of a number of standard steps.

This typical framework is shown in Figure 3.1. It begins with the extraction of acoustic features from the audio stream and the detection of speech activity. Following this, the detected speech is split into short segments and then clustered so that each speaker is represented by a single cluster. Finally, most offline systems also include a resegmentation step which further refines the boundaries between

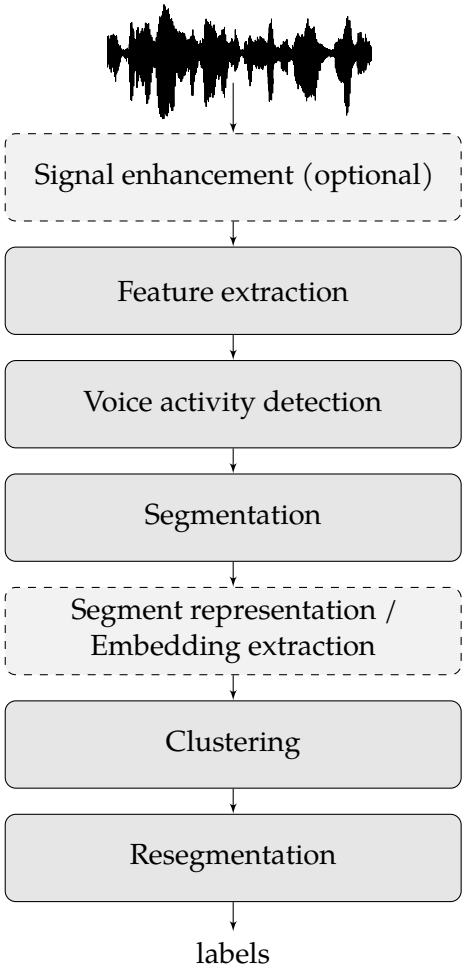

Figure 3.1: Typical framework of a step-by-step offline diarization system.

speakers.

Before this standard process, there may also be an optional first step: the application of some sort of signal enhancement, such as the suppression of noise and reverberation. Such techniques are not generally considered a core part of the diarization pipeline, but they can greatly improve the system's performance, especially with audio obtained under adverse conditions.

Typically, the clustering step employs an agglomerative ("bottom-up") hierarchical algorithm, using one of several popular distance metrics to find the closest pairs of clusters and merging them. Such systems have traditionally been referred to as *bottom-up* systems, as opposed to *top-down* systems, which use the less popular divisive ("top-down") clustering process. However, these are not the only two options.

It should also be noted that the framework as pictured here is merely the most basic scheme, which mainly applies to the classic bottom-up systems (which are still the most common type). However, there are also other approaches which do not strictly follow this scheme – although the basic concepts remain, in some systems, certain steps may be combined with others or left out entirely.

For instance, systems utilizing the top-down clustering approach may not have

a separate segmentation step. Rather, segmentation and clustering are combined into a single iterative process, in which a single starting cluster is iteratively split into individual speaker clusters. Such systems are sometimes referred to as using an *integrated approach*, as opposed to the *step-by-step* approaches which have separate segmentation and clustering steps. This terminology was first introduced by Meignier et al. (2006).

Finally, recently there has been some effort at creating end-to-end neural architectures, which would perform all or most of the diarization process within a single neural network. Thus they do not follow the standard framework shown in Figure 3.1. One example of this can be found in the work of Horiguchi et al. (2020). A more detailed examination of such approaches, as well as DNN-based speaker diarization in general, can be found in the very recent review paper of Park et al. (2021).

In the following sections, the individual steps of the offline diarization process will be explored more closely.

3.2 Signal Enhancement

Similarly to other speech processing tasks, diarization can be greatly affected by audio quality, background noise and other adverse conditions and distortions. For this reason, it can be beneficial to apply some sort of signal enhancement techniques on the input data before the extraction of audio features.

This most commonly includes the suppression of noise (e.g. Sun et al., 2018a), dereverberation (i.e. reducing the level of reverberation in the signal, e.g. Nakatani et al., 2010), or both (e.g. Plchot et al., 2016).

Additionally, acoustic beamforming also falls into this category. Such techniques can be used to take advantage of multi-channel recordings (e.g. from a microphone array or individual microphones worn by speakers). One such example is the widely used acoustic beamforming tool BeamformIt (Anguera et al., 2007), which extracts time-delay features from multiple sound channels and uses them to obtain a single enhanced speech signal.

These optional techniques fall out of the scope of this thesis, and so they will not be examined here in any detail.

3.3 Feature Extraction

The extraction of acoustic features represents one of the first major steps in many speech processing tasks and the choice of features can have a great effect on the overall performance of such systems.

In the field of speaker diarization, our main concern is distinguishing between individual speakers in an audio recording. Thus, the chosen features should contain relevant information characterizing the voice of each speaker. This is very

similar to the task of speaker recognition and as such, many of the same methods can be employed.

The options can be divided into two categories: standard *"hand-crafted"* (or "hand-engineered") features, such as the MFCC, and *learned features* obtained via deep learning.

3.3.1 Standard Acoustic Features

Some of the most popular feature extraction methods used for speech representation are the Mel Frequency Cepstral Coefficients (MFCC, Davis and Mermelstein, 1980), Perceptual Linear Prediction (PLP, Hermansky, 1990) and Linear Predictive Coding (LPC, Atal and Hanauer, 1971). Another overview of these methods can be found in e.g. (Psutka et al., 2006).

In speaker diarization, MFCC is by far the most prevalent method found in literature. These features are commonly used on their own or in combination with other, additional information (see section 3.3.2).

By contrast, the other two methods, PLP and LPC, are seen only rarely and most often in combination with the traditional MFCC, as additional features (e.g. Gallardo-Antolín et al., 2006). In particular, the use of LPC is generally limited to the detection of overlapping speech (see chapter 8). However, one rare example of a mostly PLP-based diarization system can be found in the work of Tranter et al. (2004).

Another notable example of less commonly used acoustic features are the Linear Frequency Cepstral Coefficients (LFCCs), a feature set closely related to the MFCCs. They have been shown to outperform MFCC in speaker recognition tasks under certain conditions, particularly with female voices (Zhou et al., 2011), but otherwise remain rather uncommon in literature. An example of their use can be found in (Fredouille et al., 2009) and they were also used in several of our experiments described in chapter 9 of this thesis.

3.3.2 Additional Features

Although the vast majority of the diarization systems found in literature use acoustic features based on the MFCC, these are occasionally combined with other additional information.

Besides the previously mentioned LPC and PLP coefficients, this may include prosodic features (Friedland et al., 2012), short-term i-vectors (Madikeri et al., 2015), information bottleneck features (Yella and Valente, 2011) or features obtained using deep neural networks (Yella and Stolcke, 2015; McLaren et al., 2015).

In conference meeting diarization, where multiple microphones are routinely employed, it may also be advantageous to obtain spatial features such as the time delay between microphones to improve diarization performance. One such example is the widely used acoustic beamforming tool BeamformIt (Anguera et al., 2007), which was previously mentioned in section 3.2.

3.3.3 DNN-based Features

With the rise of deep neural networks in speech processing, some authors are starting to abandon the use of traditional hand-crafted features like the MFCC, and now rely on neural networks to extract relevant information from the signal. The input of these networks can be in the form of a spectrogram or even the raw waveform.

Such systems often do not even have an explicit feature extraction step, or it is merely a byproduct of a more complicated task. One such example is the work of Miasato Filho et al. (2018), who directly extract speaker embeddings from a spectrogram.

Similarly, the SincNet neural architecture proposed by Ravanelli and Bengio (2018) performs end-to-end speaker recognition from raw waveform. However, the output of the first convolutional layer of this network can also be used as a form of feature extraction, such as in (Garcia Perera et al., 2020).

3.4 Voice Activity Detection

The second step of most diarization systems is the voice activity detection (VAD), also known as speech activity detection (SAD). Its goal is to identify regions of speech in the audio stream.

Traditional VAD methods can be divided into two categories: *energy-based* and *model-based* detection. More recently, there are also DNN-based VAD systems.

Energy-based detection distinguishes between regions of speech and silence based on short-term energy values. In earlier literature, this involves a decision process based on an adaptive threshold (e.g. Prasad et al., 2002). Later, an Expectation-Maximization (EM) approach with two Gaussian components using log-energy as features also become common (Stafylakis and Katsouros, 2011).

This sort of approach may not be able to perform correctly in the presence of a high level of noise or frequent changes in noise level. It may also not be able to distinguish between speech and music very well, making it possibly less suitable for applications where the presence of music is expected, such as in broadcast news.

One example of a diarization system which uses energy-based VAD is in the work of Zheng et al. (2014), who use an energy based 3-state HMM for this purpose.

Besides the energy itself, many approaches also employ other characteristics of the signal, such as signal-to-noise ratio, periodicity or entropy (Ramirez et al., 2007).

Model-based detection approaches involve the use of models such as GMMs to represent speech and non-speech. Depending on the target application, the latter

category may include silence, various types of noise, as well as music. Individual frames are then classified into these categories using methods such as the maximum likelihood criterion.

The model based approach may also be combined with the identification of speaker gender (e.g. Markov and Nakamura, 2007) or bandwidth (e.g. Meignier et al., 2006), by using multiple different models to represent each category of speech – such as "female speaker + telephone", "male speaker + wide band", etc.

DNN-based detection involves the use of a DNN specifically trained to distinguish between speech and non-speech. These can have very different architectures, from simple feed-forward networks (e.g. Diez et al., 2018b) to convolutional neural networks (Thomas et al., 2014; Zelinka, 2018), bidirectional LSTMs (Viñals et al., 2018a), or convolutional LSTMs (Zazo et al., 2016). Matějů (2020) used a combination of a feed-forward network and a weighted finite-state transducer.

The input of the networks likewise varies: possible choices include MFCC or similar acoustic features (Viñals et al., 2018a; Diez et al., 2018b), the log spectrum (Thomas et al., 2014; Zajíc et al., 2018), or even the raw waveform (Zazo et al., 2016).

Similarly to the model-based approaches, such networks can often also be used to distinguish between speech and various types of noise or music, or even for the detection of overlapping speech (this topic is covered in chapter 8).

3.5 Segmentation

The segmentation step of speaker diarization aims at dividing the audio recording into short segments. In the subsequent steps, these segments are merged into clusters corresponding to the individual speakers. In order for the clustering step to perform correctly, each segment should ideally only contain the speech of a single speaker.

Common approaches to the segmentation task can be divided into three groups with different levels of accuracy and complexity.

- Many systems attempt to detect the exact points where a change of speakers occurs, so that they can split the audio stream in these places and create segments containing the speech of only one speaker. Some of the methods used for this purpose will be explored in section 3.5.1.
- Other systems, mainly those used for situations where overlapped speech is less likely to occur, such as broadcast news, may rely on segmentation based on VAD, on the assumption that there is usually a short pause between the speech of two different speakers. This can be seen for example in (Markov and Nakamura, 2007).
- Finally, in some cases, the segmentation step consists simply of splitting the audio into short segments of equal length, possibly in combination with the

VAD approach. This is often seen in diarization systems intended for telephone speech, which typically has short speaker turns and frequent occurrences of overlapping speech, making it difficult to correctly detect speaker turns. Such systems usually include a resegmentation step (see section 3.7) in order to refine the speaker boundaries at a later point. One example of this approach can be found in (Senoussaoui et al., 2014).

This approach has also become more common with the rise of DNN-based speaker embeddings, many of which have been shown to handle short speech segments significantly better than i-vectors and other earlier alternatives (e.g. Snyder et al. (2017) and Patino et al. (2018b)).

In all of the above cases and particularly in the third, the length of the resulting segments has a significant influence on the performance of the subsequent clustering step and with it, on the whole diarization system.

On one hand, longer segments will contain a greater amount of information about the speakers, which should make the subsequent clustering step easier. However, there is also an increased chance of a missed change of speakers being present in the middle of a segment, resulting in segments which contain the speech of multiple speakers and contaminate the clusters. Such occurrences are particularly likely when performing segmentation without any speaker change detection, i.e. in approaches which fall into the second and third groups. In such situations, it may be preferable that the individual segments are as short as possible, so that the relative number and influence of these impure segments is limited.

Very short segments, on the other hand, face a different issue: Many of the common feature extraction methods, such as cepstral coefficients, are intended for both speaker- and speech recognition, meaning that the resulting acoustic features contain information on not only the current speaker, but also the phonetic content of the speech. This can be problematic during the clustering step of speaker diarization, as the system may attempt to create clusters based on phonetic similarity, rather than speaker-dependent characteristics (as observed by e.g. Bozonnet et al., 2011). This issue is not limited to GMMs – even the popular i-vectors demonstrably have issues with very small amounts of data (e.g. Kanagasundaram et al. (2011)).

Thus, we need to strike a balance between these two opposing requirements. The typical choice in literature is around 2-3 seconds, with 1 second being the absolute minimum.

Some more recent systems which use i-vectors or DNN embeddings, such as the one proposed by Sell and Garcia-Romero (2014), resolve the dilemma by using partially overlapping segments – thus increasing the amount of data available for i-vector extraction, while keeping a higher total number of segments.

Finally, although the ideal goal of segmentation is to obtain segments which contain only the speech of a single speaker, this may be impossible to achieve due to overlaps between the speakers. In such a situation we may want to detect these overlaps so that they can be dealt with separately. This issue is examined in section 5.2 and in chapter 8.

It should also be repeated here that not all systems have a standalone segmen-

tation step. As previously mentioned in section 3.1, some of the systems which utilize a top-down approach obtain segment boundaries during a combined iterative segmentation and clustering process. This will be explored in more detail in section 3.6.3.

3.5.1 Speaker Change Detection

The purpose of speaker change detection (SCD) is to identify the instances in an audio stream, where a change of speakers is likely to occur. This often serves as the basis of the segmentation step of a speaker diarization system.

As with the previous steps of the diarization process, there are the traditional approaches – in this case using various distance metrics – and newer DNN-based approaches.

Distance-based Speaker Change Detection

The traditional approach to the problem consists of applying a pair of sliding windows on the signal and computing the distance between their contents. A change is detected on the boundary between the two windows if the distance metric achieves a significant local extreme or, alternatively, as soon as its value exceeds a fixed threshold. Figure 3.2 offers an illustration of the process.

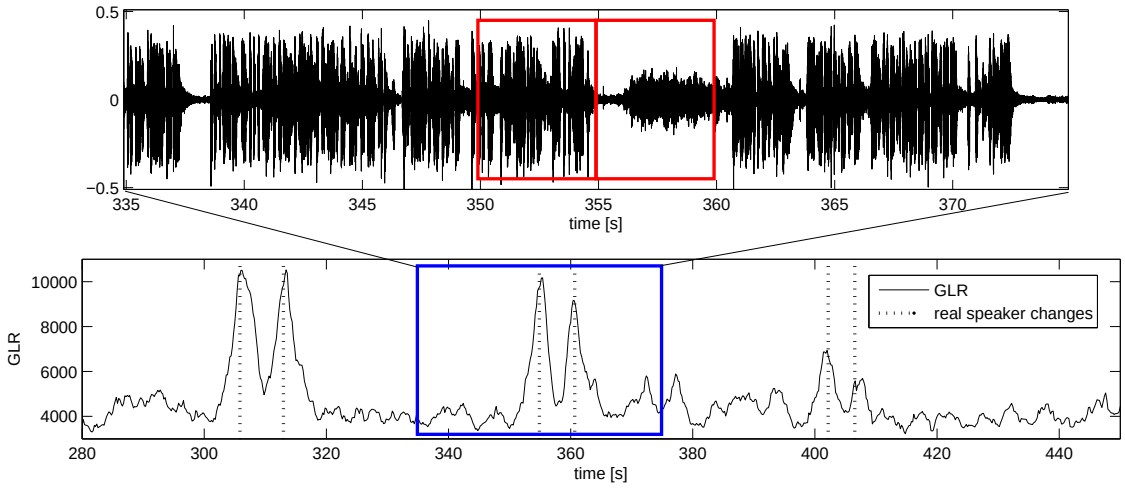

Figure 3.2: Speaker change detection. Upper plot illustrates a pair of sliding windows going through the signal, resulting in the distances shown in the lower plot (the corresponding region is framed). One can see distinct peaks in the distance function at the locations of speaker changes.

The size of the windows may be fixed, or they may be gradually increased until a change is found, at which point the window resets to the original size (e.g. Chen and Gopalakrishnan, 1998).

Historically, the most popular distance metrics used for this purpose included BIC, GLR and KL2, all of which were introduced in section 2.2. More recent alternatives include extracting i-vectors (Neri et al., 2017), DNN-based speaker em-

beddings (Bredin, 2017b) or binary keys (Patino et al., 2017) from each window and calculating the distance between them.

Different distance metrics may also be combined in order to obtain a two-step change detection, such as in the case of the DISTBIC method, proposed by Delacourt and Wellekens (2000) and later modified in (Zochová and Radová, 2005). In this method, KL2 or GLR distance is used to identify potential speaker change points, located in significant local maxima. In the second pass, BIC is used to confirm or reject these points, using the entire neighboring segments for the calculations.

Other examples of the use of two-pass segmentation include (Rouvier et al., 2013) and (Grašič et al., 2010).

It is also important to note that the distance-based speaker change detection approach requires the selection of a suitable threshold or a similar parameter with the same function (such as the penalty weight λ in the case of BIC). This value is data-dependent and typically must be found experimentally on development data.

A more detailed exploration of metric-based speaker change detection can be found in e.g. (Fischerová, 2007).

DNN-based Speaker Change Detection

In more recent works, the traditional distance metrics are sometimes replaced by a specialized neural network. The main principle is generally very similar to the above-mentioned metric-based methods – the network evaluates a sliding window of the conversation, and outputs some sort of distance measure or probability of speaker change.

Examples include the works of Gupta (2015), who uses a feed-forward DNN for this purpose, Hrúz and Kunešová (2016), where an image processing approach is used by applying a convolutional network to a spectrogram[1], Yin et al. (2017), who use a bi-directional LSTM network, and Matějů (2020), who used a combination of a convolutional network and a weighted finite-state transducer.

3.6 Clustering

Following speech segmentation, the next major step of a typical speaker diarization system performs the clustering of individual speech segments, such that each resulting cluster corresponds to a single speaker.

Based on the general clustering approach, offline systems have historically been divided into two categories: *bottom-up* or *top-down*, corresponding to the agglomerative and divisive hierarchical schemes (e.g. Tranter and Reynolds, 2006; Anguera et al., 2012). Of these two options, agglomerative approaches are significantly more common, to the point where the term *top-down* rarely appears in

[1] This is described in more detail in section 9.3.3

recent literature. However, there are also other possibilities which do not fit either of these labels.

3.6.1 Segment Representation / Embedding Extraction

Before the individual speech segments can be clustered, it is necessary to obtain some sort of practical representation of the relevant speaker-dependent data.

While older GMM-based systems typically work directly with the variable-length sequences of feature vectors, most modern approaches aim to represent each individual segment as a single vector of fixed length. This can take the form of i-vectors, x-vectors, binary keys, or various DNN-based embeddings.

The specifics of these forms of speaker representation and the process of obtaining them have already been discussed in section 2.1, so the topic will not be revisited here. Instead, the next sections will present the different clustering approaches which can be find in literature.

3.6.2 Agglomerative (Bottom-up) Clustering

The *bottom-up* category refers to systems which use the agglomerative hierarchical clustering (AHC) scheme, which is still the most popular choice found in literature.

In this approach, each speech segment first represents an individual cluster. These are then progressively merged based on the closest distance between pairs of clusters. The process typically ends when the minimum distance exceeds a fixed threshold or a target number of clusters is reached.

In older systems, the individual clusters are typically represented by GMMs, which are trained on the data from the given cluster. Similarly to the previously detailed process of speaker change detection, the distance between clusters is then usually calculated using one of the distance metrics presented in section 2.2, such as BIC (e.g. Rouvier et al., 2013), CLR or NCLR.

More recent works have switched from GMMs to vector-based speaker representation, in the form of i-vectors, binary keys, or x-vectors and various other DNN-based speaker embeddings (see section 2.1 for details). In this approach, each segment produces a single vector representing the speaker. These can then be easily clustered using simpler methods, such as the cosine distance, used by e.g. Senoussaoui et al. (2014). Alternatively, many recent systems use PLDA scoring (e.g. Silovský (2011), Sell and Garcia-Romero (2014) and many others).

Finally, a small number of authors have very recently proposed agglomerative clustering algorithms based on deep learning. Two such works were published by Aronowitz et al. (2020), who use a neural network for scoring the distances between clusters, and Singh and Ganapathy (2020), who presented a self-supervised clustering framework which also learns speaker embeddings.

3.6.3 Top-down Clustering

Top-down clustering approaches are significantly less common in literature than the previously described bottom-up approaches.

The top-down clustering process starts with a single cluster containing all of the speech segments. This represents unlabeled speech. In each clustering step, we first select a suitable unlabeled speech segment from this cluster, forming a new speaker cluster. Then, the entire speech is reclassified into the currently existing clusters (including the original one). Typically, the individual clusters are represented by GMMs which are updated at the end of each step. (Anguera et al., 2012)

It is common for such approaches to use HMMs to model the transitions between individual speakers, with Viterbi realignment being applied for reclassification. This allows the system to identify the speaker boundaries and makes a prior standalone segmentation step largely unnecessary. In (Meignier et al., 2006), only a gender and bandwidth detection is performed as an initial segmentation.

Examples of top-down systems include (Meignier et al., 2006) and (Fredouille et al., 2009).

3.6.4 Other Approaches

Besides the bottom-up and top-down approaches, there are also other clustering techniques which can be used for speaker diarization and do not fall into either of these categories. This section presents a very brief overview of some notable examples.

- In the special case where the number of speakers is known in advance (such as in most telephone speech) or can be estimated by other means, one may simply use k-means clustering instead of AHC (e.g. Shum et al. (2011), Wang et al. (2018), Zajíc et al. (2018)). There also exists a variation of the k-means algorithm, called "X-means" (Pelleg and Moore, 2000), which is capable of estimating the number of speakers automatically and was used by e.g. Dimitriadis and Fousek (2017).

- One approach, known as the Information Bottleneck (IB), is based on an information-theoretic framework. It tries to find a partition of the audio stream which "maximizes the mutual information between observations and variables relevant for the problem while minimizing the distortion between observations" (Vijayasenan et al., 2009).

- In (Rouvier and Meignier, 2012) and (Broux et al., 2018), authors search a globally optimal solution to the clustering task by presenting it as an Integer Linear Programming (ILP) problem.

- The *spectral clustering* algorithm (Ng et al., 2001) has been successfully used for speaker diarization by multiple authors, including Ning et al. (2006), Shum et al. (2012), Wang et al. (2018), and Park et al. (2019a). The main

principle of spectral clustering lies in constructing an affinity matrix from the data (e.g. i-vectors) and then using k-means clustering based on the eigenvectors of the affinity matrix.

- Kounadis-Bastian (Kounades-Bastian et al., 2017; Kounadis-Bastian, 2017) proposed a method which combines multichannel source separation and speaker diarization.

- Finally, Valente and Wellekens (2006) proposed a diarization approach based on Variational Bayes (VB). This principle has since been expanded by others (e.g. Kenny, 2008), most notably into the Bayesian HMM (VB-HMM) approach (Diez et al., 2018a). It has also become popular as a form of resegmentation or final refinement following the more traditional segmentation and clustering steps (see section 3.7).

3.7 Resegmentation

Most offline diarization systems include a resegmentation step in the final processing stages. The goal is to refine speaker boundaries after an initial labeling has been obtained. This is particularly important in cases where the initial segmentation step does not include speaker change detection (see section 3.5) as the segment boundaries do not properly correspond to the speaker turn points.

Typically, resegmentation is an iterative process. Following the main clustering step of the diarization system, the obtained segment labels are used to create a new model for each cluster, trained on all of the relevant data. These models then serve to reclassify the entire conversation, most commonly using the Viterbi algorithm (e.g. Kenny et al., 2010) or frame-by-frame, with subsequent smoothing (e.g. Zajíc et al., 2016). The whole process may be repeated a set number of times or until convergence is reached.

Alternatively, as mentioned in section 3.6, diarization approaches based on Variational Bayes (including VB-HMM) have also become relatively popular as a form of resegmentation (e.g. Kenny et al., 2010; Sell and Garcia-Romero, 2015). Here, one may also view the traditional segmentation and clustering steps as merely the *initialization* of a VB diarization system (e.g. Diez et al., 2018b).

3.8 Multimodal Speaker Diarization

In recent years, an increasing amount of research has been dedicated to multimodal diarization techniques, which combine information from the audio and video modalities in order to perform the diarization of audio-visual recordings.

When available, a video recording can offer a significant amount of additional information compared to mere audio. Multimodal systems such as the ones presented by Noulas et al. (2012), Bredin and Gelly (2016), or Kapsouras et al. (2017) employ face tracking techniques, often together with lip-movement detection, in

order to identify the most likely current speaker in the video modality. By combining this information with the traditional audio diarization methods, an improved performance can be obtained compared to an audio-only system (e.g. Campr et al., 2014; Ramos-Muguerza et al., 2018).

Multimodal diarization techniques are outside the scope of this thesis and are acknowledged here only in the interest of completeness. The rest of this thesis will focus exclusively on methods which are limited to the audio modality.

Chapter 4

Online Speaker Diarization

Following the overview of offline diarization approaches, which was presented in the previous chapter, this chapter will focus on online speaker diarization.

Online diarization can be considered a special limited case of the speaker diarization task, in which the system is required to process an incoming audio stream in a sequential manner and to output the corresponding labels in real-time.

While many common applications of speaker diarization do not have this requirement, there are other potential uses for which online processing is necessary. This includes possibilities such as HCI and HRI, as well as any other situations in which the information obtained by speaker diarization is intended to be used by other real-time systems, such as one which performs automatic speech recognition (ASR).

The online task is more difficult than offline diarization, because the system needs to make decisions based on incomplete data and in a limited amount of time. Additionally, these restrictions mean that many of the popular methods and approaches which have proved successful for offline diarization, such as hierarchical clustering or resegmentation (see Figure 3.1), cannot be employed here. As a consequence, online systems generally exhibit worse performance compared to those which operate offline. This can also be observed in the overview of recent diarization systems which will be presented in section 7.2.

This chapter contains an overview of the main differences between offline and online diarization and the additional challenges posed by the latter. This is followed by a review of the four main groups of approaches found in literature.

4.1 Additional Challenges of Online Diarization

As stated in the introduction of this chapter, online diarization systems face additional challenges and restrictions compared to those which perform speaker diarization in an offline manner. These challenges include limited resources, particularly with regards to available data and processing time, as well as the unsuitability of certain types of otherwise popular approaches. This section will focus on the former aspect, while the latter will be explored later in the chapter.

4.1.1 Limited Data

The main characteristic of online diarization is that an incoming audio stream is processed sequentially, as it is being received by the system. This is a significant difference from offline diarization, in which the entire conversation is recorded in advance and is available at the start of the diarization process.

This sequential processing means that the decisions at any given time have to be based only on previous data, as the system does not have access to future information.

This limitation is particularly significant at the beginning of an audio stream. There, the amount of available data is very small, which makes correct decision-making difficult and the system can be expected to make a higher number of errors. These may in turn negatively influence the rest of the diarization process, further degrading the performance of the system.

In order to minimize the above-mentioned issue, it is very important that online systems are properly initialized, such as by training the system on a very similar set of data, or by making use of prior information about speakers. This matter will be further explored in section 5.3.

4.1.2 Processing Time

Another significant restriction which applies to online diarization systems relates to the processing time. Specifically, the system must by necessity be able to process the audio stream faster than real time, in order to keep up with the incoming data.

Among other things, this limits the degree to which previous data can be used in the processing of a new speech segment. Particularly in longer recordings, there may not be enough time to revisit the entirety of the previously labeled speech in any way (such as for comparison with the new segment), so the system needs to be able to work with more simplified representations of the individual speakers.

A similar limitation applies to the complexity of the used methods. For example, it may often be possible to improve the performance of offline systems to some degree by e.g. increasing the complexity of speaker models at the cost of computation speed. However, in online systems, which are subject to the aforementioned time constraints, the possibilities of such improvements are significantly more limited.

4.1.3 System Latency

Besides the overall processing speed, online diarization systems are typically also bound by requirements regarding latency, meaning the delay between the input audio stream and the system's corresponding output.

While this delay cannot be completely avoided, mainly due to the reasons below, it is usually required to be as short as possible. This is particularly important in situations where the information about speaker turns, which is provided by the

diarization system, is intended to be immediately used for other time-sensitive purposes, such as in an ASR system.

System latency is essentially determined by two factors. The main source of latency is the amount of "future" data needed for a decision. Similarly to the offline segmentation methods which were discussed in section 3.5, an online system requires a certain amount of further data after a potential speaker change point, in order to detect this event and to be able to determine the correct label for the new speaker. As with offline segmentation, the exact amount of necessary data depends on the used methods, but usually, at least 1-3 seconds of speech are needed for a reasonably reliable result.

Note: The above does not necessarily mean that the individual speech segments are limited in length. It is also possible to determine the speaker based on only the first few seconds of an utterance, even before it is finished. This approach is used by e.g. Markov and Nakamura (2007).

The second factor which is relevant to the system's latency is the time needed to *process* the aforementioned minimum amount of data and to assign a correct label to it – such as by extracting a speaker embedding and comparing it to existing clusters. Such processing can occur after this interval is obtained in its entirety, directly adding to the delay given by its length, or it may be possible to perform some of the necessary calculations, such as likelihood computation, on a frame-by-frame basis as the data is received, thus reducing the added delay.

When listing the latency of a system, authors typically only include the first factor - the amount of data required for decisions. The second factor, processing time, largely depends on the computational abilities of the specific device which runs the system.

Typically, the target latency of the online diarization systems found in literature ranges between 1 and 5 seconds, as can also be seen in an overview of recent online systems which is presented in Table 7.8 on page 65. Shorter values are usually not achievable without significantly compromising the accuracy of the results (e.g. Soldi et al., 2015), while longer delays may be impractical for real-time applications.

4.2 Online Diarization Framework

Due to the previously stated additional requirements present in online diarization, the typical structure of online systems differs from the offline framework presented in Figure 3.1 in multiple points, although the majority of the basic steps still apply and can remain relatively similar.

First of all, an online system needs to perform all the individual steps on a segment-by-segment basis. This is different from offline systems, where the individual steps are typically performed consecutively, each starting only after the previous one is finished on the entirety of the data.

The second important difference, which has already been mentioned, lies in the fact that some of the techniques used in offline systems are unsuitable for on-

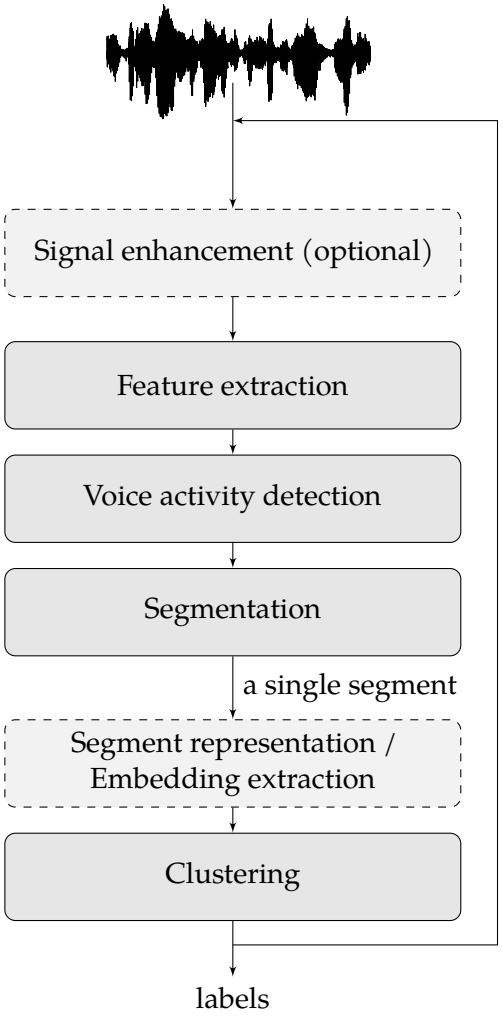

Figure 4.1: The typical framework of an online diarization system. The main differences from the offline framework shown in Figure 4.1 are the absence of a resegmentation step, as well as the sequential nature of the process, here signified by the back arrow.

line processing. This reflects in significant changes in the clustering process, as well as the complete absence of a resegmentation step.

If we use the previously explored offline framework as a baseline, the online variants of the individual steps will be subject to some changes and limitations compared to the descriptions in chapter 3. The main differences are as follows:

- The *feature extraction* and *voice activity detection* steps both need to be performed in an online, left-to-right manner. However, otherwise they do not usually significantly differ from the aforementioned offline versions, which were described in sections 3.3 and 3.4, respectively, and as such, will not be revisited here. The same also applies to the optional *signal enhancement* step.
- The *segmentation* step can be likewise relatively similar to the approaches used in offline systems (described in section 3.5). However, the lack of future data is a limiting factor in the case of segmentation using distance-based speaker change detection (presented in section 3.5.1) or, to a lesser extent,

segmentation based on VAD, as they both require access to a certain amount of data past a possible segment boundary. This can also play an important role in the system latency, as discussed in section 4.1.3.

In particular, this means that if speaker change detection is desired, it may be preferable to use a simple threshold-based approach (in which a change of speakers is found at the first point where the criterion exceeds a set value), rather than the alternative of searching for local extremes, which is common in offline systems, but would require a much greater delay.

- *Segment representation / embedding extraction* is essentially unaffected, as most approaches already function on a segment-by-segment basis.

- For the *clustering* of speech segments, we cannot use hierarchical approaches, which are typical in offline systems, as they require access to the entire audio recording at once. Instead, sequential clustering algorithms are typically utilized, allowing the system to make decisions on a segment-by-segment basis. This will be explored in more detail in section 4.3.

- Finally, as previously noted, it is likewise not possible to employ *resegmentation* in order to refine speaker boundaries (see section 3.7), as all decisions must be made in real-time and are final and unchangeable in the context of the online application.

  This restriction means that proper segmentation becomes even more critical and it is highly important that the boundaries between individual segments correspond to the actual speaker change points as accurately as possible.

4.3 Online Diarization Approaches

The previous section presented a brief summary of the main changes between the typical structure of offline and online systems. This will now be followed by a more detailed exploration of the specific methods which appear in current literature.

Because of the historically lower need for online processing in most common applications of speaker diarization, there has been relatively little interest in this area until recently. As a consequence, there have also been significantly fewer relevant publications compared to those focused at offline diarization. Additionally, a large portion of the existing literature targets specific conditions which involve additional assumptions or sources of information, rather than focusing on the most general form of the diarization task.

Based on the presence of these additional assumptions and the general approaches used, most of the systems found in literature can be divided into several groups:

- The most general approaches, which also most closely follow the framework presented in section 4.2, typically involve the use of *sequential clustering* with an unknown number of speakers.

- A number of authors simplify the task by assuming that the models of all speakers are obtained in advance. This essentially transforms the diarization problem into one of *speaker identification*.
- With sufficient processing power, one may employ *hybrid online-offline* diarization. Such systems use very fast offline algorithms to periodically revisit past data and improve online decisions.
- Finally, a number of systems rely heavily on *additional sources of information* such as microphone arrays or cameras, in order to determine which individuals are currently speaking.

As these main groups significantly differ, they will be explored individually in the following sections.

4.3.1 Sequential Clustering with Unknown Speakers

Section 3.6 presented a number of different clustering approaches which can be employed by *offline* diarization systems, the most common being AHC. However, while these methods have proved successful on the offline task, they are unsuitable for use in *online* systems, as their use generally requires access to the entirety of the data at the beginning of the diarization process. For this reason, online systems have to rely on other approaches.

Of the common solutions, the perhaps most generally applicable one involves the use of sequential clustering methods with an unknown number of clusters. In this approach, the individual speech segments, obtained by splitting the audio stream, are processed in a chronological order, with each of them being either assigned to an existing cluster or used to create a new one, as illustrated in Figure 4.2.

The implementation details can differ, but the basic clustering process typically proceeds as follows (Liu and Kubala, 2004):

- At the beginning of the diarization process, there are usually no existing clusters, although it is often possible to use previously obtained data for initialization. If no such initial clusters are used, the first cluster will be created from the first speech segment.
- Then, for each new segment X_i, the system must decide whether X_i belongs to
  a) one of the already known speakers, which are represented by N previously created clusters $C_1, \ldots, C_N$, or
  b) an entirely new speaker.

  For this, the system will usually first find the cluster C_j which is the closest to X_i according to a specific criterion, such as one of the distance metrics introduced in section 2.2.

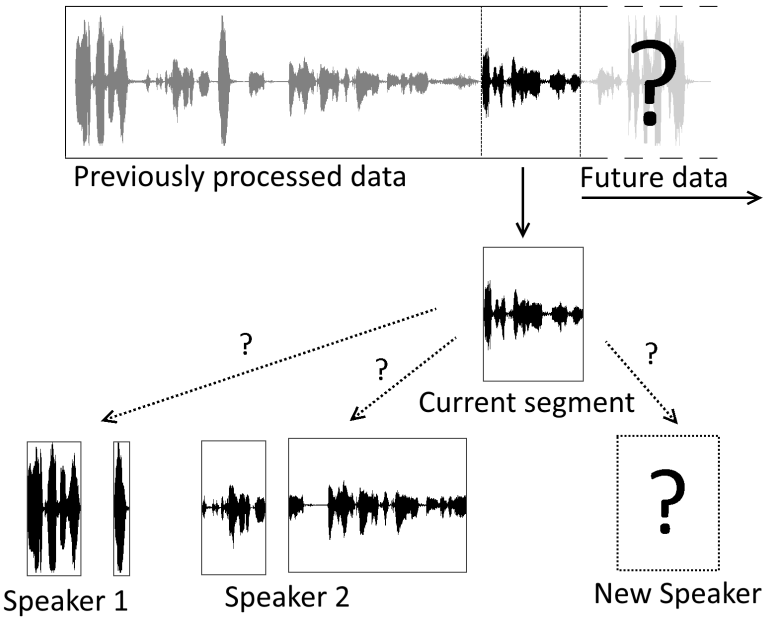

Figure 4.2: Sequential clustering with an unknown number of speakers.

Then it is necessary to decide if the distance between X_i and C_j is such that X_i and C_j are likely to represent the same speaker. This is often decided with the use of a preselected threshold, found on development data.

  - If X_i is believed to belong to C_j, X_i is labeled as the j-th speaker and C_j is updated.
  - Otherwise, X_i is used to create a new cluster C_{N+1}.

  Following the terminology used by Markov and Nakamura (2007), the above decision will be also referred to as *novelty detection*.

- After updating the correct cluster, the system obtains the next speech segment and the above step is repeated. This continues until the end of the audio stream.

The most critical part of the clustering process lies in the novelty detection. The system must be capable of correctly deciding whether a given speech segment belongs to the most similar speaker or whether it represents an entirely new one. An incorrect decision in either direction, particularly in the beginning of the audio stream, can have a significant effect on all future decisions and as such, can drastically reduce the overall performance of the system:

If a new cluster is created erroneously, there will be multiple competing clusters for the same speaker and future speech segments of this speakers may be split between them, creating the illusion of changing speakers. Conversely, if the first occurrence of a previously unseen speaker is mistakenly assigned to an existing cluster, the corresponding two speakers may remain merged and the system will be unable to distinguish between them.

Such issues will be examined more closely in chapter 5.

In older systems, the individual clusters are typically represented by GMMs. The updating of the clusters then consists of either the creation of a new GMM (this would, however, be rather computationally intensive), or of its update using e.g. MAP adaptation. Alternatively, if one uses i-vectors or a similar form of segment representation, then clusters can simply consist of a gradually expanding set of individual vectors.

Examples of online diarization systems which employ the sequential clustering approach with GMM-based cluster models include the works of Markov and Nakamura (2007, 2008), Geiger et al. (2010), Soldi et al. (2015) and Oku et al. (2012). Additionally, Grašič et al. (2010) also employ GMMs, albeit in a slightly different manner.

- Markov and Nakamura (2007, 2008) propose a GMM-based online system which starts with no speakers and iteratively creates new speaker models as time progresses, by adapting one of a pair of universal models (male and female) using an incremental EM algorithm. The system's segmentation step is entirely VAD-based with no maximum segment length, but in order to lower the latency, clustering decisions are based on only the beginning part of a segment, with the entire segment being used for updating models.

  This system also served as the basis for the experimental implementation described in section 9.2 and a detailed description of the altered system can be found there.

- Similar approaches were also chosen by Geiger et al. (2010) and Soldi et al. (2015). The main difference compared to the former system is in the use of MAP for GMM adaptation as opposed to the incremental EM algorithm.

- Grašič et al. (2010) describe a diarization system largely based on the use of the Normalized Cross-Likelihood Ratio (NCLR, see section 2.2.1) for both segmentation and clustering. Speech segments are obtained using a speaker change detection algorithm combining the BIC and NCLR metrics, as well as a special normalization technique, which includes a comparison with reference points and a window length dependent decision threshold.

  Unlike the majority of the systems described in this section, this one does not use simplified models such as GMMs to represent the individual clusters. Instead, a small subset of each speaker's assigned segments is kept as-is and is used for comparison between each new segment and the clusters, using the NCLR distance metric as a criterion.

- Oku et al. (2012) suggest an approach which incorporates phoneme recognition. A phoneme recognizer is used to find the boundaries between phonemes and to classify feature vectors into two classes: vowels and consonants. Only the phoneme boundaries are considered as potential speaker change points and both speaker change detection and speaker clustering are based on the use BIC, using models consisting of two Gaussian components – one for each phoneme class.

It is worth noting that all of the above systems employed a UBM trained specifically on very similar data as their evaluation set, with most using a different subset

of the exact same corpus. This raises the question of whether such systems can generalize or if they have be trained for a specific purpose.

More recently, several authors have proposed online systems based on i-vectors or neural speaker embeddings.

- One example of an i-vector based online diarization system was presented by Zhu and Pelecanos (2016) Their system, which is based on the work of Shum et al. (2013), extends the latter's offline approach to online diarization by introducing an adapted i-vector transform which is applied to all observed i-vectors when processing every new segment. This allows them to better discriminate between segments of different speakers as the length of the conversation increases. Although limited in its restriction of the problem to only two speakers, the paper nevertheless offered a novel approach with a potential for future improvements.

- Patino et al. (2018b) likewise employ i-vectors, with sequential clustering based on cosine distance.

- Wang et al. (2018) developed a new online clustering algorithm called *links clustering* (published as Mansfield et al., 2018) and then used it in combination with either i-vectors or LSTM-based speaker embeddings (*d-vectors*).

  The links clustering approach models individual speakers as clusters consisting of multiple smaller subclusters. The algorithm estimates the probability distributions of individual clusters and subclusters based on the currently available vectors. These estimates are used to classify new incoming vectors and are updated after each new one. While the assignment of vectors to subclusters is permanent, the subclusters themselves can be merged or reassigned to a different cluster.

- Zhang et al. (2019) built upon the aforementioned system by proposing a new DNN-based clustering approach, using what they call *unbounded interleaved-state recurrent neural network* (UIS-RNN). In this system, different speakers are represented by different RNN states.

- Ghahabi and Fischer (2019) proposed so-called *speaker-corrupted embeddings*. These are extracted by a DNN from UBM supervectors which had been "corrupted" by data from other speakers. This is meant to improve the generalization power of the network. The resulting speaker embeddings are then clustered using a simple sequential algorithm with cosine similarity.

- Finally, von Neumann et al. (2019) presented an unusual system which performs diarization via source separation. A recurrent neural network processes the audio stream in blocks of 2.5 seconds, and iteratively separates each block into multiple signals corresponding to individual speakers. The system tracks speaker identities between blocks, thus acting similarly to sequential clustering and providing speaker diarization which can handle overlapping speech.

4.3.2 Speaker Identification Approaches

A number of diarization systems bypass the issue of novelty detection by assuming that the models of all speakers are obtained in advance, essentially transforming the diarization problem into one of speaker identification. In some circumstances, such as meetings with known participants, this may be a reasonable assumption, though the approach generally fails in the presence of one or more unexpected additional speakers.

Unlike the sequential clustering approaches which were in the previous section, the systems found here do not entirely follow the basic online framework which was presented in section 4.2, although much of the process remains similar.

An example of this approach can be seen in the system proposed by Vinyals and Friedland (2008) (later also described by Friedland et al., 2012). This system requires one minute of speech from each of the participants to be recorded prior to a meeting, in order to construct a GMM for each speaker. Alternatively, it is possible to use models obtained from a more traditional offline diarization of an earlier meeting with the same participants.

The online system itself then performs speaker identification by calculating the likelihoods of individual frames against each GMM and applying majority voting over a window of 2.5 seconds. The models themselves remain unchanged during this process.

Similarly, Soldi et al. (2016) propose a "semi-supervised" system partly based on their earlier work (Soldi et al., 2015) which used the sequential clustering approach. In this newer system, all speakers are assumed to be known a priori, but only a very small amount of speech (as little as 3 seconds) from each is required in advance. This small amount is used to construct initial speaker models, which are then incrementally adapted using MAP adaptation during the online diarization process.

The clear weakness of the two above approaches lies in the inability to correctly identify unexpected speakers whose models were not obtained in advance. This could be potentially mitigated by various methods of "unknown speaker detection", although such attempts may also lead to a significant number of false alarms. For instance, according to the authors of (Friedland et al., 2012), all such experiments "decreased [the] total score significantly on the development set."

4.3.3 Hybrid Online-Offline Approaches

As the performance of modern computers increases and fast GPU-based computation becomes more common, it is possible to process larger amounts of data or perform more computationally demanding calculations in relatively short time. In regards to speaker diarization, this not only enables the use of more complicated algorithms and more complex models, but also means that simpler offline diarization systems can now operate at a very small fraction of real time. For instance, Friedland (2012) reported a real-time factor of 0.004 in a GPU-optimised

system.

Such high speed allows the creation of hybrid online-offline systems which continuously perform *offline* diarization of previously seen data and use the resulting information to improve the decision process of the *online* component.

One of the earliest examples is the system proposed by Vaquero et al. (2010), which combines speaker-identification-based online diarization with standard bottom-up clustering. The system consists of two subsystems running in parallel:

- One subsystem constantly performs offline bottom-up diarization on all available data up to the current time.

  It starts with the first 60 seconds of the audio stream and every time the process is completed, it is initiated again with the addition of the data obtained in the meantime. Meanwhile, the resulting labels are remapped to ensure consistency with previous iterations and sent to the online subsystem.

- The second subsystem performs speaker identification in a similar way to the system of Vinyals and Friedland (2008) which was previously described in section 4.3.2. However, instead of using an unchanged set of speaker models for the entire duration, the models are adapted every time the offline subsystem outputs a new set of labels. This way, the speaker models gradually improve and new speakers can be discovered during the diarization process, albeit with a delay.

As the amount of available data increases, so does the accuracy of the offline subsystem, but also the time between updates. This means that in very long conversations, it can take a long time for the online subsystem to start correctly identifying a late appearing speaker. However, this effect can be mitigated by limiting the amount of data used by the offline subsystem to only the most recent X seconds, such as in (Friedland, 2012) and (Dimitriadis and Fousek, 2017).

Naturally, the offline subsystem needs to be very fast and simple, and should not require any significant recalculations when adding more data. A simple AHC clustering of i-vectors (or similar vector-based representations) makes a suitable choice, as these only need to be extracted once and, as discussed in section 2.2.2, allow for very fast clustering. This option was used by the above-mentioned Dimitriadis and Fousek (2017).

In this thesis, the hybrid diarization concept will also be briefly explored. The relevant experiment will be presented in section 9.3.5.

4.3.4 Multimodal Approaches

A number of systems rely on additional sources of information, such as microphone arrays or cameras, in order to determine which individuals are currently speaking. In the context of online diarization, such setups are particularly common in systems intended for HCI and HRI applications.

As it was already stated in the analogous section 3.8 that multimodal techniques are out of scope of this report, following will be only a very brief overview of several notable examples.

In one interesting and unusual example, Minotto et al. (2015) propose a multimodal system which performs speaker diarization purely by means of face tracking and signal source localization, with no voice comparison being performed. The system employs a microphone array, a color camera and a depth sensor and is capable of determining the spatial location and voice activity of up to three simultaneous speakers in a human-computer interaction scenario.

In another system, Gebre et al. (2014a) similarly perform speaker and signer diarization solely on the basis of movement, working on the assumption that speech and gesticulation are highly correlated. In a subsequent paper (Gebre et al., 2014b), this was extended by also adding more traditional acoustic modeling.

Finally, Ito et al. (2018) suggest an online diarization algorithm which is based purely on estimated direction of arrival of each source signal, obtained via a microphone array.

Other notable online systems include the works of Noulas and Krose (2007), who combine acoustic information with face tracking, and Schmalenstroeer et al. (2009), who also add a microphone array, in order to track the location of speakers within a room. More recently, Gebru et al. (2017) employed visual and spatial tracking using a camera and a pair of microphones.

Chapter 5

Main Issues in Speaker Diarization

Despite the large amount of research that has been dedicated to speaker diarization, there are still multiple problematic areas and restrictions which have not been fully resolved or are difficult to deal with. This chapter explores a few of such issues.

During the early stages of work on this thesis, three different issues were identified as important obstacles to speaker diarization: very short speaker turns, overlapping speech, and the problem of initialization in an online diarization system. The first two of them apply to the diarization task in general, while the last one is specific to online diarization.

Besides these three topics which are examined below, diarization performance can also be significantly negatively affected by poor audio quality – including background noise, reverberation, channel distortions and other similar effects. However, this is a problem shared by all speech processing tasks, and not specific to speaker diarization.

5.1 Very Short Speaker Turns

One of the common obstacles which can significantly degrade the performance of a diarization system is the presence of very short speaker turns. This can be very difficult to detect and also poses problems during segmentation and clustering.

Background:

Most of the diarization approaches which were introduced in chapters 3 and 4 involved splitting the audio stream into short segments of speech and merging these into clusters corresponding to the individual speakers.

In section 3.5, it was also stated that in order to achieve the best results during the clustering process, these individual segments must be sufficiently long, so that they contain an adequate amount of information about the speakers. This ensures that segments are clustered based on the speakers themselves, and not the phonetic content.

Based on the results reported in literature, it appears that in the absence of short speaker turns, the ideal segment length for GMM-based systems is at least 2–3 seconds. With shorter segments, system performance tends to significantly

degrade, as shown for example by Soldi et al. (2015) and Markov and Nakamura (2007). Speaker recognition studies have also shown that many other forms of speaker representation, including the popular i-vectors, have issues with very short utterances (e.g. Kanagasundaram et al. (2011)).

A similar requirement also applies to the speaker change detection methods presented in section 3.5.1, which attempt to identify the points where speakers change by calculating the distance between a pair of sliding windows. Here, too, it is important to have a sufficient amount of data in order to obtain an accurate result, and the recommended window length is similar. For traditional GMM-based distance metrics it typically ranges between 1–5 seconds (Tranter and Reynolds, 2006).

The Problem:

In ideal circumstances, the above-mentioned length of 2–3 seconds is sufficient for both the length of the speaker change detection windows and the speech segments themselves. However, it proves problematic in the presence of very short speaker turns.

When working with spontaneous speech, particularly in the case of telephone conversations, one may frequently encounter one-word utterances such as "Yes", "Hello" or "Maybe". These can be spoken very quickly, leading to speaker turns which are shorter than one second.

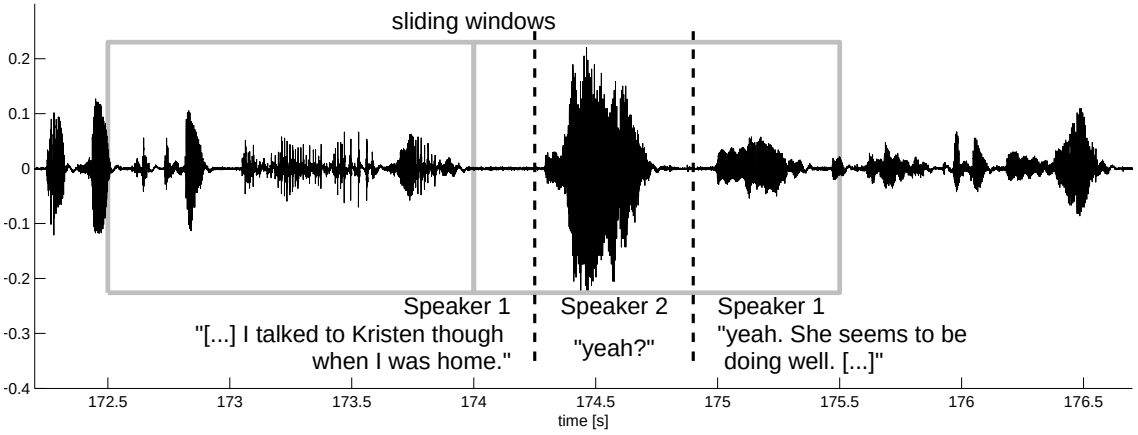

Figure 5.1: Illustration of an attempted distance-based speaker change detection with very short speaker turns, shown on an excerpt from the CALLHOME American English corpus of telephone speech (Canavan et al., 1997). The length of the two windows used here is 1.5 s, while the utterance of Speaker 2 ("yeah?") is only 0.55 s long, making it difficult to detect the two changes of speakers with this setup.

In Figure 5.1, we can see this issue illustrated on an example from the CALLHOME corpus of telephone speech. The illustration shows an attempt to perform speaker change detection using a pair of windows which are both 1.5 s long. However, the utterance of Speaker 2 ("yeah?") is only 0.55 s long and we can make the following observations:

a) At any given position in the surrounding area, both of the windows will contain mostly the speech of Speaker 1.

b) While moving the sliding windows over the signal, there is a relatively long interval in which the entirety of the second speaker's utterance lies within a single window and the amounts of speech from each speaker within each window are constant.

As a consequence, we can expect that the resulting distance between the contents of the two windows will likely remain relatively low, with a slightly increased value over a larger area, rather than a single distinct peak at the boundary between speakers. This, in turn, makes it significantly more difficult to detect the presence and locations of both change points.

Similarly, even if we were to correctly locate the two points, the resulting speech segment would be too short to reliably cluster.

Possible Improvements and Workarounds:

Because of situations such as the one described above, diarization systems which focus on telephone speech and other natural conversational data, where such occurrences are most common, often forgo speaker change detection altogether. Instead, they often simply split the audio stream into short segments of equal length, compensating for the inaccurate segment boundaries by performing a resegmentation step at a later point (e.g. Senoussaoui et al., 2014; Sell and Garcia-Romero, 2015). In (Zajíc et al., 2016), we confirmed this to be a reasonable solution in this context, as also shown in section 9.3.3 of this thesis. However, this method of compensation is not available to online systems.

In either case, there is also still the problem of obtaining accurate representations of speakers from these short segments, so that the system can cluster them correctly.

- One idea pursued by several authors is the use of phone-based normalization for lowering the variability of very short utterances, and thus allowing the use of shorter segments.

  Bozonnet et al. (2011), for instance, have observed that the linguistic content of speech has a significant influence on the performance of bottom-up clustering, as such a system may attempt to cluster short segments based on phonetic similarity, rather than speaker-dependent characteristics.

  Following this observation, in a subsequent work (Bozonnet et al., 2012) they propose a process for suppressing the phonetic variability, by training a set of transforms used for phone-dependent normalization of acoustic features. They refer to this process as Phone Adaptive Training (PAT) and show that it improves the performance of their diarization system. In a later paper by Soldi et al. (2014), this approach is also expanded to i-vector based speaker recognition.

Similarly, Larcher et al. (2012) explore the possibility of improving the accuracy of i-vector based speaker verification in utterances under 3 s of length by applying phonetic-content-dependent i-vector normalization.

- Alternatively, the authors of some i-vector or x-vector based systems, such as Sell and Garcia-Romero (2014), employ partially overlapping segments – thus achieving a higher density of segmentation while keeping more information in each segment. This was also used in the system described in section 9.3 of this thesis.

- Finally, while many forms of speaker representation, including the popular i-vectors, demonstrably have issues with small amounts of data (e.g. Kanagasundaram et al. (2011)), DNN-based speaker embeddings such as x-vectors have been shown to be more resilient in this regard (e.g. Snyder et al. (2017) and Patino et al. (2018b)) and can allow for finer segmentation. One notable example is in the work of Wang et al. (2018), who use a maximum segment length of 400 ms in their LSTM-based system.

5.2 Overlapping Speech

A second potential issue often faced during speaker diarization is the presence of overlapping speech, i.e. intervals in which multiple speakers are talking simultaneously. This frequently occurs in spontaneous conversations, where speakers may regularly interrupt each other or interject short utterances while the original speaker keeps talking.

Such occurrences are generally very difficult to detect or correctly label, yet they can have a significant negative effect on the performance of a diarization system. Besides contributing to missed speech rate (see section 7.1), incorrectly labeled overlapping speech can also contaminate the models of individual speakers, decreasing the overall system performance. Alternatively, such segments may end up being assigned to a separate cluster, leading to the system mistakenly creating an additional "speaker".

Currently, many of the diarization systems found in literature still ignore the issue of overlapping speech altogether. Among the rest, some merely detect its presence in order to exclude such segments from the clustering process. Correctly identifying the exact speakers present in the overlapped regions is a more difficult task.

The detection of overlapping speech was chosen as one of the focus areas of this thesis, so a more detailed exploration of the topic can be found in chapter 8.

5.3 Initialization of Online Diarization

The last issue explored in this chapter is limited to online speaker diarization. More specifically, it deals with the lack of available data at the beginning of the diarization process, as alluded to in section 4.1.1.

Assuming that the speakers are not known in advance, the arguably most difficult part of online diarization concerns the detection of new speakers. Given a new speech segment, we need to decide whether it contains the speech of one of the speakers we have previously encountered, or whether it is an entirely new speaker. This is a non-trivial task in any situation, but is especially challenging at the very beginning of the audio stream when only the first speaker is known and their model is based on very limited data.

Similarly, if the system performs distance-based speaker change detection, this process itself can also involve a decision threshold which needs to be chosen correctly.

When faced with a new speech segment or a potential speaker change point, we then have to ask the question "Does this part seem different because there's a change of speakers, or is it only because of a slight change in tone or background?". With no basis for comparison, this may be extremely difficult to answer correctly.

There are, in essence, three possible approaches to this issue:

The one chosen by the authors of the systems described in section 4.3.2 is to obtain a model of each speaker in advance, thus completely circumventing the problem. However, this has its own drawbacks and is often simply not possible.

A second, very common option is to obtain a set of development data that is very similar to what the system is intended for. This can be used to set all relevant parameters such as decision thresholds to appropriate values. Such an approach works best if the development data were recorded under the exact same conditions as used by the live system (e.g. previous meetings in the same room and with the same microphones, previous episodes of a specific TV broadcast, etc.), but failing that, at least a similar type of recording is required.

Finally, if neither of these options are available, one may have to resort to a long initialization period at the start of the diarization process. During this time, the system will not provide speaker labels, but merely gather information in order to adapt to the specific data at hand. For instance, one may perform offline diarization on the first several minutes of a conversation in order to obtain the models of some of the speakers and to adjust the decision threshold for novelty detection. After this, the system can continue in an online manner.

The hybrid online-offline systems which were described in section 4.3.3 could also be considered an example of such an initialization - they perform offline diarization of past data and use the results for improving future online decisions.

Chapter 6

Main Goals of the Thesis

The introductory chapter of this thesis listed the following points as the initial aims of the research:

- Survey existing speaker diarization methods – both offline and online
- Identify some of the main challenges and obstacles in speaker diarization
- Create an overview of previous results found in literature, comparing the reported performance of individual systems
- Implement some of the described methods in a new diarization system and propose new methods or improvements
- Address one or more of the previously identified challenges

The first two points have already been covered by chapters 3 and 4, which presented an overview of different offline and online speaker diarization approaches, and chapter 5, which listed three problematic areas which were discovered during early stages of research – namely, very short speaker turns, overlapping speech, and the issue of initialization in an online diarization system.

With this background, it is now possible to solidify the rest of this initial framework into the specific goals of the thesis and their motivation.

Motivation and Goals of the Thesis

The diarization approaches presented in the previous chapters were divided into two main groups: offline and online. Offline systems process data retroactively, after an entire conversation has been obtained. Online systems, by contrast, operate in a sequential manner, typically processing an audio stream in real-time.

In the past, most of the research on speaker diarization has concentrated on offline approaches. While not rare, online systems are comparatively fewer in literature, likely due to the additional challenges posed within and the historically relatively low demand for applications which require real-time output.

However, this is slowly changing now. With increasing interest in real-time information extraction, voice-controlled devices and other forms of human-computer interaction, online speaker diarization is becoming more relevant and is starting to attract more attention than before.

One specific motivation for further research of online speaker diarization is also its potential use in automatic subtitling of television broadcasts. The information about changes of speakers, which is provided by a diarization system, could be used to improve the performance of a real-time ASR system, such as by allowing such a system to switch between different acoustic models when speakers change.

Finally, when the work in this thesis was first started, a significant majority of the existing online diarization systems was still based on the use of the traditional GMMs – a stark difference from offline diarization, where there already existed a greater variety of approaches, including the vastly popular i-vectors, as well as other less wide-spread options. Though the situation has since improved, online diarization still remains behind.

Given all of the above, it was decided in the initial thesis report (Kunešová, 2017) that some of the future work would go specifically towards online diarization, and particularly towards applying some of the more modern methods, such as i-vectors, to the task.

Yet at the same time, offline approaches are still in higher demand and organized evaluation campaigns and challenges likewise focus mostly on this topic. During the course of the doctoral study, participation in several collaborative projects lead to work on a variety of different tasks, including the development of both online and offline diarization. Thus, they are covered in this thesis in equal measure.

The above-mentioned thesis report also suggested a number of different options as the possible directions of further research. However, the field of speaker diarization has significantly evolved in the intervening time, and the state of the art has changed. It was therefore necessary to modify and update the original plans, as well as limit them to a smaller number of viable research directions.

For instance, the issue of very short speaker turns, which was one of the challenges discussed in chapter 5, seems to have lost much of its importance, as the newest state-of-the-art systems already appear to handle them relatively well. On the other hand, overlapping speech is still as problematic as ever, and represents a promising avenue of research.

In the end, based on the information which was presented in the previous chapters and the recent developments in the field, the final focus of the thesis is twofold: firstly, improving speaker diarization in general (both online and offline) and secondly, detecting overlapping speech, which is very relevant for diarization.

- Speaker diarization remains the primary topic of the thesis. In this regard, the main goal is to first implement a working diarization system based on existing methods, and then to propose further improvements.

  The specific methods concentrate on the most common scenario of single-channel speech with no additional sources of information (i.e. excluding multi-modal approaches).

  Between the needs of collaborative projects and independent research, as well as the rapidly evolving field of speaker diarization, several different ap-

proaches were explored: from GMMs, to i-vectors, and eventually x-vectors. Additionally, equal attention was given to both online and offline speaker diarization. To take advantage of this, a hybrid approach is also examined – one which employs offline methods in an online system. Such an approach also helps to address the issue of initialization in online diarization, which was one of the challenges discussed in chapter 5.

At the time the research was first started, i-vectors were the newest state of the art, and thus they became the focus of several of the experiments described in this thesis. Including, as originally planned, an i-vector based online diarization system. In the intervening time, the field has progressed to x-vectors and similar DNN embeddings, but in order to avoid switching in the middle of work, the experiments in this thesis have mostly continued using i-vectors. Nevertheless, x-vectors were eventually utilized as well, during work on the DIHARD Speaker Diarization Challenge, which will be covered in section 9.4.

- Overlapping speech is another one of the obstacles of speaker diarization which were discussed in chapter 5. It was found to have a significant effect on the performance of a diarization system, and as such, its detection was selected as the secondary focus of the thesis, alongside diarization itself.

  In regards to overlap detection, the main objective was to propose a single functional overlap detector oriented towards improving speaker diarization. In pursuit of this goal, it also became necessary to construct a new custom dataset for the training of this overlap detector.

Finally, besides the experiments and new or improved methods, the thesis also aims to provide an extensive overview of existing systems, with focus on comparing their reported performance.

This will be covered in chapter 7, which presents comparisons of many recent diarization systems and their reported results. A similar comparison of selected systems for detection of overlapping speech can be found in chapter 8.

Chapter 7

Evaluation of Speaker Diarization

The goal of this chapter is to provide an extensive comparison of existing diarization systems in regards to their reported performance. To facilitate this, the first section describes the standard evaluation metric which is most commonly used for this purpose, the Diarization Error Rate.

7.1 Diarization Error Rate

In order to evaluate the results of speaker diarization and compare the performance of different systems on a given set of testing data, a metric called Diarization Error Rate (DER) has been introduced by the National Institute of Standards and Technology (NIST). It was first used for their Rich Transcription evaluations (NIST, 2009) and now represents the de facto standard evaluation metric for speaker diarization systems.

DER is measured as the fraction of speaker time that is not correctly assigned to a speaker. It can be calculated using a script provided by NIST (`md-eval.pl`[1]), or using other implementations such the `pyannote.metrics` toolkit[2] (Bredin, 2017a).

The script first finds an optimal one-to-one mapping between the reference speaker IDs and the system output. The speech file is then split into segments at all speaker change points, including both reference and system labels.

DER itself is defined by NIST (2009) as

$$Error_{SpkrSeg} = \frac{\sum_{\text{all segments}} \{\text{dur}(seg) \cdot (\max(N_{\text{Ref}}(seg), N_{\text{Sys}}(seg)) - N_{\text{Correct}}(seg))\}}{\sum_{\text{all segments}} \{\text{dur}(seg) \cdot N_{\text{Ref}}(seg)\}}, \tag{7.1}$$

where for each segment seg,
$\text{dur}(seg)$ is the duration of seg,
$N_{\text{Ref}}(seg)$ is the number of reference speakers in seg,
$N_{\text{Sys}}(seg)$ is the number of "system speakers" (i.e. the clusters created by the system) in seg,
$N_{\text{Correct}}(seg)$ is the number of correctly matched speakers in seg.

[1] Part of the Speech Recognition Scoring Toolkit (SCTK), NIST (2018), Available from: https://github.com/usnistgov/SCTK
[2] Available from: https://pyannote.github.io

DER can also be expressed as the sum of three types of error rates: *missed speech* (which includes both speech incorrectly labeled as silence and missing speakers in segments with overlapping speech), *false alarm* (silence incorrectly labeled as speech, or excess speakers), and *speaker error* (SER) or (*speaker*) *confusion rate* (speech labeled as the wrong speaker), all calculated as a percentage of total speaker time.

Customarily, a forgiveness collar of 0.25 seconds is used around the reference speaker boundaries, in order to account for inconsistent annotation of speech and the difficulty of pinpointing the exact point when speech starts or ends.

By the original definition stated here, segments containing overlapping speech are counted multiple times, once for each speaker. In practice, however, these segments are sometimes excluded from the evaluation in systems which do not detect such occurrences, particularly in the case of those aimed at telephone speech (e.g., Sell and Garcia-Romero, 2014; Senoussaoui et al., 2014).

7.1.1 Other Evaluation Metrics

Besides DER, diarization performance can also be evaluated in terms of *purity* and *coverage*. These measures essentially express how well the individual speech frames are clustered between the reference speakers.

Purity measures how homogeneous the clusters are - a low value indicates that created clusters contain the speech of multiple speakers. *Coverage* expresses whether the speech of a specific speaker is fully contained within a single cluster (coverage = 100%), or split between multiple.

In the context of agglomerative clustering, high purity and low coverage would indicate that the clustering was stopped too early and the results are still too fractured. Low purity and high coverage would indicate the opposite.

With minor modifications, these measures can also be used to evaluate only the segmentation step of a diarization system.

Two different ways to calculate these measures appear in literature:

- In the `pyannote.metrics` evaluation toolkit, Bredin (2017a) defines the purity of a single cluster as the ratio between the number of frames which belong to the most common reference speaker in the cluster and the total number of frames in the cluster. Similarly, for one reference speaker, coverage is calculated as the ratio between the number of the speaker's frames in his/her most commonly assigned cluster and the total number of frames belonging to the speaker.

  The overall purity and coverage are then obtained as

$$\text{purity}(S, R) = \frac{\sum_k \max_j |s_k \cap r_j|}{\sum_k |s_k|} \tag{7.2}$$

  and

$$\text{coverage}(S, R) = \frac{\sum_j \max_k |s_k \cap r_j|}{\sum_j |r_j|} \tag{7.3}$$

where $S = \{s_1, \ldots, s_K\}$ is the set of clusters found by the system and $R = \{r_1, \ldots, r_J\}$ corresponds to the reference speakers (notation based on Manning et al., 2008, p. 357).

- Earlier works, such as (Gauvain et al., 1998; Ajmera et al., 2002; Kotti et al., 2008) defined cluster purity and coverage differently. Using the same notation as above, they can be expressed as

$$\text{avg. cluster purity}(S, R) = \frac{1}{N} \sum_k \sum_j \frac{|s_k \cap r_j|^2}{|s_k|} \tag{7.4}$$

and

$$\text{avg. cluster coverage}(S, R) = \frac{1}{N} \sum_j \sum_k \frac{|s_k \cap r_j|^2}{|r_j|} \tag{7.5}$$

where N is the total number of speech frames.

There are two main differences between these two versions of purity and coverage. First, the `pyannote.metrics` version can handle overlapping speech, while the second version's definition assumes that each speech frame belongs to only a single reference speaker.

Secondly, when calculating purity, Equation 7.2 only considers the most common speaker for each cluster, while Equation 7.4 counts all the speakers. Thus, the resulting values will be different. The difference in calculating coverage is analogous.

Purity and coverage were mainly used for evaluation in older systems, before the popularization of DER. In most recent publications, speaker diarization is evaluated solely on the basis of DER or its components. For this reason, purity and coverage are not listed in the overview of state of the art in this chapter, nor used for evaluating the experiments in chapter 9.

7.2 Overview of the State of the Art

This section presents a comparison of the reported performance of the diarization systems published in the last decade. Offline systems are grouped by domain: telephone speech, conference meetings, and radio or television broadcasts, with a separate table focusing on the results of the recent DIHARD diarization challenge. Online systems, which are considerably fewer, are listed in a single table at the end.

Note: For the most part, the listed error rates of individual systems should not be directly compared with each other, as they were often obtained under different evaluation conditions or even on different subsets of the same corpus. The only exceptions are where the results come from an official evaluation campaign, such as the DIHARD Challenge in Table 7.6.

7.2.1 Telephone Speech

In telephone speech diarization, the traditional evaluation dataset is the CALLHOME corpus, a multilingual corpus of spontaneous telephone conversations. Table 7.1 shows an overview of some of the most notable offline systems which have been evaluated on this corpus or its parts.

While the CALLHOME corpus itself includes conversations between up to 7 speakers, some systems are explicitly limited to 2-speaker conversations only. For this reason, the results are presented here in two columns: one for error rates reported for the 2-speaker scenario and the other for those achieved with variable number of speakers.

Additionally, some systems used the so-called "NIST SRE 2000 CALLHOME" (LDC2001S97, disk 8), which includes 6 different languages, while others were limited to different subsets of the corpus, such as only the American English recordings (Canavan et al., 1997).

Table 7.1: Comparison of recent diarization systems aimed at telephone speech, listing results on the CALLHOME corpus or its parts. Unless otherwise stated, all systems ignored regions of overlapping speech during evaluation and the errors were computed with oracle VAD – therefore the values essentially represent only the speaker error component of DER. With the exception of [6], [8], [9] and [12], all systems are strictly offline.

		Error [%]	
Sys.	Description	2 speakers	any number
[1]	speaker factors + eigenvoices; VAD & overlaps unspecified	8.7[a]	13.7[b]
[2]	i-vectors + spectral clustering		
	a) known number of speakers	5.2[c]	8.9[c]
	b) estimated number of speakers	13.9[c]	14.4[c]
[3]	i-vectors + PCA + iterative optimization	14.6[c]	14.5[b,c]
[4]	i-vectors + normalization + PCA, unspecified VAD	7.5	12.1[b]
[5]	i-vectors with DNN UBM	4.7	10.3
[6]	a) i-vectors + normalization, 2 speakers only, online	3.2[d]	N/A
	b) above system, offline	2.1[d]	
[7]	DNN speaker embeddings → AHC	–	12.8
	+ VB resegmentation	–	9.9
[8]	ASR → speaker change detection between words,	(SER / DER)[e]	
	i-vec+WCCN → online X-means clustering	4.86 / 16.23	4.82 / 16.24
	+ retroactive label updates	3.02 / 14.39	3.24 / 14.66

(continued on the next page)

Table 7.1: (continued)

Sys.	Description	Error [%] 2 speakers	Error [%] any number
[9]	CALLHOME Am. Eng., performed VAD	(SER)	(SER / DER)
	i-vectors, spectral clustering (offline)	–	14.59 / 20.54
	i-vectors, "links" clustering (online)	–	25.40 / 31.36
	d-vectors, spectral clustering (offline)	5.97	6.03 / 12.48
	d-vectors, "links" clustering (online)	–	11.02 / 17.47
	entire CALLHOME – d-vectors, spectral	–	12.0 / 18.8
[10]	baseline: i-vectors + PLDA	–	17.6
	triplet network speaker embeddings	–	12.7
[11]	Bayesian HMM & VB, random init.	–	12.0
	random init. x5	–	9.0
	i-vector/PLDA & VB, AHC init. (baseline)	–	9.7
[12]	system from [9] + updated d-vectors extraction		(SER)
	spectral clustering (offline)	–	8.8
	UIS-RNN clustering (online)	–	7.6
[13]	Bi-directional LSTM, spectral clustering		
	a) i-vectors + LSTM + spect. cl.	–	8.53
	b) x-vectors + LSTM + spect. cl.	–	7.73
	c) weighted sum of a), b)	–	6.63
[14]	CALLHOME Am. Eng., VAD, ASR; overlaps unspecified	(SER / DER)	
	x-vectors + word embeddings from ASR→ spect. clust. with word-level speaker turn probabilities	1.73 / 6.03	2.9 / 6.97
[15]	End-to-end DNN system: MFCC → speaker labels; includes VAD and overlapping speech	8.50	15.29
[16]	x-vectors + PLDA, AHC; Kaldi VAD? (implied)	–	8.00
	x-vectors → deep self-supervised AHC clustering	–	8.26
	fusion system	–	7.38

[1] Castaldo et al. (2008)
[2] Shum et al. (2012)
[3] Shum et al. (2013)
[4] Senoussaoui et al. (2014)
[5] Sell et al. (2015)
[6] Zhu and Pelecanos (2016)
[7] Garcia-Romero et al. (2017)
[8] Dimitriadis and Fousek (2017)
[9] Wang et al. (2018)
[10] Song et al. (2018)
[11] Diez et al. (2018a)
[12] Zhang et al. (2019)
[13] Lin et al. (2019)
[14] Park et al. (2019a)
[15] Horiguchi et al. (2020)
[16] Singh and Ganapathy (2020)

a "Segmentation error" obtained with NIST's script *seg_scoring.v2.1.pl* (https://www.nist.gov/document/segscr-v21tgz)
b Value reported in (Sell et al., 2015)
c Estimated from plots separated by number of speakers, total err. was calculated as a weighted average, based on the number of files in each group
d Sys. directly uses ref. transcripts for segmentation
e System performs VAD with a relatively high miss rate ($\sim$ 11% miss, $\sim$ 0.5% FA). Overlaps not mentioned.

It should be noted that the error rates shown in this table do not represent the DER as defined by NIST, but more closely resemble only the speaker error component. In telephone speech diarization, or at least on the CALLHOME corpus,

it appears to be an accepted practice to make certain changes in the evaluation procedure. Namely, the voice activity detection step is customarily replaced by information taken directly from reference transcripts, presumably to prevent the choice of VAD algorithm from influencing the performance of the rest of the system. Additionally, regions of overlapping speech are typically excluded during the final calculation of a system's error rate. This differs from the systems listed in the later sections of this chapter, most of which use the standard definition of DER, as described in section 7.1.

Additionally, the resulting error rate is not entirely equivalent to the speaker error (SER) of a system with a proper VAD – in such systems, a high miss rate may mean that the most problematic regions are excluded, making the rest easier to correctly process.

Besides the achieved error rates, Table 7.1 also summarizes the basic methods employed by each system. One may notice that the majority of these systems employ i-vectors (or, in one case, the closely related speaker factors), but the most recent works have replaced this with DNN-based speaker embeddings.

7.2.2 Meeting Data

For the conference meetings scenario, the best-known evaluation data come from the Rich Transcription (RT) Evaluations[3], which were organized by NIST, most recently in 2009. Table 7.2 presents an overview of some of the diarization systems which were evaluated on the RT datasets, and their achieved results.

The RT meeting datasets contain recordings obtained with the use of multiple microphones (typically several distant microphones, microphone arrays and individual head microphones worn by the speakers) and the official evaluation tasks included several options with different input configurations. Where available, table Table 7.2 lists the results corresponding to the two most common options: the multiple distant microphones (MDM) and single distant microphone (SDM) scenarios.

While the systems listed in the table used a wide range of approaches, they all share a common aspect: for the MDM scenario, all systems employed the BeamformIt toolkit (Anguera et al., 2007), an acoustic beamforming tool which can be used to transform multiple input channels into a single enhanced speech signal. Additionally, with the sole exception of (Bozonnet et al., 2010), all MDM systems also used time difference of arrival (TDOA) as additional features.

From the listed error rates, one may observe that the MDM systems consistently score better than their SDM counterparts. This shows that when available, the additional information obtained from multiple sound sources is helpful in improving diarization performance.

Outside of organized evaluations, some more recent publications have listed results on another notable dataset – the AMI Meeting Corpus. These are shown in Table 7.3. The AMI corpus was also used for some of the experiments shown in

[3]Rich Transcription Evaluation,
https://www.nist.gov/itl/iad/mig/rich-transcription-evaluation

this thesis, and a more detailed description is included in section 9.1.3. The listed results on the AMI corpus may not be directly comparable with each other, as different authors appear to use different subsets of the corpus or different audio channels (recorded via distant microphones or individual headsets) and there are also multiple different sets of reference transcripts in existence.

Table 7.2: Comparison of offline diarization systems for conference meetings, evaluated on NIST RT datasets under MDM or SDM conditions. In addition to the listed methods, all MDM systems also performed beamforming with the use of the BeamformIt toolkit and with the sole exception of [6], all employed time delay features. Results marked with a cross in the "Overlaps scored" column correspond to error rates calculated without regions of overlapping speech.

System	Dataset	Description	Multi-channel	Overlaps scored	DER [%]
[1]	RT-07	Results from multiple evaluation participants	✗	✓	11.6
			✓	✓	7.5
	RT-09	Results from multiple evaluation participants	✗	✓	17.7
			✓	✓	10.1
[2]	RT-09	BIC-based bottom-up clustering	✗	✓	29.0
			✓	✓	21.5
[3]	RT-09	Viterbi-based initialization (30 clusters), then bottom-up clustering, resegmentation	✗	✓	16.0
			✗	✗	10.7
		Quantized TDOA → 9 initial clusters, then bottom-up clustering, resegmentation	✓	✓	9.2
			✓	✗	3.8
[4]	RT-09	Fundamental freq. est., BIC-based bottom-up clust.	✓	✓	21.4
[5]	RT-09	Ergodic HMM/GMM, BIC-based top-down clustering	✗	✓	34.5
			✓	✓	32.0
[6]	RT-09	Evolutive HMM (top-down approach)	✗	✓	21.1
			✗	✗	16.0
			✓	✓	20.3
			✓	✗	15.2
[7]	RT-09	BIC-based bottom-up clustering with added prosodic features	✗	✓	31.3
			✓	✓	17.2
[8]	RT-06	i-Vectors + information bottleneck	✓	✓	20.4[(a)]
	RT-09		✓	✓	21.3[(a)]

[1] NIST Rich Transcription Evaluations participants
Values represent the average of the best results for each recorded meeting among four evaluation participants, compiled by Anguera et al. (2012)
[2] Huijbregts et al. (2009)
[3] Nguyen et al. (2009)
[4] Pardo et al. (2009)
[5] Luque and Hernando (2009)
[6] Bozonnet et al. (2010)
[7] Friedland et al. (2012)
[8] Madikeri et al. (2015)
a Reported value is only the Speaker Error Rate.

Table 7.3: Comparison of diarization systems for conference meetings, evaluated on the AMI Corpus or its parts. Individual systems may have been evaluated on different subsets of the corpus or using slightly different ground-truth references.

System	Dataset	Description	Multi-channel	Overlaps scored	DER [%]
[1]	AMI (IS)	BIC-based AHC	✓	✓	30.7
		+ HMM-based overlap detector (see [1] in Table 8.1)			29.2
	AMI (all)	BIC-based AHC	✗	✓	32.8
		+ HMM overlap detector			32.0
		+ oracle overlap detection			23.7
[2]	AMI (IS)	GPU version of system [7] in Table 7.2	✓	✓	32.1
		+ online: offline process performed every 2.5 s			32.0
[3]	AMI	uniform initialization into 20-55 clusters → Viterbi → BIC-based AHC, + HMM overlap detection	✓	✓	22.8
		+ real VAD			29.6
[4]	AMI	online: i-vectors + sequential clustering with a threshold	✗	?	34.2
[5]	AMI	ref. VAD, i-vec. + PLDA, AHC, no reseg.			
		a) single distant microphone	✗	✗	12.8
		b) multiple distant microphones	✓	✗	7.6
		c) headset mix (single channel)	✗	✗	4.8
[6]	AMI	ref. VAD, Viterbi segmentation → BIC or i-vec clustering			
		a) base GMM/BIC system (MFCC only)	✗	?	24.0
		b) i-vectors + PLDA – MFCC only			23.0
		c) i-vectors + PLDA – MFCC + long-term features			18.2
[7]	AMI	LSTM overlap detection, overlap-aware VB-HMM resegmentation			
		baseline (no VB resegmentation)	✗	✓	29.7
		+ VB-HMM resegmentation			28.9
		+ overlap assignment			23.8
[8]	AMI	SincNet features → x-vectors → compositional speaker embeddings for identifying speakers in overlaps	✗	✓	26.0
		+ a dedicated overlap detector			22.9
[9]	AMI	overlap detection, overlap-aware clustering, combination of multiple systems (DOVER-Lap method)			
		a) VB-HMM, top two speakers assigned in overlaps	✗	✓	21.5
		b) overlap-aware spectral clustering (Raj et al., 2020a)			23.6
		c) region proposal networks (Huang et al., 2020)			25.5
		combined outputs - DOVER-Lap approach			20.5

[1] Boakye et al. (2008b) [2] Friedland (2012)
[3] Zelenák et al. (2012) [4] Patino et al. (2018b); overlap scoring not explicitly mentioned
[5] Maciejewski et al. (2018) [6] Zewoudie et al. (2018); best scores among all feature combinations; overlap scoring not explicitly mentioned
[7] Bullock et al. (2020)
[8] Li and Whitehill (2020) [9] Raj et al. (2020b)

7.2.3 Radio and Television Broadcast

For the broadcast news domain, there have been several evaluation campaigns. Some of the most notable ones include the ESTER 2 (Galliano et al., 2009), REPERE (Galibert and Kahn, 2013) and ETAPE (Galibert et al., 2014) campaigns, all of which focused on French-language radio or television broadcasts.

Specifically, the ESTER 2 dataset contains radio broadcast news and debates, the REPERE dataset consists of television news and debates and the ETAPE dataset includes television news, debates and entertainment shows, as well as various radio shows.

Table 7.4 shows an overview of some of the diarization systems which were evaluated on these datasets, including official evaluation results. However, some of the latter appear to have only been made public in an anonymized form, and as such, the full specifications of the participating systems are not available. Nevertheless, the achieved error rates can serve for comparison with the other systems.

The Albayzin series of speech evaluations has also included speaker diarization tasks, most recently in 2018 as the "IberSpeech-RTVE 2018 Challenge" (Lleida et al., 2019). The 2018 challenge dataset contained a variety of Spanish language TV broadcasts by the Spanish Television (RTVE). The official results of participating teams are shown in Table 7.5.

Table 7.4: Comparison of recent offline diarization systems aimed at TV broadcast, evaluated on the datasets from three different French evaluation campaigns. Included are also official evaluation results, most of which are available only in anonymized form, without system details.

System	*Dataset*	*Description*	*DER [%]*
[1]	ESTER 2	Official evaluation results (details not specified):	
		Team IRIT	14.0
		Team LIA	15.1
		Team LIG	10.9
		Team LIMSI	12.4
		Team LIUM (the same system as [4])	10.8
[2]	REPERE	Official evaluation results (anonymized):	
		Team "A"	13.7
		Team "B"	13.4
		Team "C" (most likely the same system as [4])	11.1
[3]	ETAPE	Official evaluation results – all 7 participants	15.61–28.70[(a)]
[4]	ESTER 2 ETAPE	BIC-based clustering, resegmentation, followed by CLR-based clustering	10.8 18.9
	REPERE ETAPE	BIC-based clustering, resegmentation, then ILP + i-vector clustering	11.1 18.5

(continued on the next page)

Table 7.4: (continued)

System	*Dataset*	*Description*	*DER [%]*
[5]	ESTER 2	a) CLR-based agglomerative clustering	9.6
		b) Gaussian supervectors + SVM classifiers	12.5
		c) combination of both above approaches	8.3
[6]	ETAPE	Exploiting similarities between different episodes of the same show + LIUM diarization toolkit ([4])	16.2
[7]	ETAPE	a) initial BIC-based clustering, then iterative recluster-ing + resegmentation	23.8
		b) above system + overlap detection	17.6
[8]	REPERE	binary keys	15.2
[9]	(Custom A)	best baseline: i-vectors → PLDA or cosine distance →	8.5
	(Custom B)	AHC or Connected Components (CC) clustering	9.9
	(Custom A)	i-vectors → triplet network embeddings → CC	7.9
	(Custom B)	clustering	9.6
[10]	ESTER 2	S4D Speaker Diarization Toolkit: Gaussian divergence	6.2
	ETAPE	+ BIC segmentation; BIC + AHC or i-vectors + ILP;	15.6
	REPERE	HMM/Viterbi resegmentation	9.2
[11]	ETAPE	almost fully RNN-based system: LSTM VAD, SCD, speaker embeddings and resegmentation	
		a) clustering: AHC	28.5
		b) clustering: Affinity Propagation	24.2
		baseline: S4D Toolkit (see [10])	24.5

[1] ESTER 2 Evaluation participants (Galliano et al., 2009)
[2] REPERE Evaluation participants (Galibert and Kahn, 2013)
[3] ETAPE Evaluation participants (Galibert et al., 2014)
[4] LIUM_SpkDiarization Toolkit (Rouvier et al., 2013), results adapted from Meignier et al. (2013)
[5] Le et al. (2010)
[6] Khemiri et al. (2013)
[7] Charlet et al. (2013)
[8] Delgado et al. (2015)
[9] Le Lan et al. (2017), Data = custom mix of REPERE, ETAPE and ESTER
[10] Broux et al. (2018)
[11] Yin et al. (2018), evaluated without a collar and including overlapping regions
a Results reported in the paper were anonymized and no system details were specified. Most participants provided multiple submissions – the listed range corresponds to the best scoring system configuration from each participant.

Table 7.5: Comparison of diarization systems participating in the Albayzin 2018 Evaluation / IberSpeech-RTVE 2018 Challenge. Column "Closed-set?" indicates if systems competed in the closed-set category (trained using only specific challenge-provided data) or open-set category (external training data were allowed). Listed DER represents the final results on the evaluation set, as published in the official paper (Lleida et al., 2019)

System	*Team*	*Description*	*Closed-set?*	*DER [%]*
[1]	G1-GTM-UVIGO	x-vectors, two-stage segmentation and "Chinese Whispers" clustering, music detection	✗	11.4
[2]	G11-ODESSA	C1c: uniform seg., binary key + AHC	✓	30.2
		C2c: BiLSTM SCD, triplet-loss embeddings, affinity propagation clustering	✓	37.6
		fusion: C1c + C2c	✓	26.6
		C1o: uniform seg., x-vectors + AHC	✗	20.3
		C2o: BiLSTM SCD, triplet-loss embeddings + AHC	✗	36.7
		fusion: C1o + C2o + C1c	✗	25.9
[3]	G20-STAR-LAB	DNN speaker embeddings → initial clusters, then VB with DNN bottleneck based i-vector subspaces	✗	30.8
[4]	G21-EMPHATIC	CNN & LSTM speaker embeddings, PCA, spectral clustering, HMM reseg.	✗	31.0
[5]	G22-JHU	fusion: x-vectors + PLDA & x-vectors + PLDA	✗	28.2
			✓	39.1
[6]	G4-VG	S4D toolkit ([10] in Table 7.4): BIC segmentation, AHC, HMM reseg.	✓	25.4
[7]	G8-AUDIAS-UAM	DNN embeddings (BiLSTM)	✓	31.4
		total variability	✓	28.7
[8]	G10-VIVOLAB	i-vectors + PLDA, unsupervised PLDA adaptation	✓	17.3
[9]	G19-EML	speaker vectors based on GMM supervectors + WCCN & LDA, mean-shift based clustering	✓	26.6
[10]	[*later publication*]	online system: "speaker-corrupted" DNN embeddings, sequential clustering	–	27.7

[1] Ramos-Muguerza et al. (2018) (paper describes a multimodal system by the same team)
[2] Patino et al. (2018c)
[3] Castan et al. (2018)
[4] Khosravani et al. (2018b)
[5] Huang et al. (2018)
[6] E. L. Campbell et al. (2018)
[7] Lozano-Diez et al. (2018)
[8] Viñals et al. (2018b)
[9] Ghahabi and Fischer (2018)
[10] Ghahabi and Fischer (2019) (later publication)

7.2.4 The DIHARD Speech Diarization Challenge

This section presents an overview of the official results of the recent First and Second DIHARD Speech Diarization Challenge[4] (DIHARD I and DIHARD II). The DIHARD evaluation series, which first took place in 2018, focuses on challenging recordings from a variety of domains. The original DIHARD I evaluation dataset included recordings from 10 different corpora and was expanded with additional data for DIHARD II.

Table 7.6 lists the results achieved by all participating teams in the first run of the challenge, as well as a very brief summary of the published system description.

Table 7.7 similarly shows the participants of DIHARD II. However, system descriptions from this run have never been published and as such, the details of most systems are missing.

Our team has also participated in the challenge, and our results appear in the two tables as Zajíc et al. (2018) and Zajíc et al. (2019). Further details about the challenge, as well as a description of our system, will be presented in section 9.4 of chapter 9.

[4] https://dihardchallenge.github.io/dihard1/ and .../dihard2/

Table 7.6: Comparison of diarization systems participating in the First DIHARD Speaker Diarization Challenge (best system from each team, except team CPqD). DER values correspond to the official challenge results - evaluated without any tolerance collar and including overlapping regions. Track 1 and Track 2 correspond to results with and without reference speech labeling.

			DER [%]	
	Team	Description	Track 1	Track 2
[1]	JHU	x-vectors + AHC, VB resegmentation	23.73	37.19
[2]	USTC-iFLYTEK	denoising, initial BIC-based clustering → i-vectors + PLDA, resegmentation	24.56	36.05
[3]	BUT	dereverb., x-vectors + AHC, VB resegm.	25.07	35.51
[4]	ViVoLab	BIC segm., i-vectors + PLDA, Variational Bayes	26.02	38.00
[5]	ZCU-NTIS	domain classification, i-vectors + AHC (domain-dependent threshold)	26.90	45.78
[6]	STAR-LAB	domain classification → VAD settings, bottleneck DNN features → i-vectors, VB diarization	27.61	41.56
[7]	LEAP	i-vectors + PLDA, AHC (threshold)	28.52	–
[8]	BISC	binary keys	29.33	–
[9]	CPqD	a) DNN-based VAD + LIUM Toolkit (Rouvier et al. (2013), [4] in Table 7.4)	32.76	41.17
		b) DNN: log spectrum → DNN embeddings + VAD + overlap; then k-means clustering	40.94	48.85
[10]	SAIVT	i-vectors + PLDA, AHC (threshold)	33.15	57.14
[11]	CDS	KL2 segmentation → 2x (Viterbi reseg. + BIC) → gender id., spk. id. based clust.	33.79	52.38
[12]	IntelligentVoice	LSTM speaker embed., PCA, spectral clust.	36.73	–
–	SINICA	[*did not provide a system description*]	37.46	–
–	–	baseline (same label for all data)	39.14	68.48

[1] Sell et al. (2018)
[2] Sun et al. (2018b)
[3] Diez et al. (2018b)
[4] Viñals et al. (2018a)
[5] Zajíc et al. (2018)
[6] McLaren et al. (2018)
[7] Ganesh et al. (2018)
[8] Patino et al. (2018a)
[9] Miasato Filho et al. (2018)
[10] Himawan et al. (2018)
[11] Gupta and Alam (2018)
[12] Khosravani et al. (2018a)

Table 7.7: Comparison of diarization systems participating in the Second DIHARD Speaker Diarization Challenge (best system from each team). DER values correspond to the official challenge results at the end of Phase 2 - evaluated without any tolerance collar and including overlapping regions. Track 1 and Track 2 correspond to results with and without reference speech labeling. Individual system descriptions were never officially made public.

			DER [%]	
	Team	Description	Track 1	Track 2
[1]	BUT	x-vectors, AHC, Variational Bayes HMM	18.42	27.11
[2]	DKU_LENOVO	overlap detection, ResNet-LSTM VAD, ResNet spk. embeddings, LSTM similarity scoring, spectral clustering, VB	18.84	27.90
–	YD_lab	[*description not available*]	20.75	–
[4]	DI-IT	DNN-based speaker embeddings, recording environment classifier, AHC (environment-specific threshold)	20.83	33.45
–	nelslip	[*description not available*]	21.03	35.75
[6]	LEAP	i-vec. + x-vec., weighted average of PLDA scores, AHC, then a modified VB-HMM	21.90	42.69
–	ty	[*description not available*]	22.62	48.56
[8]	Speed	domain type classification (4 groups), speech enhancement, LSTM VAD, x-vectors + AHC, LSTM+GMM resegmentation	22.82	31.03
[9]	USC_SAIL	DNN embeddings, spectral clustering, overlap detection, Viterbi resegmentation	22.89	46.72
–	THS	[*description not available*]	23.31	–
[11]	UWB-NTIS	domain classification, i+x-vectors, AHC or PLDA+k-medoids (domain-specific settings)	23.47	–
–	PDL	[*description not available*]	23.50	31.04
–	JHU	[*description not available*]	23.70	–
–	HUM	[*description not available*]	24.97	–
[15]	VIVOLAB	i-vectors, tree-based sequential clustering	25.02	37.22
–	Elektronika	[*description not available*]	25.37	45.03
		DIHARD II baseline system		
–	(baseline)	x-vectors + PLDA, AHC	25.99	50.12
		+ denoising	–	40.86
–	IITB_DAPLAB	[*description not available*]	26.33	–
–	CSTR-Edinburgh	[*description not available*]	26.49	–
–	lizeqian	[*description not available*]	30.78	–
–	Shazam	[*description not available*]	63.35	–
[21]	(*later work*)	End-to-end DNN sys.: MFCC → spk. labels	–	32.59

[1] Landini et al. (2020)
[2] Lin et al. (2020)
[4] Novoselov et al. (2019)
[6] Singh et al. (2019)
[8] Sahidullah et al. (2019)
[9] Park et al. (2019b)
[11] Zajíc et al. (2019)
[15] Viñals et al. (2019)
[21] Horiguchi et al. (2020) (non-participants, see also in Table 7.1)
baseline: https://github.com/iiscleap/DIHARD_2019_baseline_alltracks

7.2.5 Online Diarization

For online diarization, there appears to be no single universally accepted evaluation standard or baseline, as different authors focus on different target domains and conditions. Nevertheless, Table 7.8 aims to provide a basic overview of some of the more notable examples of online systems and their specific features.

Several of these systems were evaluated on datasets which were discussed in the previous sections – namely the CALLHOME corpus, AMI corpus, and data from the NIST RT and Albayazin 2018 evaluations. Besides these, the following also appear in the table: 1996 HUB-4 (English-language broadcast news database), BNSI (Slovenian broadcast news database) and TC-STAR 2006 and 2007 (European Parliament plenary speeches).

Table 7.8: Comparison of online diarization systems. Dataset types are telephone (T), broadcast news (B), conference meetings (M) and plenary speeches (P). Listed latency corresponds to the length of speech segments or decision windows and does not include processing time or other factors.

System	Dataset	Type	Description	Unknown spk.	Online	Latency [s]	DER [%]
[1]	TC-STAR	P	GMM, sequential clustering with incremental EM model adaptation, gender dependent decision threshold	✓	✓	1.0 2.0 3.0 4.0 5.0	11.9 8.3 6.1 5.4 5.3
[2]	BNSI HUB-4	B B	NCLR-based sequential clustering	✓ ✓	✓ ✓	up to 20.0	20.5 17.3
[3]	HUB-4	B	GMM, sequential clustering with MAP model adaptation	✓	✓	1.0 2.0 3.0	42.2[(a)] 45.2 39.1
[4]	RT-09	M	GMM, hybrid system: parallel offline bottom-up and online speaker ID	✓ ✓	✓ ✗	2.5[(b)] N/A	37.8 18.1
[5]	RT-09	M	a) GMM-based speaker ID b) above system + MDM	✗	✓	2.5 2.5	44.6 39.3
[6]	AMI (IS)	M	GPU version of [7] in Table 7.2 + online = offline diar. every 2.5 s	✓ ✓	✗ ✓	N/A 2.5	32.1 32.0
[7]	HUB-4	B	GMM, sequential clustering with MAP model adaptation	✓	✓	0.3 0.5 1.0 2.0 3.0 4–7	57.0[(c)] 50.0 45.0 44.0 40.0 ~ 42

(continued on the next page)

Table 7.8: (continued)

System	Dataset	Type	Description	Unknown spk.	Online	Latency [s]	DER [%]
[8]	CALLHOME	T	i-vectors + normalization, only 2 speakers	✓	✓	∞[d]	3.2[e]
				✓	✗	N/A	2.1
[9]	RT-07, 09	M	GMM, speaker ID with only a small amount of initial data and gradual adaptation	✗	✓	3.0	18.9
						5.0	16.3
						7.0	15.3
[10]	CALLHOME	T	hybrid sys.: ASR $\rightarrow$ possible speaker turns only between words, BIC segm., i-vec. + WCCN $\rightarrow$ repeated X-means clust.	✓	✓	?	16.2
			+ retroactive label updates	✓	✓	∞	14.7
[11]	AMI	M	i-vectors + sequential clustering	✓	✓	3.0	34.2
[12]	(Custom)	M	multiple micr. $\rightarrow$ direction of arrival	✓	✓	~ 0	15.4
[13]	CALLHOME (AmEng)	T	LSTM spk. embeddings: d-vectors				
			i-vectors, spectral clust. (offline)	✓	✗	N/A	20.5
			i-vectors, "links" clust. (online)	✓	✓	0.4	31.4
			d-vectors, spectral clust. (offline)	✓	✗	N/A	12.5
			d-vectors, "links" clust. (online)	✓	✓	0.4	17.5
[14]	CALLHOME	T	[13] with updated d-vector extraction. err. = SER only				
			spectral clustering (offline)	✓	✗	N/A	8.8
			UIS-RNN clustering (online)	✓	✓	1.6	7.6
[15]	Albayzin 2018	B	"speaker-corrupted" DNN embeddings, sequential clustering	✓	✓	2.0	27.69

[1] Markov and Nakamura (2008)
[2] Grašič et al. (2010)
[3] Geiger et al. (2010)
[4] Vaquero et al. (2010)
[5] Friedland et al. (2012)
[6] Friedland (2012)
[7] Soldi et al. (2015)
[8] Zhu and Pelecanos (2016)
[9] Soldi et al. (2016)
[10] Dimitriadis and Fousek (2017), see also in Table 7.1
[11] Patino et al. (2018b)
[12] Ito et al. (2018)
[13] Wang et al. (2018), see also in Table 7.1
[14] Zhang et al. (2019)
[15] Ghahabi and Fischer (2019)

a Error was measured as a custom "misclassification rate", rather than DER. Listed values are the average of results on two different evaluation files.
b After an initialization period of 60 s
c Error rates for this system were estimated from a plot, some values were omitted.
d Segments correspond to entire utterances with no set maximum length (the average is 2.1 s)
e System uses reference transcripts in place of segmentation and excludes overlapping speech from evaluation, both of which reduce the resulting error rate

The table also lists the latency of each system. Some authors have provided results for multiple different latency settings and we can observe that this attribute greatly affects the system performance. Very short latencies in particular generally correspond to significantly larger error rates.

It is difficult to make comparisons between these systems, as there are great differences in the type of data, system latency, the amount of prior knowledge and even evaluation criteria, all of which affect system performance.

7.2.6 Summary

The previous sections provided an overview of recent state-of-the-art diarization systems aimed at different domains and their reported results on a variety of different datasets.

From the values presented in the tables, we can obtain the approximate range of error which can be achieved in different areas of speaker diarization. However, it is difficult to determine which approaches are best, as there are often significant differences in the evaluation conditions, particularly in cases of different datasets or even different domains. This includes aspects such as the sound quality, the level of noise, the amount of overlapping speech and the length of the individual utterances in each dataset, all of which can greatly affect the final system performance.

The results for telephone speech in particular cannot be directly compared to those from the other domains, as most of the values in Table 7.1 were obtained using different evaluation criteria than the standard DER and as such, are likely significantly lower than they would have been otherwise.

Nevertheless, we can at least observe that many of the online systems shown in Table 7.8 achieve very large error rates, greatly exceeding those of seemingly comparable offline systems. This illustrates the relative difficulty of online diarization.

Finally, if we look at the approaches used by each system and their years of publication, we can see how drastically the state of the art has shifted over a relatively short time. While a decade ago, everyone still used GMMs and BIC-based clustering (e.g. most of the systems in Table 7.2), by 2015 i-vectors became standard for offline diarization, with only online systems lagging behind. Then a mere few years later, x-vectors and other DNN-based speaker embeddings came to the forefront. And now, the first end-to-end neural systems are making their appearance. It is almost certain that there will be more of them in the near future.

Chapter 8

Overlapping Speech

One of the main issues in speaker diarization which were discussed in chapter 5 was overlapping speech. This chapter expands upon that topic by providing a more detailed introduction to the problem as well as an overview of the standard techniques for overlap detection and a comparison of notable systems. Related experiments will be found in section 9.5.

8.1 Introduction

The term "overlapping speech" refers to instances during which two or more individuals speak at the same time. This is a very common occurrence in any sort of natural conversation: it includes situations such as speakers interrupting each other, one participant offering backchannel responses to the active speaker (e.g. "yeah", "uh-huh"), or simply brief natural overlaps during rapid turn-taking.

There have been multiple works analyzing the amount of overlaps in conversational speech. Ten Bosch et al. (2005) found overlaps to be present in 44 % of all speaker changes in face-to-face dialogues and in 52 % in the case of telephone speech. Similarly, Heldner and Edlund (2010) reported a value of 40 % on their data, with a median duration of 470 ms in such overlaps.

As previously discussed in chapter 5, such overlapping speech can have multiple negative effects on the accuracy of speaker diarization: In terms of DER, undetected overlaps directly contribute to missed speech rate. Incorrectly labeled overlapping speech can also contaminate the models of individual speakers, decreasing the overall system performance. Alternatively, such segments may end up being assigned to a separate cluster, leading to the system mistakenly creating an additional "speaker".

The effect on DER can be very large: Huijbregts and Wooters (2007) found that "the lack of ability to model overlapping speech [was] the source of 22 % of the total diarization error rate" of their system, while Boakye et al. (2008b) reported up to 27.6 % relative improvement in DER when using oracle overlap labels.

A similar result was also obtained during one of the experiments described in this thesis (section 9.4.7, also published in Zajíc et al., 2019) – on the development set of the DIHARD II corpus, the addition of overlap handling with ground-truth overlap labeling decreased the DER of our system from 20.78% to 16.16% (22% relative improvement).

These observations all suggest that automatic detection of overlapping speech

has great potential for improving the performance of speaker diarization. For this reason, it was chosen as one of the subtopics of this thesis.

8.2 Detection of Overlapping Speech

Common methods for detecting overlapping speech follow one of two different directions: Until relatively recently, the best systems have relied on combining a variety of carefully selected hand-crafted features. However, the current boom in deep learning has also lead to the appearance of DNN-based approaches. These usually rely on the neural network itself to extract the most relevant information from unprocessed data, often with equal or better results.

8.2.1 Overlap Detection Using Hand-crafted Features

The more traditional approach to overlap detection is based on the use of HMM and Viterbi decoding, typically modeling three classes: non-speech, single-speaker speech and overlapping speech. A variety of different features are used for the purpose.

One very common choice is LPC residual energy. LPC analysis estimates the formants of a speaker, using them to model the spectrum of the speech. The residual energy is then the difference between the linear approximation and the spectrum. The use of LPC for overlap detection arises from the assumption that LPC can model the speech of a single speaker reasonably well, but fails in the presence of multiple different speakers, resulting in a larger residual energy. Systems which use this residual energy for overlap detection include (Zelenák et al., 2012) and (Boakye et al., 2008a).

Boakye et al. (2008a; 2008b) also explore the use of various other features such as spectral flatness or harmonic energy ratio in a HMM-based overlap detector.

Charlet et al. (2013) propose two different approaches. One of these is based on the standard HMM with models for overlapped and non-overlapped speech, while the other relies on multi-pitch detection using spectral combs.

If multiple sound sources are available, it is also possible to use this additional information for detecting simultaneous speech. This is used by e.g. Pfau et al. (2001) and Pardo et al. (2006).

In (Zelenák et al., 2012), overlapping speech is detected by a combination of the HMM-based approach (with LPC features) and spatial features obtained from multiple microphones. After obtaining speaker models from non-overlapped data, the system also attempts to assign correct labels to the overlapped segments by selecting the two most likely models for each such segment.

Finally, Geiger et al. (2013a) suggest that linguistic content may also help to detect overlapping speech. In particular, they found that certain words (such as "uh-huh", "yeah", "but" or "wait") were more likely to occur during short overlapping segments, suggesting that automatic speech recognition may be able to

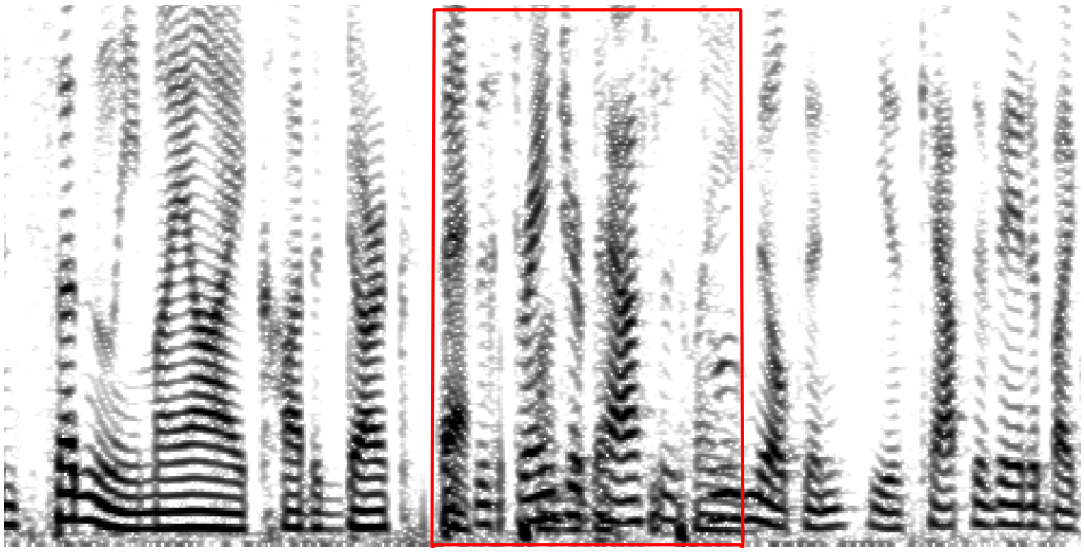

Figure 8.1: Part of a spectrogram with relatively distinguishable overlapping speech: there are two speakers who overlap in the middle (highlighted in a red frame).

offer useful information in a diarization system. However, this assumes that the automatic transcription is accurate, which may be difficult to achieve for overlapping speech.

8.2.2 Overlap Detection Using Deep Neural Networks

As an alternative to manually selecting hand-crafted features, one may also take advantage of deep learning – a well-trained neural network can be capable of extracting the relevant information from appropriate "raw" input or from less specialized features.

One relatively popular choice of input data is the spectrogram. One may observe that when looking at a spectrogram of a multi-speaker conversation, it is sometimes possible to distinguish between a single speaker and overlapping speech by sight. An example of this can be seen in Figure 8.1: On the left side of the image, the first speaker's speech shows very clear and regular parallel lines, corresponding to the harmonics. Meanwhile, the overlapped region in the middle looks significantly more chaotic. A neural network should also be able to detect such patterns.

Examples of the use spectrograms for overlap detection include Sajjan et al. (2018) and Kazimirova and Belyaev (2018). Such an approach was also used for the overlap-related experiments in section 9.5 of this thesis (also published as Kunešová et al., 2019). Miasato Filho et al. (2018) also presented a DNN which uses the log spectrum to provide information about overlaps together with speaker embeddings and VAD.

Additionally, Shokouhi et al. (2015) also detect overlaps from spectrograms (or, more precisely, from *pyknograms*, which are enhanced spectrograms), but they do not use deep learning. Instead, their system simply detects sudden jumps in the harmonic structure by calculating the euclidean distance between consecutive

frames.

Andrei et al. (2017) use the signal's frequency spectrum, but also combine it with hand-crafted features – specifically MFCC, signal envelope and auto-regressive model coefficients.

Other examples of DNN-based overlap detection include the works of Diez et al. (2018b), who use a feed-forward network with an input of MFCC and pitch features, Geiger et al. (2013b), who were among the first to employ a LSTM network for overlap detection, with a variety of hand-crafted features as its input, and Bullock et al. (2020), who use a BiLSTM with SincNet features.

8.2.3 Evaluation of Overlap Detection

Table 8.1 shows an overview of some of the recent works featuring overlap detection, including many of the examples which were listed above.

As evidenced in the table, there is no single established standard for evaluating overlap detection. While most authors report some combination of frame-level precision, recall, F-score and accuracy, some evaluate using Equal Error Rate (EER) based on per-overlap miss and false alarm, or in terms of Overlap Detection Error (ODE), which is calculated as the total duration of missed and false alarm overlaps, relative to the total duration of overlapping speech in the recording. Alternatively, some authors simply state the relative decrease in a diarization system's DER.

Table 8.1: Overview of recent systems featuring the detection of overlapping speech and their reported results.

Sys.	Dataset	Prec.	Rec.	F-score	Acc.	Other	
[1]	AMI (IS)	0.76	0.25	0.38	–	–	–
	AMI (all)	0.67	0.26	0.37	–	–	–
[2]	AMI	0.85	0.30	0.44	–	ODE	75%
[3]	AMI	0.82	0.28	–	–	ODE	78.3%
[4]	AMI	0.79	0.32	–	–	ODE	76.9%
[5]	AMI	–	–	0.51	–		20.3%
	RT-09	–	–	0.46	–	rel. DER decrease	6.7%
	ICSI	–	–	0.49	–		7.2%
[6]	AMI	0.66	0.45	0.53	–	ODE	78.67%
[7]	Prof-Life-Log	–	–	–	–	EER	5–45%
[8]	(Custom)	0.81	0.78	0.80	0.80	–	–
[9]	AMI	0.78	0.36	–	–	OL detect. err.$^{(*)}$	14.13%
						EER	8.52%
[10]	SSPNet	0.71	0.78	0.75	0.92	EER	11.31%

(continued on the next page)

Table 8.1: (continued)

Sys.	Dataset	Prec.	Rec.	F-score	Acc.	Other	
[11]	DIHARD I (dev)	–	–	–	–	rel. DER decrease	1.2%
	VAST only	–	–	–	–		6.3%
[12]	AMI	–	–	–	–	single spk. acc.	87.9%
						overlap acc.	71.0%
[13]	AMI + ICSI	0.51	0.41	0.46	–	–	–
	DIHARD I (dev)	0.62	0.10	0.17	–	–	–
[14]	AMI	0.87	0.66	0.75	–	rel. DER decrease	24.8%
	DIHARD II	0.65	0.27	0.38	–	–	–
	ETAPE	0.70	0.62	0.65	–	–	–
[15]	AMI	–	–	–	0.95	Average Precision	56.6%
	CHiME	–	–	–	–		51.1%

[1] Boakye et al. (2008b), handcrafted features → HMM, see also in Table 7.3
[2] Zelenák et al. (2012) – handcrafted features + spatial f. from multiple microphones → HMM
[3] Geiger et al. (2013a) – HMM using handcrafted features + linguistic information from transcript
[4] Geiger et al. (2013b) – handcrafted features → LSTM predictions, both → HMM
[5] Yella and Bourlard (2014) – HMM: handcr. f. + OL prob. based on silence and spk. changes
[6] Dighe et al. (2014) - OL detection + speaker identification for diarization, using Vector Taylor Series; listed result is with GMMs trained on oracle segmentation
[7] Shokouhi et al. (2015), enhanced spectrograms → detecting sudden jumps in harmonic structure (Euclid. dist. betw. frames); EER depends on the amount of added noise
[8] Andrei et al. (2017), handcrafted features → DNN
[9] Hagerer et al. (2017) – MFCC → bidirectional LSTM
(*) Unlike other listed systems, overlap detection error in [9] appears to be calculated relative to the length of the *entire* recording
[10] Kazimirova and Belyaev (2018), spectrogram → CNN, evaluated only on voiced frames
[11] Diez et al. (2018b) – MFCC + pitch features → feed-forward NN (2 hid. lay.)
[12] Sajjan et al. (2018) – spectrogram → LSTM classifier, Viterbi decode
[13] Miasato Filho et al. (2018) – spectrogram → DNN for joint embedding extraction, overlap detection and VAD. Listed results correspond to system "B1".
[14] Bullock et al. (2020) – SincNet features (Ravanelli and Bengio, 2018) → BiLSTM
[15] Cornell et al. (2020) – MFCC → temporal convolutional network

8.3 Data for Overlap Detection

Training and evaluating an overlap detector, especially a DNN-based one, generally requires a large amount of well-annotated data with frequent overlaps. Unfortunately, there do not appear to be any publicly available datasets made specifically for this purpose, and other corpora often lack sufficiently precise labels.[1]

One option is to obtain improved time labels by performing force-alignment (e.g. Boakye et al. (2008b) and Sajjan et al. (2018) on the AMI corpus). This is viable for data where we have both accurate transcriptions and separate single-speaker channels available, but there still may be issues with the amount of data.

[1] as seen for example in section 9.5.2 with the SSPNet Conflict Corpus

The second, commonly used option, is to create synthetic data by artificially combining multiple single-speaker recordings (e.g. Hagerer et al., 2017; Edwards et al., 2018; von Neumann et al., 2019).

This leads to somewhat unrealistic data compared to a natural conversation – in a real scenario, speakers usually do not talk independently, but rather react to each other. For example, a speaker who was interrupted may abruptly cut off mid-sentence, or one of the speakers may raise their voice, trying to drown out the other. It is also important to take care in combining the data so that there are no discernible seams or sudden changes in background noise which a DNN may inadvertently learn to detect instead of the overlap itself.

On the other hand, by automatically generating such a dataset from clean single-speaker recordings, it is possible to obtain large amounts of data with accurate timing and plenty of overlaps. This option was also used for training an overlap detector as part of the experiments in section 9.5.2.

When preparing data for overlap detection, one more thing to consider is the classification of overlaps with non-speech sounds such as laughter or humming. From an acoustic point of view, these sounds can clearly be identified as a specific speaker and such overlaps should be excluded from the clustering process. On the other hand, this may not be desirable in the final labeling.

This is also a concern when evaluating an overlap detector: speech transcriptions often do not include non-speech sounds such as laughter or humming, especially when they happen in the background of another speakers' speech, so such regions may be (in this case incorrectly) marked as non-overlap. This may in turn lead to a seemingly high false alarm rate of an overlap detector evaluated on such data.

8.4 Other Overlap-related Speech Processing

Section section 8.2 provided an overview of the existing literature focused on the detection of overlapping speech. Besides this topic, there are also other overlap-related speech processing tasks which can be relevant for speaker diarization.

8.4.1 Identification of Simultaneous Speakers

Some authors attempt to identify the speakers involved in overlapping speech, beyond simply assigning the two labels which are individually most likely. However, most such research is in the context of speaker identification and as such, assumes that the models of these speakers are known in advance.

Tsai and Lee (2010) in particular identify simultaneous speakers by a priori training models for every possible combination of two different speakers found in their database, then following a standard speaker identification approach with these combined models as separate "speakers".

Another example is the work of Sundar et al. (2013), who model the speech

signal as a combination of known speaker models, estimating the weights of these individual models. Based on the estimated weights, they decide which speakers are present in a given speech interval.

Walsh et al. (2007) proposed a system for joint source separation and identification of known speakers in a multi-microphone scenario. This is done using an expectation propagation approach.

More recently, Li and Whitehill (2020) proposed compositional embeddings, which "extend single-speaker embeddings through a composition function that is trained to estimate the location in the embedding space of where the union of two (or more) speakers is located. By composing the embedded one-shot examples (i.e., a sample of each person speaking in isolation) and comparing the result to the embedding of the test audio, the set of speakers can be inferred."

Finally, Dighe et al. (2014) combine overlap detection and speaker identification using a Vector Taylor Series approach, and directly apply it to speaker diarization. Though their system can use speaker models obtained from speaker diarization, it appears to work significantly better with GMMs based on oracle segmentation.

8.4.2 Signal Source Separation

Finally, there is also the related task of signal source separation: when faced with overlapping speech, one may also attempt to separate it into multiple signals corresponding to individual speakers (e.g. Xu et al., 2018).

This is more relevant in the field of ASR, where such separation can help in recognizing the speech of simultaneous speakers (e.g. Kanda et al., 2020). However, a small number of authors have also used blind source separation techniques for speaker diarization.

One example is the work of von Neumann et al. (2019), which was previously mentioned in section 4.3.1: A recurrent neural network processes the audio stream in blocks of 2.5 seconds, and iteratively separates each block into multiple signals corresponding to individual speakers. The system tracks speaker identities between blocks, thus acting similarly to sequential clustering and providing speaker diarization which can handle overlapping speech.

Similarly, (Kounades-Bastian et al., 2017), proposed a method which combines multi-channel voice separation and speaker diarization.

8.5 Overlapping Speech in Speaker Diarization

In the context of speaker diarization, the problem of properly handling overlapping speech can be divided into two tasks:

1. Detecting the presence of overlapping speech
2. Identifying the exact speakers involved

Not all systems fully implement both of these tasks – while obtaining correct labels is desirable, identifying the exact speakers is arguably more difficult than mere overlap detection.

However, even if we do not know which speakers are active in an overlapping region, we can still at least prevent the overlaps from negatively influencing the rest of the diarization process by excluding the data from any clustering or model adaptation. The effects of this were explored by e.g. Otterson and Ostendorf (2007) and Boakye et al. (2008b).

In offline systems, detected overlapping speech is typically left unlabeled until the rest of the conversation is processed. Afterwards, each such interval can be assigned one or two labels. These may be chosen on the basis of acoustic similarity to individual clusters, or simply as the speakers of the nearest non-overlapping speech on each side of the interval.

This simple approach can be also further improved by e.g. using different decision thresholds for overlap exclusion and for assigning multiple labels, as in (Diez et al., 2018b) and (Yella and Bourlard, 2014). This way, all potential overlaps are excluded from clustering, but only those with a higher confidence are given two labels.

In an online system, the process can be similar, although it is made somewhat more difficult by the need to assign labels immediately, without having future information.

Finally, some authors have recently proposed modified diarization approaches which take overlaps into account as part of the main diarization steps. This includes compositional speaker embeddings by Li and Whitehill (2020), overlap-aware spectral clustering (Raj et al., 2020a), the region proposal network approach by Huang et al. (2020) (combined overlapped speech segments proposal and embeddings extraction), overlap-aware VB-HMM resegmentation (Bullock et al., 2020), and the DOVER-Lap method for combining the outputs of multiple overlap-aware speaker diarization systems (Raj et al., 2020b).

Chapter 9

Experiments

The experiments in this chapter were focused on two different implementations of speaker diarization. Earlier experiments involved the implementation of an online GMM diarization system and the exploration of its possible improvements. Later experiments were centered around an i-vector based system. This second system primarily operated offline, but an online variant was also explored. The offline system was also used for participation in the DIHARD Speaker Diarization Challenge, which is likewise described here.

A final set of experiments, described in section 9.5, focused on the detection of overlapping speech.

9.1 Used Datasets for Speaker Diarization

The systems and algorithms described in this thesis were developed and tested using several different datasets with distinct characteristics. This section gives a brief overview of the most important ones. Relevant statistics are also summarized in Table 9.1.

This list does not include the synthetic data which were created for overlap detection – for the details on these, see section 9.5.2.

Table 9.1: Overview of the datasets used for evaluating speaker diarization systems. Values for DIHARD vary between different subsets. Overlap is relative to the total amount of speech and the values for AMI are based on word-level transcripts, excluding non-speech sounds.

Dataset	files	hours	speakers per file	unique speakers	overlap % per file	overlap total %
Czech Parliament	8	28	10–56	131	0.1-0.5	0.2
AMI Meeting Corpus	171	100	3–5	189	2.7–30.5	13.5
CALLHOME (AmEng, 2 spk)	109	17	2	208	1.9–32.5	8.8
DIHARD I (development)	164	19	1–10	∼500	0.0-94.0	7.9
DIHARD I (evaluation)	172	20	1–9	∼500	0.0-86.4	8.9

9.1.1 Czech Parliament Sessions

Initial experiments with online diarization were performed on a set of television broadcasts of Czech parliament sessions. This consisted of eight video recordings, each several hours in length, with a total length of 28 hours. All sound sources were on a single channel, sampled at 16 kHz.

In contrast to the other corpora, these recordings contained very long speaker turns (the majority being longer than 10 seconds, with many several minutes long) of single individuals with very little overlapping speech.

Additionally, as these parliament sessions took place in a large hall, with the speakers' voices being amplified by loudspeakers, there is very high reverberation.

Unlike the other datasets listed in this section, these recordings were not part of a publicly available corpus.

9.1.2 The CALLHOME American English Corpus

First experiments with i-vector based diarization were focused on the American English subset of the CALLHOME corpus (Canavan et al., 1997). CALLHOME is a multilingual corpus of telephone conversations between up to seven participants in six different languages. However, only English-language recordings with two speakers were included in the experiments.

This consisted of 109 separate conversations with a total length of 16.5 hours and a typical length of 5–10 minutes per conversation. The speakers in each recording were mixed into a single telephone channel, sampled at 8 kHz.

As the corpus consists of spontaneous conversations, the data contain many very short speaker turns (see Figure 5.1 for an example of a short speaker turn) as well as frequent instances of overlapping speech.

The CALLHOME Corpus is a very popular choice for evaluating diarization systems. For an overview of recent state-of-the-art results on this dataset, see Table 7.1.

9.1.3 AMI Meeting Corpus

The AMI Meeting Corpus[1] (Carletta et al., 2006) is a corpus of recorded meetings, usually with 4 participants, although a small number of recordings have 3 or 5 speakers. This includes both real, natural conversations and staged "scenario" meetings.

The corpus consists of 171 conversations from six sets of meetings at different locations, with a total time of 100 hours and individual lengths of 8–90 minutes. There are 189 different speakers and usually 2-4 recordings with each group of 3–5 participants.

[1] Available from: https://groups.inf.ed.ac.uk/ami/corpus/

The conversations are recorded using multiple microphones – including a microphone array as well as individual headsets worn by each speaker. However, for the experiments described within this thesis, only the single-channel "headset mix" recordings were used.

Most of the recordings contain relatively large amounts of overlapping speech (up to 30 % of all speech in the conversation) as well as some noise. The average amount of overlapping speech is 13.5%. If overlaps with non-speech sounds (such as laughter, breath and whistling) are also considered, this increases to 15.8%, with a maximum of 40%.

For an overview of past state-of-the-art results on the AMI corpus, see Table 7.3. However, there are some difficulties in comparing the results with each other. Aside from the fact that various authors use different subsets of the data, such as only meetings from a specific location, there are also several different sets of reference labels in existence – the original data contain both word-level ("words") and utterance-level ("segments") transcripts and the speaker boundaries in these are not identical. Additionally, some authors use force-aligned references.

For the experiments in this thesis, the reference labels were generated from word-level transcripts, ignoring non-speech sounds.

9.1.4 DIHARD Challenge Data

Certain experiments which are described in this chapter were performed during the author's participation in the First and Second DIHARD Speaker Diarization Challenge (DIHARD I, DIHARD II).

These two challenges used a set of data from multiple corpora with a variety of different domains – such as meetings, radio interviews and YouTube videos. A more detailed description of this dataset can be found in section 9.4.

9.2 GMM-based Online Diarization

One of the initial objectives of the research was specifically online diarization of television broadcasts from Czech parliament sessions, so early work focused on this scenario.

This lead to the following expectations and requirements:

- Long duration of individual recordings (multiple hours)
- Long individual utterances, infrequent speaker changes with very little overlap.
- A large number of speakers in each recording, the total number is not known in advance.
- New speakers may appear at any point in time.
- Although the list of possible speakers (i.e. current members of parliament) is technically known in advance, the system should not require such prior knowledge for its operation.

The proposed system started as a re-implementation of the work of Markov and Nakamura (2007), which is one of the GMM-based sequential clustering systems referenced in section 4.3.1. Its main principle is iterative creation of new speaker models by adapting one of a pair of gender dependent UBMs (male and female). This specific approach was chosen because the aforementioned system had been designed for a very similar type of data (European parliament plenary speeches) and fits all the requirements listed above.

Subsequent work then focused on improving the implemented system. The results obtained on the target data were also published as part of two conference papers (Campr et al., 2014; Kunešová and Radová, 2015), portions of which are reproduced in the text of this chapter.

It is important to note that the work on this system started in 2013, in the first months of the author's doctoral study. Although GMM-based speaker diarization can be considered obsolete today, at the point when these experiments were originally started, it was still the state-of-the-art for online diarization (as seen in the overview of online diarization systems in Table 7.8).

9.2.1 Online Diarization System

Following is a description of the implemented system, adapted from (Kunešová and Radová, 2015).

The system uses a sequential clustering approach, starting with only two GMMs, one for each gender, which are trained in advance. The audio stream is divided into short segments and for each of them, the system decides if the segment corresponds to an already known speaker or a new one by comparing the likelihoods of the gender dependent and speaker models. In the case of a new

speaker, a new model is created by copying one of the gender dependent models. Otherwise, one of the existing models is selected. The assigned model is then adapted using the data from the segment. The entire process is also illustrated in Figure 9.1.

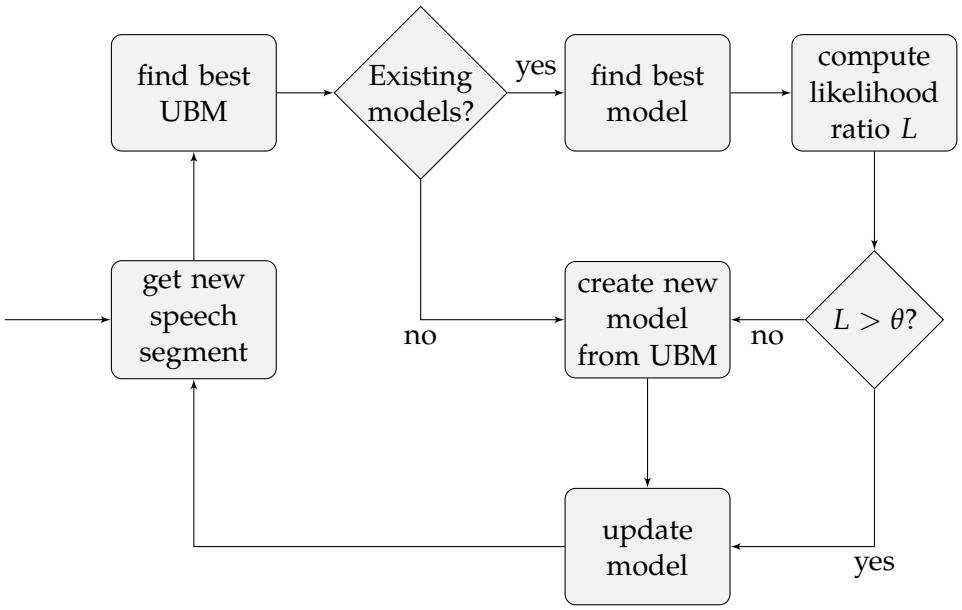

Figure 9.1: The decision process of the implemented diarization system.

The system consists of several modules:

1. Feature extraction and voice activity detection
2. Speech segmentation
3. Speaker identification and novelty detection
4. Online GMM learning

Feature Extraction and Voice Activity Detection

The system used 20 LFCCs as features. These were extracted using 25 filters in range from 50 Hz to 8 kHz, with a 25 ms FFT window and 10 ms shift. The 20 cepstral coefficients were computed without the energy coefficient and no cepstral normalization was performed.

The feature extractor also performs energy-based VAD, with every frame being labeled as *speech* or *silence* based on a threshold.

Speech Segmentation

Using the information obtained from the VAD and parameters such as the minimum and maximum segment length and the maximum pause length in a segment, the speech is divided into short segments. Of each segment, only the frames which

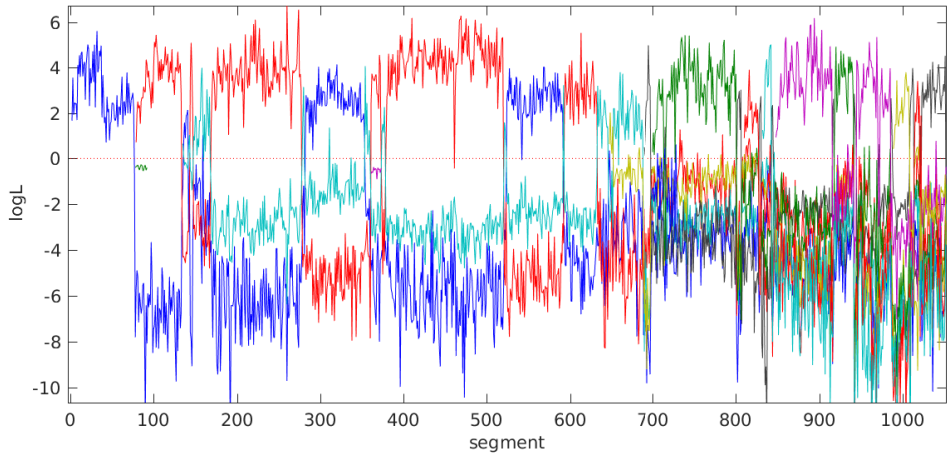

Figure 9.2: Logarithm of the likelihood ratio $L(X, \lambda_{sp})$ from (9.1) for all speaker models in a part of one recording. Each color represents a different model. Highest value of $L(X, \lambda_{sp})$ corresponds to the winning model for the given speech segment.

were labeled as *speech* by the energy-based VAD are used in the subsequent steps, as experiments have shown that this leads to both reduced computation time and improved performance of the system.

For the Czech Parliament data, segments were between 1 and 5 seconds long. For later tests on the AMI corpus (section 9.2.4), this was lowered to 1–2 seconds, as there were more frequent speaker changes.

Speaker Identification and Novelty Detection

For each speech segment the system uses a maximum-likelihood classification to determine both the speaker's probable gender (using the gender dependent models) and their most likely identity out of the existing speaker-model candidates. Afterwards, a likelihood ratio test is used to decide whether the segment belongs to the chosen identity, or represents an entirely new speaker.

The likelihood ratio is as follows:

$$L(X, \lambda_{sp}) = \frac{P_{\lambda_{sp}}}{P_{\lambda_{gen}}}, \tag{9.1}$$

where X is a speech segment and $P_{\lambda_{sp}}$ and $P_{\lambda_{gen}}$ are the likelihoods of the winning speaker and the appropriate gender dependent model, respectively.

If $\log L(X, \lambda_{sp}) \geq \theta$, the segment X belongs to the speaker represented by the model λ_{sp}. Otherwise it belongs to an entirely new speaker. In this latter case, a new model is created by duplicating the corresponding UBM.

The optimal value of decision threshold θ was found experimentally.

The speaker identification process is illustrated in Figure 9.2, which shows an example graph of the likelihood ratios of all models in a part of one recording.

Online GMM Learning

Once the new speech segment is assigned to an appropriate speaker model (either a previously existing one or one which was newly created from a UBM), the model is adapted using the data from the segment. This is achieved using an online variant of the EM algorithm, which was proposed by Sato and Ishii (2000).

This single-pass algorithm works by updating the parameters of a GMM after every new observation, allowing the diarization system to adapt individual speaker models gradually, on a frame-by-frame basis. The details of the algorithm can be found in the above-mentioned paper.

For the experiments, the adjustable parameters of the algorithm (the learning rate) were chosen to be the same as used by Markov and Nakamura (2007).

9.2.2 Improvements

As is usual in systems of this type, one of the most problematic areas of the system is the selection of the data-dependent decision threshold θ, which is used to decide whether a speech segment belongs to a new speaker or an already known one. If this threshold is set too low, multiple speakers may be assigned the same model. Conversely, if it is too high, speech belonging to a single real speaker may be divided between several different models.

To combat this issue, it was decided to select a higher decision threshold, so that an excess number of speaker models is created, but then implement an additional algorithm that identifies any models that are likely to correspond to the same speaker and cluster them all into one.

For this purpose, several approaches to clustering were explored. In the first round of experiments, the task was simplified by performing an offline clustering after the whole audio recording had been processed (Campr et al., 2014). Later experiments improved the offline approach and then extended it in order to perform the clustering process online, as part of the main diarization system (Kunešová and Radová, 2015).

In (Campr et al., 2014), the resulting offline diarization system was also combined with face models obtained from the video domain into a multimodal system which achieved slightly improved results over the audio modality alone. However, these aspects of the final system were handled solely by the other authors of the paper and as such are not included in this thesis.

Offline Clustering

The offline clustering approach used the Cross-Likelihood Ratio (CLR, described in section 2.2.1) to compute distances between all pairs of models and identify groups of models which likely correspond to the same real speakers.

The clustering itself is then performed as follows:

1. Let $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_N\}$ be the set of speaker models obtained from the online diarization.
2. Find model $\lambda_i \in \Lambda$ which had the lowest number of speech frames assigned to it during the online part of the diarization.
3. Set $\Lambda = \Lambda - \{\lambda_i\}$.
4. Find model $\lambda_j \in \Lambda$ such that $CLR(\lambda_i, \lambda_j)$ is minimal
5. If $CLR(\lambda_i, \lambda_j) < \phi$, where ϕ is a data-dependent threshold, consider λ_i and λ_j to represent the same speaker and reassign speech from λ_i to λ_j. However, do not update λ_j or its distances to other models.
6. Repeat steps 2–5 until Λ only contains one model.

As with the novelty detection threshold θ, ϕ is found experimentally.

Online Clustering

To identify similarities between models online, a method based on the offline variant was used. It has the benefit of requiring very little additional computation time, as most of the necessary calculations, namely the likelihoods of speaker models for each speech segment, are already being performed as part of the base system.

To find the distance between models λ_i and λ_j at time t, the following modification of the CLR distance was employed:

$$d(\lambda_i, \lambda_j, t) = \min\left(D(i, j, t), D(j, i, t)\right) \ , \tag{9.2}$$

$$D(i, j, t) = \frac{1}{N_i(j, t)} \cdot \sum_{\mathbf{x} \in S_i(j,t)} \log\left(\frac{\max(P(\mathbf{x}|\lambda_m), P(\mathbf{x}|\lambda_f))}{P(\mathbf{x}|\lambda_j)}\right) \ . \tag{9.3}$$

Here, $S_i(j, t)$ represents a set of all the speech segments which were assigned to speaker model λ_i between the creation of λ_j and the current time t (and for which we thus have calculated the likelihood of λ_j), $N_i(j, t)$ is the total number of frames of the speech segments contained in $S_i(j, t)$.

Once the system decides that several of the models represent the same speaker, there are two possible approaches to clustering, apart from simply discarding all of the models except one. We can either use a suitable method to transform all of the similar models into a single GMM, or we can retain all of them while treating them as a single speaker.

- Merging multiple GMMs into a single one

  The simplest choice which was considered for the merging of several models into a single one is to obtain a *weighted sum of the original Gaussian mixtures*. This is computationally very simple and thus causes no immediate delay

for the system. Yet the increased number of Gaussian components causes redundancy in the model and will also slow down future calculations.

As an alternative which preserves the original number of Gaussian components, it was chosen to replace all of the models to be merged with a *single new GMM trained using all the data* originally assigned to all. This causes a significant delay in the whole process, so it is not suitable for use in practical applications where online diarization is required. However, this approach can be used to represent the best way to merge the models together, which is why it was used for comparison.

- Treating multiple GMMs as belonging to a single speaker

  In this approach to dealing with similarities of speaker models, the system retains all of the models. However, the models are treated as belonging to the same speaker. It means that all of them are being updated with the same data every time one of them is assigned a new segment.

  Because of this, after a certain number of updates these models should become nearly identical. At that point, it may be possible to discard all of them except one. Further experiments showed that the model which was trained on *least* data should be retained. The likely reason for this is that the "bigger" models can be more "polluted" by noise and misclassified speech.

  Similarly to the weighted sum of GMMs described above, this approach also causes no immediate delay for the system, but the additional GMM updates will slow down the calculations for future speech segments.

9.2.3 Results

The original experiments were performed on a set of recordings from Czech parliament sessions (detailed in section 9.1.1).

The UBMs were also created from these data. The two models (male and female speakers) were trained by using up to 1 minute of speech from each named speaker in the corpus (28 women, 103 men). The use of in-domain data allowed the models to be very small - the final system used models with only 8 GMM components (with full covariance matrices). This size was chosen during the initial testing as a good trade-off between computation time and error rate.

Note: It would have been more appropriate to use different subsets of the data for training the UBMs and for testing. However, this was impractical due to the small number of available recordings. It is the author's belief that this should not affect the results in any significant way, given the large number of different speakers and small complexity of the models.

The achieved results, comparing the three approaches to online speaker clustering as well as the offline variant, can be seen in Table 9.2. The errors were calculated with the customary 0.25 s forgiveness collar around reference speaker boundaries.

In addition to the evaluation of the immediate decisions, which are obtained after a segment is processed (*online results*, Table 9.2a), the table also contains the

Table 9.2: Comparison of the diarization performance on test data (Czech parliament sessions) in terms of DER [%], adapted from (Kunešová and Radová, 2015). The slight differences in missed speech and false alarm rates among different variants are caused by the removal of very short pauses within the speech of a single speaker.

(**a**) Online results

	miss	FA	SE	DER
Without additional clustering	1.56	1.05	5.96	8.57
Weighted GMM summation	1.55	1.07	4.94	7.55
GMM retraining	1.51	1.06	4.08	6.66
Delete one GMM immediately ($N_{upd} = 0$)	1.55	1.06	3.52	6.13
Multiple GMMs for a speaker, $N_{upd} = 30$	1.52	1.07	**3.35**	**5.94**
Multiple GMMs for a speaker, $N_{upd} = \infty$	1.52	1.07	3.46	6.05

(**b**) Final results

	miss	FA	SE	DER
Offline clustering	1.51	1.05	**2.18**	**4.75**
Weighted GMM summation	1.50	1.06	3.90	6.46
GMM retraining	1.47	1.06	3.66	6.19
Delete one GMM immediately ($N_{upd} = 0$)	1.50	1.06	3.00	5.57
Multiple GMMs for a speaker, $N_{upd} = 30$	1.49	1.06	2.99	5.54
Multiple GMMs for a speaker, $N_{upd} = \infty$	1.48	1.06	3.00	5.54

final values which can be achieved by retroactively relabeling previous speech whenever two models are found to represent the same speaker (*final results*, Table 9.2b).

The *online* results show that rather than attempting to create a new model by merging two similar ones, it is better to treat them as belonging to the same speaker and discard one of them after some time. Best results were obtained when discarding the model which was trained using the greater amount of speech after 30 updates. In this case, there was a relative improvement in DER of 30.69% in comparison with the base system.

Even better results can be achieved with offline clustering, but this method cannot be used for online diarization.

Due to the relatively unique nature of the Czech parliament recordings which were used as test data, it is difficult to compare these results with those of other past works. The most similar system, both in the choice of test data and in design, is that of Markov and Nakamura (2007; 2008), which originally served as inspiration for this approach. As previously mentioned, these authors evaluated their work on a very similar dataset of *European parliament* plenary speeches, and reported a DER between 5.3 and 11.9 % (depending on the selected latency of their system – see Table 7.8 on page 65). This is comparable to the results achieved here, which were 8.57 % DER in the initial system and 5.94 % with additional online clustering.

9.2.4 Application to Conversational Data

After the initial success on the original task, further experiments explored the system's suitability for more complicated data such as natural conversations.

The AMI corpus was selected for this purpose. As described in section 9.1.3, the corpus contains real and staged conversations between small numbers of speakers. Unlike the previously used parliament data, AMI has very frequent speaker changes and relatively large amounts of overlapping speech and noise, all of which complicates the diarization process.

The majority of the system's parameters were kept the same as in previous work. However, some aspects were modified during the testing on the AMI corpus. The number of LFCC features was increased to 52, including delta coefficients. Instead of real voice activity detection, the system used oracle speech/silence labels obtained from reference transcripts. Longer speech intervals were cut into segments of only 1–2 seconds (as opposed to the previous 1–5 seconds), to account for more frequent speaker changes.

UBMs had 16 Gaussian components and were trained on the LibriSpeech corpus, using approximately 80 s of randomly selected speech from each of the 2338 speakers (1128 female, 1210 male) in the three LibriSpeech training sets.

Influence of Overlapping Speech

The AMI data contain a large amount of overlapping speech. This is problematic for reasons previously noted in section 5.2. Among other issues, if a speaker model is updated with a segment containing overlapping speech, it may later lead to confusion between speakers, decreasing the system's overall performance.

In order to determine how much this affects the results on AMI, the system was also evaluated with such intervals excluded from the diarization process. This used oracle overlap labels, based on the reference transcripts.

Tables 9.3 and 9.4 provide results for the following different options:

a) The standard system with no information about overlaps

b) Intervals with overlapping speech are excluded from the diarization process and labeled as silence (resulting in a higher rate of missed speech)

c) Overlaps are initially excluded, but they are later assigned to the most similar speaker model, using maximum-likelihood classification. This classification is currently done in an offline manner at the end of the diarization process, as the alternative would require significant modification of the existing system.

d) The offline classification in c) assigns *two* labels to each overlap, representing the two most similar models. This will result in the lowest miss rate.

By default, the system's output is evaluated using the entire conversation. For

options a) and b), the error rates are additionally also calculated with overlap intervals not scored.

The latter pair of values provide the most straightforward measure of how much the presence of overlapping speech affects the results on the rest of the data. In both Table 9.3 and in Table 9.4 (for results with updated models), the difference between the two options is noticeable, suggesting that the system may benefit from some form of overlap detection.

Initial Results on AMI

Unfortunately, the initial tests have proven rather poor, resulting in a Diarization Error Rate in excess of 50 %, as seen in Table 9.3. Meanwhile, most of the state-of-the-art systems which were listed in the overview in Table 7.3 (page 58) achieved between 20–35 % DER on AMI.

An analysis of the system's output has shown that the system in its current state is not able to correctly distinguish between speakers, particularly when deciding if a new speech segment belongs to an existing speaker or a new one. Regardless of the selected threshold θ, the system creates an excessive number of additional models while at the same time combining other speakers together.

The most likely explanation is that the used models were not representing the individual speakers sufficiently well. This could be an issue of the models' size or structure, the updating process, or the initial UBM itself.

In the original experiments with Czech Parliament recordings, as well as in three of the four GMM-based sequential clustering systems which were mentioned in section 4.3.1, UBMs were trained on data from the same corpus. This is not the case here and it may be one of the major reasons behind the system's poor performance.

Overlapping speech also played a role in the result. However, it was clearly not the sole factor, as attempts to exclude overlaps prior to diarization lead to only a partial improvement.

AMI Diarization with Known Speakers

To better understand the problems with the AMI corpus, a different experiment was performed. Instead of sequential clustering with unknown speakers, the system was modified to work on a speaker identification basis, similar to the approach described in section 4.3.2.

The system receives the models of all speakers at the start and merely classifies each segment. Two alternatives were considered:

a) The models are unchanged during the entire process

b) The models are still being updated after each use, as they would be in the standard system

Table 9.3: Results of the GMM-based diarization system on AMI data with UBM trained on LibriSpeech, $\theta = 0$ and oracle VAD. These numbers correspond to the final labels, with merged models retroactively relabeled (using $N_{upd} = 30$). "eval OL" denotes if overlaps are scored during evaluation (✓) or excluded (✗)

option	eval OL?	DER	miss	FA	SE
base	✓	61.53	8.30	0.00	53.23
exclude overlaps, label as silence	✓	61.73	15.93	0.00	45.80
exclude + classify at the end (1 spk)	✓	57.70	9.06	0.00	48.64
exclude + classify at the end (2 spk)	✓	55.68	2.18	0.00	53.50
base, overlaps not scored	✗	58.34	0.94	0.00	57.39
exclude + do not score overlaps	✗	55.33	1.91	0.00	53.42
offline clustering - base	✓	41.48	8.30	0.00	33.17
+ overlaps not scored	✗	37.81	0.94	0.00	36.87
offline clustering - exclude	✓	42.81	15.93	0.00	26.88
+ overlaps not scored	✗	33.16	1.91	0.00	31.25

In either case, no additional models are created and none are merged or deleted. This means that there is no issue with novelty detection, and any errors will be simply due to confusion between speaker models.

The initial speaker models were created from LibriSpeech UBMs using a single pass of the same online GMM learning process which is used during diarization. Each speaker's model was updated using a small amount of speech selected either from the test file itself (option 1), or from a different file with the same speakers (option 2).

Most of the AMI participants recorded more than a single session; usually there are 2-4 meetings with each group of speakers. This means that one of the meetings from each set can be used to obtain speaker models for the other files in the set. In most cases the files share the same name, distinguished by a final letter (e.g. "ES2002a", "ES2002b", "ES2002c" and "ES2002d" all share the same 4 speakers). For the purpose of this experiment, files "a" were used for preparing the models and the corresponding "b", "c" and "d" files were used for testing.

Table 9.4 shows the results for all four combinations of settings (with or without model updates, models from the tested file or from a different one), with models updated using different amounts of data from each speaker. (However, this is merely the *maximum* duration; in some cases the amount of available data from a speaker is lower.)

The results in Table 9.4 suggest that when using only a single pass of the online GMM learning algorithm, the system requires approximately one minute of speech from each speaker to sufficiently adapt the LibriSpeech UBMs. With smaller amounts of training data, there is relatively high confusion between speakers. This helps explain the poor results in Table 9.3 – many errors likely originate from a single mistake at the beginning of the diarization process. This then increases through snowballing effect – as models are updated with incorrect data,

even more subsequent segments are classified incorrectly.

This effect is also clearly demonstrated here by the difference between the results with and without model updates, particularly in the 10 s and 30 s cases – with smaller amounts of training data, results with updates are significantly worse than when models are unchanged.

The benefits of considering overlapping speech are also more evident here than in Table 9.3. In the results with updated models, we can see that there is always a significant decrease in speaker error when overlaps are completely excluded by the system. In the case where models are fixed and never updated, the difference is much smaller.

Assigning labels to overlaps results in a slightly increased speaker error, as the classification is not completely accurate. However, as this also decreases the miss rate by a much larger amount, the overall DER still improves by a significant margin.

9.2.5 Conclusion

The basic approach which was implemented here appears to be best suited for situations matching the original scenario: long recordings with a large number of speakers, where additional speakers appear throughout the entire duration. Under these conditions, the system was able to achieve a very low DER of less than 6 %, a comparable result to those previously reported on a similar dataset.

However, in cases where there is only a small number of speakers who are present from the start, this approach proves problematic. Most importantly, the success is highly dependent on whether the speakers can be correctly tracked during their first appearance. If a pair of speakers is mistaken for the same person at the very start, they will likely remain combined for the entire duration and a large portion of the system's output will be incorrect.

It also appears that a suitable choice of UBMs may be critical. Based on the poor results on the AMI corpus with LibriSpeech UBM (section 9.2.4), it is suspected that the UBM needs to match the target data as closely as possible - preferably by being trained on recordings from the same source (such as previous episodes of the same broadcast or different meetings recorded in the same room). This suspicion could be verified by conducting further experiments with a UBMs trained directly on AMI, but this was not pursued at the time.

Finally, some fault also likely lies with the relatively simplistic speaker models which were used. Increasing their complexity may lead to improvements. However, at this point in time, GMMs are no longer considered the state-of-the-art solution, so it was decided to abandon their use entirely and move towards other options for speaker representation. For this reason, further experiments in this thesis are focused on the more recent i-vectors and x-vectors.

Table 9.4: Results of the GMM-based diarization system on AMI data, with speaker models precomputed using different (maximum) amounts of data. Evaluated on a subset of 106 files, with oracle VAD. False alarm rate was equal to zero in all cases. Non-zero miss rate is due to both overlapping speech and the exclusion of very short utterances during segmentation. "eval OL" denotes if overlaps are scored during evaluation (✓) or excluded (✗). Listed values are averages calculated from individual results for all files.

			speaker error			
option	eval OL?	miss	10s	30s	60s	100s
speaker models from the same file, models updated during diarization						
base	✓	8.21	35.09	16.06	5.08	3.90
exclude overlaps, label as silence	✓	15.80	29.50	11.48	2.53	1.87
exclude + classify at the end (1 spk)	✓	9.05	30.93	12.26	2.99	2.32
exclude + classify at the end (2 spk)	✓	2.29	34.17	14.90	5.34	4.69
base, overlaps not scored	✗	0.94	39.13	17.74	5.47	4.13
exclude + do not score overlaps	✗	2.00	34.09	13.33	2.97	2.21
speaker models from the same file, no model updates						
base	✓	8.21	27.28	11.60	4.60	3.27
exclude overlaps, label as silence	✓	15.80	24.47	9.19	3.02	1.88
exclude + classify at the end (1 spk)	✓	9.05	26.46	10.57	3.65	2.33
exclude + classify at the end (2 spk)	✓	2.29	29.54	13.43	6.21	4.78
base, overlaps not scored	✗	0.94	29.72	12.21	4.78	3.39
exclude + do not score overlaps	✗	2.00	28.58	10.83	3.57	2.23
speaker models from file "a", models updated during diarization						
base	✓	8.21	39.88	29.64	13.40	9.76
exclude overlaps, label as silence	✓	15.80	33.32	25.34	9.63	6.96
exclude + classify at the end (1 spk)	✓	9.05	34.76	26.53	10.21	7.50
exclude + classify at the end (2 spk)	✓	2.29	37.82	29.39	12.80	10.01
base, overlaps not scored	✗	0.94	44.45	32.86	14.73	10.58
exclude + do not score overlaps	✗	2.00	38.61	29.09	11.11	7.99
speaker models from file "a", no model updates						
base	✓	8.21	35.13	17.29	9.02	9.49
exclude overlaps, label as silence	✓	15.80	32.26	14.82	7.24	7.86
exclude + classify at the end (1 spk)	✓	9.05	34.40	16.30	8.06	8.53
exclude + classify at the end (2 spk)	✓	2.29	37.41	19.18	10.73	11.13
base, overlaps not scored	✗	0.94	38.58	18.69	9.66	10.27
exclude + do not score overlaps	✗	2.00	37.52	17.40	8.45	9.05

9.3 Speaker Diarization Using i-Vectors

This section details the work which was done with i-vector-based speaker diarization. This initially started as an *offline* system (section 9.3.1), but was later also adapted into an *online* variant (section 9.3.2). The segmentation experiments in section 9.3.3 examine both of these variants. Finally, section 9.3.5 presents a hybrid system which employs offline methods for online diarization.

Although all the experiments in this section utilize only i-vectors, the described system was later modified to use x-vectors as well. This was during the work on the DIHARD Speaker Diarization Challenge, which will be covered in section 9.4.

9.3.1 Baseline Offline Diarization System

Offline diarization experiments used an i-vector-based system which was developed together with Z. Zajíc (Zajíc et al., 2016; Kunešová et al., 2017; Zajíc et al., 2018; Zajíc et al., 2019). Its original purpose was the diarization of telephone speech, but it was later adapted for a wider variety of domains (see section 9.4).

The system follows the standard "bottom-up" framework of segmentation, clustering and resegmentation, as introduced in section 3.1. In its basic form, it operates as follows (some passages have been adapted from the above-mentioned publications):

First, each recording is divided into short segments, from which i-vectors are extracted. The i-vectors are then clustered to determine which parts of the signal were produced by the same speaker. Finally, the system performs GMM-based resegmentation to refine the positions of boundaries between speakers.

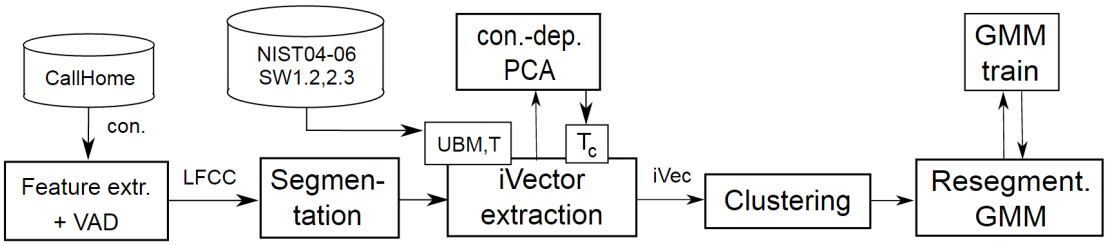

Figure 9.3: Diagram of the offline diarization system.

Feature Extraction Similarly to the GMM-based system (see section 9.2.1) this also used LFCC features with a Hamming window of 25 ms and 10 ms shift. In this case, there were 40 triangular filter banks linearly spread across the frequency spectrum, resulting in 25 LFCCs and delta coefficients, for a total of 50 features.

Voice Activity Detection The base system does not employ VAD. Following established practice in evaluating telephone speech diarization, the speech/non-speech information was taken from the ground-truth references.

Segmentation The baseline system used a simple time-based segmentation into intervals of equal length. Following the example of Sell and Garcia-Romero (2014), there are partial overlaps between neighboring segments. This increases the amount of data available for i-vector extraction while maintaining a higher number of segments. Other possibilities for segmentation have also been considered and are described in section 9.3.3.

Segment Description In the baseline system, each segment is represented by an i-vector. These are extracted via Factor Analysis (FA) (Kenny and Dumouchel, 2004; Machlica and Zajíc, 2012), using a UBM with 1024 components, trained on a variety of different corpora.

The size of the total variability matrix was 400 for telephone data (CALLHOME corpus) and 100 for later work with the DIHARD corpus. The dimensionality of the i-vectors is also further reduced using PCA, as in (Shum et al., 2011).

Clustering The initial version of the system was intended for the diarization of telephone speech, with the assumption of only two speakers in each conversation. Therefore, the system used a simple k-means clustering with cosine distance, into two target clusters.

Later, this was changed to AHC to allow an unknown number of speakers within an expected range:

The system starts with each i-vector in a separate cluster and then merges the closest pairs until it reaches a stopping point. The distance between two clusters is calculated as the average cosine distance between each pair of i-vectors. There is a fixed stopping threshold. However, if the resulting number of clusters would not fall within the expected range, the stopping point is adjusted so that the system reaches either the minimum or maximum allowed number of clusters.

If two partially overlapping segments are assigned to different clusters, the shared interval is initially left unlabeled and the final decision is left for the resegmentation step. Alternatively, if resegmentation will not be performed (such as in the online variant of this system), these intervals are simply split in the middle, with each half being assigned to the closer segment.

The k-means clustering variant was used for CALLHOME data (section 9.3.4), while AHC was used for the DIHARD Challenge (section 9.4).

Resegmentation The final step of the offline system is a frame-wise resegmentation of the entire data. This will refine the speaker boundaries and help to correct mistakes caused by imprecise segmentation.

First, the original feature vectors are used to train a GMM from each cluster. The number of GMM components depends on the amount of data in the clusters: 1 GMM component for every 2 segments, rounded down to the nearest power of 2 and with a maximum of 64.

As the second step, the entire conversation is redistributed frame by frame according to the likelihoods of the GMMs, filtered by a Gaussian window (with a length of 75 ms and 50 ms shift) to smooth the peaks in the likelihoods.

9.3.2 i-Vector-based Online Diarization

Experiments with i-vector based online diarization started with a modified version of the *offline* system which was described in section 9.3.1.

As the initial purpose of this version of the system was simply to investigate the sequential segmentation and clustering process, without the need for actual real-time output, it was decided against implementing a complete, fully online diarization system. Rather, the original offline process was simply adjusted so that each of the steps separately operates in a left-to-right manner, simulating an online system.

As such, the initial steps of both systems are identical. However, the original k-means clustering is replaced by a sequential algorithm, while both the conversation-dependent PCA reduction of i-vectors and the final resegmentation step, which are not possible to perform online, are removed entirely.

As the clustering step, the system employs the i-vector adaptation process proposed by Zhu and Pelecanos (2016), which is given by

$$T_n = \alpha V_n V_n^T + (1 - \alpha_n) I\,, \qquad \alpha_n = \frac{n}{n+R}\,, \tag{9.4}$$

where n is the number of i-vectors which have been processed so far, V_n is the first principal component of the i-vectors, T_n is an i-vector transformation matrix and R is the relevance factor which controls the rate of the adaptation.

The resulting sequential clustering then works as follows: For each new i-vector (which corresponds to a new segment), the system first updates the transformation matrix T_n using the formula in Equation 9.4 and uses it to transform all i-vectors seen up to this point. Then the cosine distance is calculated between the new transformed i-vector and all existing clusters, where the distance to a cluster is calculated as the average of the distances to all of its i-vectors. If the distance to the closest cluster is lower than a fixed threshold θ or the maximum number of clusters is reached (for CALLHOME, this number was 2), the new i-vector is assigned to this cluster. Otherwise, a new cluster is created.

Because all decisions made by the system are final and unchangeable, an incorrect decision at an early point in a recording can significantly impact the rest of the clustering process. In this regard, extremely short segments, particularly those under 0.5 seconds are the most problematic, as they typically do not contain sufficient information about the speaker in order to be correctly clustered.

Some of the segmentation approaches which were examined in section 9.3.3 may produce such short segments, so it was necessary to slightly adjust the clustering algorithm in order to avoid this issue. This is achieved by excluding any segments under 1 second in length from the regular clustering process. Instead, the corresponding i-vectors are simply labeled as the nearest existing cluster (they

are never used to create a new one), but they are not included in the calculation of T_n in Equation 9.4 nor considered in later distance calculations.

The results obtained by this system on CALLHOME data can be found in section 9.3.4.

9.3.3 Segmentation Experiments

The baseline system uses a simple segmentation into intervals of equal length. However, options involving speaker change detection were also considered and tested with both the offline and online versions of the system. These experiments are described in this section.

As previously stated in chapter 3, speaker change detection is often applied for the segmentation step of speaker diarization systems, as it allows to obtain segments which ideally contain only the speech of a single speaker. However, due to some of the common obstacles typically present in spontaneous telephone conversations, namely very short speaker turns and frequent overlapping speech, diarization systems aimed at telephone speech often omit the speaker change detection process. Instead, they use a simple constant length segmentation of regions of speech found by a speech activity detector (e.g. Sell and Garcia-Romero, 2014; Senoussaoui et al., 2014), with the expectation that the resegmentation step would resolve any inaccuracies.

This reasoning is often stated in relevant literature. However, to the author's best knowledge, no source had previously presented a detailed comparison of the two segmentation approaches on telephone data. For this reason, experiments comparing the two approaches were performed in order to investigate this matter.

Both offline and online diarization were considered – the individual clustering approaches are evaluated using the offline and online versions of the i-vector based system, which were described in sections section 9.3.1 and section 9.3.2, respectively.

Segmentation Approaches

Four different segmentation approaches were considered. All of these assume the possibility of their use in online diarization, i.e. they operate sequentially or could likely be relatively easily adjusted in such manner.

Some of the described approaches rely on information about the presence of silence and speech which would under real conditions be provided by a VAD. However, in order to avoid any specific VAD method from influencing the results of the segmentation, we chose to use *oracle* VAD obtained from the reference transcripts.

Fixed Length Segments The simplest segmentation option is to split all speech into short intervals of equal length, without considering any potential speaker boundaries. This is the baseline approach which was described in section 9.3.1.

As previously stated, the implementation follows the example of Sell and Garcia-Romero (2014) by using overlapping segments. This allows us to increase the amount of information contained in a single i-vector while retaining a higher precision of the segmentation. Specifically, we chose to use segment length of 2 seconds with a 1 second overlap between neighboring segments.

GLR-based Speaker Change Detection Two of the segmentation options employed speaker change detection. The first one of these followed a more traditional distance-based approach, using the Generalized Likelihood Ratio (GLR, described in section 2.2.1). In order to obtain segments of consistent length, comparable to the constant length approach, a two-step algorithm was implemented, incorporating a fixed minimum and maximum segment length. The two-pass nature of this algorithm means that it is not suitable for true online diarization in its current form. However, it should be possible to implement a relatively similar algorithm in a strictly left-to-right form.

In the first step of the segmentation process, the system identifies a smaller number of the most likely speaker change points by performing standard GLR-based speaker change detection using two neighboring sliding windows of 2 s with a step size of 0.1 s.

Likely speaker changes are identified as the locations of significant local maxima of the distances. For this purpose, the system calculates the prominence[2] of individual peaks in the distances and selects those with values exceeding a threshold.

The second step of the segmentation consists of further splitting any segments which are longer than the maximum allowed length. The point where a long segment is split is found in the following manner:

First, the system identifies an interval where a split can occur, such that neither of the resulting new segments would be shorter than the minimum allowed length. If there are any peaks within this smaller interval, the one with the highest prominence (as calculated during the first step of the segmentation) is selected as the new segment boundary. If no peaks are present, the segment is cut at the edge of the interval, at the point where the distance is highest. Figure 9.4 illustrates this process.

Finally, any segments which contain only a small percentage of speech frames (as determined by VAD), are labeled as silence and subsequently discarded.

[2]Peak prominence measures how much a given peak stands out within the signal. It was calculated using MATLAB's built-in `findpeaks` function.

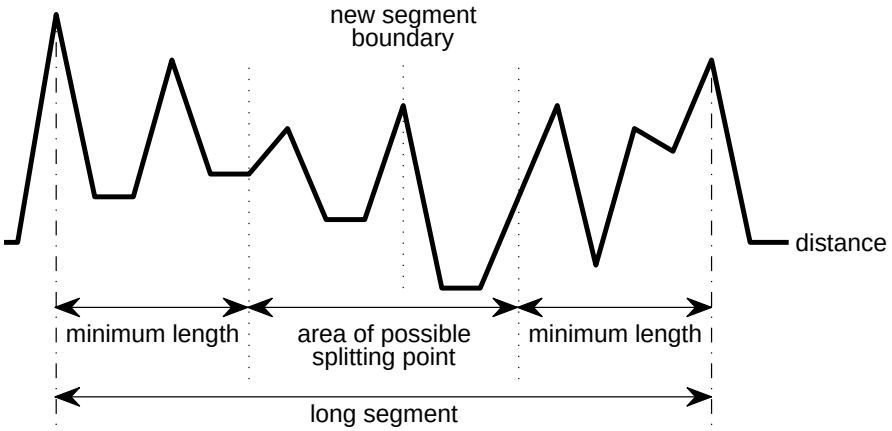

Figure 9.4: The process of splitting longer segments in the GLR segmentation approach.

CNN-based Speaker Change Detection The third approach which was considered was also based on speaker change detection, this time using a Convolutional Neural Network (CNN). This method was primarily designed by M. Hrúz (Hrúz and Kunešová, 2016; Hrúz and Zajíc, 2017).

The CNN was trained on spectrograms of acoustic signal using the method described in (Hrúz and Zajíc, 2017). It receives a short window (1.4 s) of a spectrogram and outputs the probability of speaker change in the middle of this window. The process is illustrated in Figure 9.5.

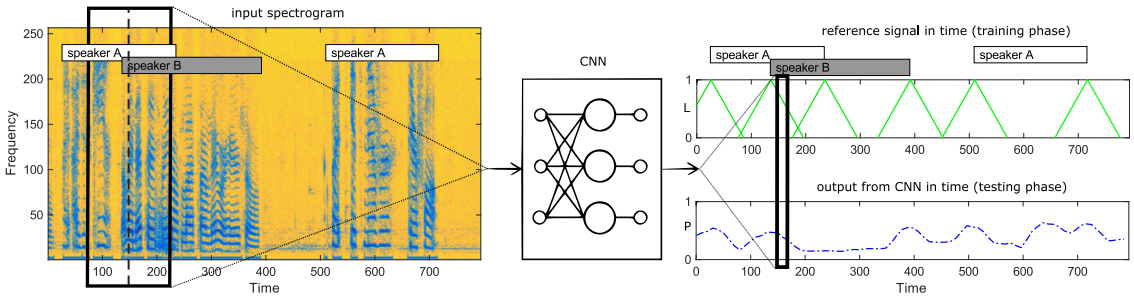

Figure 9.5: Illustration of CNN-based speaker change detection. The input speech as spectrogram is processed by the CNN into the output function $P(t)$ (probability of speaker change at specific time, shown in blue at the bottom right). The reference signal for the CNN training is depicted on top. Image reproduced from (Kunešová et al., 2017).

Speaker changes are identified as peaks in the output signal $P(t)$, with a threshold to remove insignificant local maxima. The signal between two detected speaker changes is considered to be one segment.

In the offline version of this segmentation approach, we discard any segments under 1 second in length, as they are considered unreliable. They are only processed later during resegmentation. However, in the online variant, which does

not have resegmentation, we keep all segments, regardless of length.

Processing the spectrogram window using a CNN takes only a very short time, which makes this approach suitable for online diarization.

This segmentation option also included an additional modification of i-vector extraction – during the statistics accumulation process, the data in each segment are weighted by $1 - P(t)$. The reasoning behind this is that we still cannot be certain that each segment only contains the speech of a single speaker. The parts of the audio segment with a high probability of a speaker change are likely less appropriate to represent the speaker than those with a small value of $P(t)$ (Zajíc et al., 2017).

Oracle Segmentation For comparison purposes, we also implemented oracle segmentation. In this approach, the conversations are split according to the reference transcripts: each individual record from the transcript becomes a single segment. As many of these segments are very short (often under 1 second), we adjust them slightly by joining any two segments from the same speaker which are separated by a silence of less than 0.5 seconds (this does not, however, eliminate all short segments). These relabeled silences are removed again at the end. Otherwise, the segments are kept exactly as recorded in the transcripts, including any partial overlaps.

9.3.4 Results on Telephone Data

For the evaluation of the offline and online system and the four different segmentation approaches which were described in section 9.3.3, we used the American English subset of the CALLHOME corpus of telephone speech (described in section 9.1.2). As a portion of the recordings had been used for training the CNN which we use for one of the segmentation approaches, we limited our experiments to the remaining 77 conversations with only two participants.

The results are evaluated in terms of DER, as described in section 7.1, with the customary tolerance collar of 0.25 seconds around speaker boundaries. Contrary to a common practice in telephone speech diarization, we *do not* ignore overlapping segments during the evaluation.

Table 9.5 presents the results achieved with the four segmentation methods by the offline and online versions of the system, which were described in section 9.3.1 and section 9.3.2, respectively.

The online system was evaluated with a fixed decision threshold $\theta = 0.6$ and different values of the relevance factor R, which controls the rate of the adaptation (see section 9.3.2). The adaptation process proposed by Zhu and Pelecanos (2016) can improve the final DER in all four cases, but the individual segmentation approaches have different optimal values of R.

The results of these experiments were initially published as part of two conference papers (Zajíc et al., 2016; Kunešová et al., 2017). In the original publications, the reported values for the fixed length segmentation were erroneously

high. Based on this, it was mistakenly suggested that this naïve approach works best with resegmentation and is unsuitable for online diarization. The mistake has since been corrected, but the updated results in Table 9.5 unfortunately no longer support the original conclusion – for the online system, fixed length segmentation surpasses both variants of speaker change detection. However, as the results with oracle segmentation show, there is still room for improvements.

Overall, the achieved results fall within the range of those of other past systems, which were shown in Table 7.1 on page 54. However, it is difficult to draw direct comparisons. As discussed in section 7.2.1, despite using the same corpus, different authors use different subsets of the data and evaluate them under different conditions (e.g. conversations with only two speakers or multiple; with system VAD, oracle VAD or even oracle segmentation; with overlapping speech excluded or not).

To obtain a more precise comparison, we instead sought a more standardized method of evaluation. This resulted in our participation in the DIHARD Speaker Diarization Challenge, which will be covered in section 9.4.

Table 9.5: Offline and online diarization results for different segmentation approaches, measured in terms of Diarization Error Rate (DER) [%]. R is the relevance factor of the i-vector adaptation, with the value of ∞ being equal to no adaptation. Decision threshold for the online approach was $\theta = 0.6$. (Results have been updated since the publication of the original paper.)

(**a**) Offline system (after resegmentation)

	DER [%]
fixed length segments	7.63
GLR speaker change detection	7.44
CNN speaker change detection	7.69
oracle segmentation	6.80

(**b**) Online system

R	∞	16384	...	1024	512	256	128	64	32
fixed l.	12.50	**12.36**	...	12.67	12.98	13.22	13.63	13.87	14.39
GLR	15.04	–		14.29	14.15	14.12	**13.74**	14.23	14.29
CNN	14.90	–		**14.52**	14.69	14.66	15.68	16.49	17.33
oracle	9.66	–		9.35	8.47	8.25	**8.04**	8.55	9.57

9.3.5 Hybrid Speaker Diarization

One of the online diarization approaches which were listed in chapter 4 was *hybrid online-offline diarization* (section 4.3.3). Such systems repeatedly perform fast *offline* diarization of past data in order to improve future online decisions.

This section details a brief experiment which explored such an approach.

The Hybrid System

The system used here is based on the earlier online i-vector experiments which were described in section 9.3.2 (including the i-vector adaptation process). However, the implementation was largely rewritten from scratch, in order to allow truly online processing with the option of parallelizing the offline part. Due to some preliminary simplifications and differences in configuration, the results may not be directly comparable with those in Table 9.5.

The hybrid aspect of the system is currently achieved via a simple reclustering process which is performed after every new segment. It is a centroid-based reclustering using cosine distance between i-vectors. In order to maintain continuity in labeling, the initial centroids are obtained from the existing clusters. As in section 9.3.2, segments shorter than 1 s are excluded from this process, and are simply given the label of the closest cluster at the time of their creation.

After the entire recording is processed, there is an optional final round of reclustering which retroactively relabels past segments. Since it also starts from the existing clusters, it is not fully equivalent to an offline clustering from scratch.

Currently, there are three segmentation options used by the system:

- Oracle segmentation - identical to section 9.3.3
- CNN segmentation - is based on the same CNN-based speaker change detection as in section 9.3.3. However, due to some differences in the implementation of the segmentation process, the resulting segments are slightly different.
- Uniform segmentation - the entire recording is split into intervals of 2 s. Unlike in earlier experiments, there are no overlaps between neighboring segments

At this moment, the system does not consider overlapping speech – all frames are given only a single label.

Results

The experimental hybrid system was evaluated on the same subset of the CALLHOME American English Corpus as in section 9.3.4: the 77 two-speaker conversations which were not used for training the speaker change detector.

Table 9.6 shows the results of four different configurations:

a) Baseline online system with no reclustering

b) Final reclustering only – there is only a single round of reclustering, performed at the end of the online diarization process. All segments are retroactively relabeled.

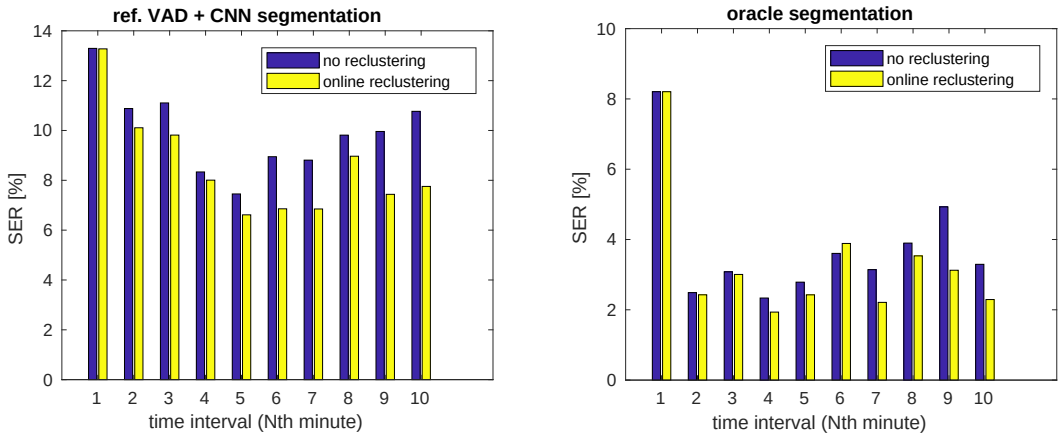

Figure 9.6: Development of the hybrid system's speaker confusion error (SER) on CALLHOME over time during diarization with and without periodic reclustering. SER is calculated from specific one-minute intervals of all recordings (n-th column corresponds to SER calculated from only the n-th minute of each recording). Plots show SER obtained when overlapping intervals are excluded from evaluation.

c) Online reclustering only – the system performs reclustering after each new segment. These results correspond to the immediate system outputs, *without* retroactive relabeling

d) Both online and final reclustering – these results correspond to the "Online reclustering" option, but with all segments retroactively relabeled at the end

Table 9.6: Results of the hybrid online-offline system with periodic offline reclustering (with 2 target clusters) on a subset of the CALLHOME corpus. Unlike in previous experiments in section 9.3, the uniform segmentation did not use overlapping segments. (However, oracle segmentation did – this is the reason behind its variable *miss*.)

System	miss [%]	FA [%]	Speaker Error (SER) [%]			
			no reclust.	final reclust. only	online reclust. only	both reclust.
Oracle seg. + ref. VAD	2.06–2.23	0.00	4.88	3.07	4.58	2.37
+ overlaps not evaluated	0.02	0.00	5.24	3.30	4.93	2.55
Uniform seg. + ref. VAD	3.52	5.07	12.94	7.74	10.89	7.36
+ overlaps not evaluated	0.05	5.45	13.91	8.32	11.71	7.91
CNN seg. + ref. VAD	3.51	2.48	10.14	7.53	9.54	7.61
+ overlaps not evaluated	0.03	2.67	10.90	8.09	10.25	8.18

The system is evaluated here primarily in terms of the speaker error (SER) component of DER, as it is the only one significantly affected by the reclustering.

Figure 9.6 also shows how the system's performance on the same recording changes over time. To obtain these results, all recordings were processed in full,

but only a specific one-minute interval was evaluated. The individual bars in the plot correspond to the speaker error (SER) calculated on these specific intervals (i.e. the third bar corresponds to the third minute of each recording).

The plots show that even in the baseline system the highest SER is at the start, when there are not enough segments yet to properly represent the clusters. After some time, the clustering quality improves. This effect is most dramatic in the oracle case, likely because the pure segments allow for the most accurate clustering.

Still, with online reclustering the SER decreases even further, particularly during later parts of the recordings, as the reclustering process begins improving the existing clusters.

Conclusion

The main purpose of this experiment was to replicate the hybrid diarization concept which is used by a small number of online systems, and to evaluate its potential. Although the chosen implementation is relatively simple, the results in Table 9.6 show a small but noticeable improvement in all tested cases.

The reclustering process is also sufficiently fast to be used in a real online scenario. The main bottleneck of the system is currently i-vector extraction, but this only needs to be performed once. If all the i-vectors are already precomputed at the start, then the *entire* hybrid diarization of a single ten-minute recording (including i-vector adaptation during sequential clustering) takes less than five seconds on a modern computer.

9.4 The DIHARD Speaker Diarization Challenge

The previous sections described the i-vector-based diarization system which was created in collaboration with Z. Zajíc. The system was initially evaluated on data from the CALLHOME corpus, but this did not give us a complete idea of how it compares to other contemporary works: as seen in chapter 7, comparisons using other authors' reported results can be complicated, even among systems tested on similar data. The exception is during organized evaluation campaigns, when it can be ensured that all systems are tested under the same conditions.

Thus, when we heard of one such evaluation being organized, we decided to participate with our own system, to see how it compares with the works of other research groups.

The evaluation in question was the DIHARD Speaker Diarization Challenge - a new series of diarization evaluations focusing on *offline* diarization of "difficult" data. This section describes our participation in the first two runs of the challenge (DIHARD I and DIHARD II) as part of team "ZCU-NTIS" / "UWB-NTIS".

The descriptions of the updated system and its results in both challenges were also published in two papers (Zajíc et al., 2018; Zajíc et al., 2019). Parts of the following text and tables are adapted from these sources.

9.4.1 Introduction

The First DIHARD Speaker Diarization Challenge (Ryant et al., 2018b) took place in February and March 2018, and was repeated for a second run from March to June 2019. The focus of the challenge was on *offline* diarization of such data where speaker diarization is difficult or where standard methods may fail.

This involved issues such as long single speaker monologues, very short utterances, large amounts of overlapping speech, varying levels of background noise, or speakers with unusual voices (e.g. very young children).

The challenge data come from a variety of different source corpora (listed in section 9.4.2). For this reason, there are great differences between individual recordings. This includes channel variations due to different recording equipment, the number of speakers, and the amount and type of background noise, which sometimes includes music.

Development data included information about the source corpus of each recording. However, for evaluation data, this information was not provided.

The systems in the challenge were evaluated under two different conditions – using reference VAD (Track 1) and with VAD provided by the system (Track 2).

9.4.2 The Data

In the first run of the challenge, the DIHARD datasets consisted of the following separate corpora (Ryant et al., 2018a):

- SEEDLingS = Child language acquisition recordings (Bergelson, 2016)
- SCOTUS = Supreme Court oral arguments
- DCIEM = "Map tasks"
- ADOS = Clinical interviews
- YouthPoint = Radio interviews
- RT-04S (development) and ROAR (evaluation) = Meetings
- LibriVox = Audiobooks (1 speaker per recording)
- SLX (development) = Sociolinguistic interviews in the field
- MIXER6 (evaluation) = Sociolinguistic interviews in a laboratory setting
- VAST = Web videos (audio only)
- CIR = Restaurant conversations (evaluation set only)

The development and evaluation datasets for DIHARD II included all of the original recordings from DIHARD I, as well as two additional corpora in the development set (CIR and SLX, both of which were previously only in the evaluation set) and one entirely new corpus in the evaluation set (DASS - sociolinguistic interviews conducted in the field). Recordings from the SEEDLingS and VAST corpora were also re-annotated to correct errors which were present in DIHARD I data.

9.4.3 The Modified Diarization System

We entered the challenge with our existing offline diarization system, which was described in section 9.3.1. However, the specifics of the challenge required some changes in the system, particularly to deal with the variety of different data and higher number of speakers. We also made some additional updates in the time between the two runs of the challenge - these will be noted where appropriate.

The main distinguishing characteristic of our new approach was the use of a classifier to automatically identify the source of each recording in the evaluation set. This in turn allowed us to fine-tune the parameters of the diarization system separately for each of the known corpora, as opposed to using a single, universal setup.

Aside from the domain classifier, the most important change from the earlier CALLHOME experiments in section 9.3.4 was the use of agglomerative clustering rather than k-means, to allow the system to be used with an unknown number of speakers. For DIHARD II, we also added x-vector extraction alongside the original i-vectors, combining both into a single "xi-vector".

For comparison purposes, we have also tested an official Kaldi implementation of speaker diarization, as well as a combined system which switches between the two alternative options based on a per-corpus basis. These are described in section 9.4.4.

Corpus classification

The corpus classifier is neural network with a single hidden layer, which classifies each recording as one of the corpora seen in the development data, or as "unknown corpus". The network's input is a single i-vector computed from the entire recording.

The classifier was designed and implemented by Z. Zajíc.

Classification accuracy was 95% on the DIHARD I development set and 90% on the evaluation set (excluding unseen corpora).

For DIHARD II this was extended to a two step classifier: the first level distinguishes between recordings with a single speaker and those with multiple speakers. The second step attempts to identify the source of multi-speaker data.

Feature Extraction

As a newly added step, Cepstral Mean Normalization (CMN) is applied to compensate for channel variations. Otherwise this is unchanged.

Voice Activity Detection

Results for Track 2 of DIHARD I were obtained with a DNN-based voice activity detector provided by J. Zelinka (Zelinka, 2018). During DIHARD II, we focused solely on Track 1 (diarization using reference VAD), so no voice activity detection was performed.

Segmentation

During DIHARD I, we tried two segmentation approaches: the default fixed length segmentation and an updated version of the CNN-based speaker change detection which was described in section 9.3.3.

In both cases, the entire conversation is first split into long intervals containing only speech; silence is excluded from subsequent processing.

As a second step, the speech is further segmented, either

a) into fixed length intervals of 2 seconds, with 1 second of overlap between neighboring segments, or

b) according to the speaker changes found by the CNN-based speaker change detector. As previously, each interval between two detected speaker changes (or between a speaker change and silence) is a single segment. In the DIHARD I version of the system, these are not split further.

To ensure that each segment contains sufficient information about the speaker, we set the minimum duration of each segment to 0.5 s. Shorter segments are dis-

carded from the clustering stage and the decision about the speaker is left for the resegmentation step.

In the final evaluation of DIHARD I, the segmentation based on speaker change detection achieved worse results than the fixed length segmentation (as shown in Table 9.9). Upon later examination, we found that the system had failed to detect some of the speaker changes and as there was initially no upper limit for segment length, this could sometimes result in very long segments containing multiple speakers.

During DIHARD II, we resolved this issue by using a segmentation approach which combines both of the above options: The initial speech intervals are first split according to the detected speaker changes, and then further cut into shorter segments of 2 seconds, with 1 second of overlap.

Speaker Representation

For the first DIHARD, the system used standard i-vectors, as described in section 9.3.1. For DIHARD II, we replaced our previous i-vector extractor with a Kaldi implementation of both i-vector and x-vector extraction[3].

We compared the Kaldi i-vectors and x-vectors, and as shown in Table 9.7, we achieved the best results by extracting *both* an i-vector and an x-vector from each segment and concatenating them into what we refer to as a "xi-vector".

For generating i-vectors, we trained a UBM with 2048 components and a transformation matrix with a latent dimension of 128. The x-vectors were extracted from the second-to-last layer of a Time Delay Neural Network and also had a dimension of 128.

We performed whitening of the resulting xi-vectors by subtracting the mean of the development set's xi-vectors. Finally, the dimension of the vectors is reduced using conversation-dependent PCA computed on the data in the current conversation. The final dimension depends on the identified corpus.

Table 9.7: Average DER [%] on the DIHARD II development set for an earlier version of our system and for the Kaldi system (see section 9.4.4) with different segment descriptors (x/i/xi-vectors).

System		DER
proposed	i-vectors	24.31
	x-vectors	23.81
	xi-vectors	**22.51**
Kaldi	i-vectors	25.83
	x-vectors	25.32
	xi-vectors	24.13

[3] https://github.com/kaldi-asr/kaldi/tree/master/egs/callhome_diarization/v1 and /v2

Clustering

The system uses one of two different clustering approaches, depending on the identified corpus of each recording.

A small number of corpora in the development set almost always have the same number of speakers in each conversation. For these, we can continue using k-means clustering with a fixed number of clusters. For DIHARD II, we replaced this option with k-medoids clustering, using PLDA rather than cosine distance.

For the rest of the data, the system uses an AHC algorithm, based on the average cosine distance between vectors. There is a fixed stopping threshold. However, if the resulting number of clusters would not fall within an expected range, the stopping point is adjusted so that the system reaches either the minimum or maximum allowed number of clusters. The AHC clustering approach always uses cosine distance. We have also tested PLDA as an alternative, but this did not lead to an improvement in our system.

The parameters of both clustering approaches were selected on a per-corpus basis using the development set. The target number of speakers is based on the actual numbers in each conversation, while the optimal threshold for the merging distance in AHC was found experimentally. The specific values used for DIHARD II are listed in Table 9.8.

Table 9.8: Experimentally chosen parameters for each DIHARD II corpus. In the "PCA dimension" column, "-" indicates that no PCA was performed.

Corpus	Clustering algorithm	No. of clusters	AHC threshold	PCA dimension
LibriVox	none	1	-	-
SEEDLingS	AHC	2–3	0.62	6
CIR	k-medoids	4	-	-
ADOS	k-medoids	2	-	-
SCOTUS	AHC	5–10	0.46	12
DCIEM	k-medoids	2	-	-
RT-04S	AHC	3–10	0.46	6
SLX	AHC	2–6	0.76	6
MIXER6	k-medoids	2	-	-
VAST	AHC	1–9	0.58	3
YouthPoint	AHC	3–5	0.54	9
unknown	AHC	2–6	0.10	-

9.4.4 Kaldi Diarization System

During DIHARD II, we also compared our diarization system with one based on an official Kaldi recipe for diarization[4].

[4] https://github.com/kaldi-asr/kaldi/tree/master/egs/callhome_diarization/v1 and /v2

In the Kaldi system, we used the same LFCC features as in our main system and we also combined i-vectors and x-vectors into xi-vectors. Otherwise it followed the standard Kaldi recipe.

- Segmentation is based only on VAD, without speaker change detection. Detected speech intervals are divided into segments of 1.5 s and overlap 0.75 s. The minimum length of a segment is 0.5 s.
- As in our main system, individual segments are represented by xi-vectors, a concatenation of i-vectors and x-vectors. The vectors are whitened by subtracting the mean and via transformation by an LDA matrix.
- The Kaldi system performs standard AHC clustering, with a simple stopping threshold set on development data. This threshold was found for the entire development set – this system does not treat different subsets of the DIHARD II data differently.
- The Kaldi system did not use resegmentation, as this was not a part of the original Kaldi recipe at the time of the challenge.

Combined System

As our third option for the challenge, we have also implemented a combined system, which simply switches between the first two systems based on the detected corpus of each recording:

Our main diarization system works very well when faced with known corpora. As long as it is properly configured, it surpasses the performance of the Kaldi system. However, for optimal results, its parameters have to be specifically tuned for the target domain.

On the other hand, the Kaldi system is more universal. It does not use the information from the domain classifier, and its setting is very general. Therefore, it is able to work on any of the evaluation data relatively well.

For this reason, our combined system uses the following approach:

- If we can identify the corpus of a specific conversation with a sufficient certainty, we process it using our own system, with the corresponding settings.
- If we cannot identify the corpus (i.e. the domain classifier marked the conversation as "unknown corpus"), we use the more generic Kaldi system, as it is more suitable in this situation.
- In the final evaluation, we also used the Kaldi system for the two most problematic corpora – SEEDLingS and VAST. On average, our system slightly outperformed the Kaldi system on the development data for these corpora, but the DERs of individual conversations had a higher variance than the ones from Kaldi.

The combined system resulted in the lowest DER of the three options, as shown in Table 9.10.

9.4.5 Official DIHARD Evaluation Metrics

In the official DIHARD evaluation, system performance was measured in terms of DER, as well as an additional secondary metric. During DIHARD I this was Mutual Information (MI) between reference labels and those assigned by the system, while for DIHARD II, the secondary metric was Jaccard Error Rate (JER).

- To obtain Mutual Information, the reference and system labels are first converted to a sequence of individually labeled 10 ms frames. MI is then calculated as follows (Ryant et al., 2018b):

$$\mathrm{MI} = \sum_{i=1}^{N_{Ref}} \sum_{j=1}^{N_{Sys}} \frac{n_{ij}}{N_f} \log_2 \frac{n_{ij} N_f}{r_i s_j} \tag{9.5}$$

  where N_{Ref} is the number of reference speakers, N_{Sys} is the number of system speakers (i.e. the clusters created by the system), r_i is the number of frames belonging to the i-th reference speaker, s_j is the number of frames belonging to the j-th system speaker, n_{ij} is the number of frames belonging to both r_i and s_j, and N_f is the total number of frames.

- Jaccard Error Rate is new metric which was first introduced for DIHARD II (Ryant et al., 2019). Similarly to DER, calculating JER requires finding the optimal one-to-one mapping between the reference and system speakers.

  Then, for each reference speaker ref and the corresponding system speaker sys (if any), a speaker-specific Jaccard error rate JER_{ref} is calculated as

$$\mathrm{JER}_{ref} = \frac{\mathrm{FA} + \mathrm{MISS}}{\mathrm{TOTAL}} \tag{9.6}$$

  where

  - *MISS* is the duration of speech which is assigned to reference speaker ref, but not to the corresponding system speaker sys
  - *FA* is the duration of speech which is assigned to sys, but not to ref
  - *TOTAL* is the total duration of speech which is assigned to either ref or sys

  If reference speaker ref was not paired with any of the system speakers, then $\mathrm{JER}_{ref} = 1$.

  Finally, the overall JER is obtained as the average of JER_{ref} over all speakers:

$$\mathrm{JER} = \frac{1}{N_{Ref}} \sum_{ref} \mathrm{JER}_{ref} \tag{9.7}$$

In contrast to usual practice, the systems in DIHARD challenge were evaluated without any sort of tolerance collar around speaker boundaries. Overlapping speech was also evaluated.

9.4.6 Final Results in the Challenge

The DIHARD Challenge consisted of two separate tracks:

- Track 1: Diarization using reference VAD
- Track 2: Diarization using system VAD

During DIHARD I, we participated in both of these tracks, with the help of an external voice activity detector. However, we focused most of our efforts on optimizing the Track 1 system. During the second run of the challenge, we participated only in Track 1.

DIHARD II evaluation was divided into 2 phases: the initial evaluation ran for exactly one month in March–April 2019, but it was possible to submit further improvements for another three months after the first deadline. While several other teams took advantage of this extended second phase, we were unfortunately unable to devote any more time to this task.

Based on DER, our final placement was as follows:

- DIHARD I:
  - Track 1 (diarization using reference VAD): 5th place out of 14 teams
  - Track 2 (diarization from scratch): 7th place out of 11 teams
- DIHARD II:

  Track 1 only
  - First phase: 4th place out of 12 teams
  - Second phase: 11th place out of 20 teams (our system was unchanged)

Tables 9.9 and 9.10 show the official results of our submitted systems in the challenge, as well as the results of the winning teams. A full overview of the results achieved by all participating teams can be found in Tables 7.6 (page 63) and 7.7 (page 64).

Table 9.9: Official results (DER [%] and MI [bits]) on the DIHARD I evaluation data for both types of segmentation.

	Track1		**Track2**	
System	DER	MI	DER	MI
Fixed length segmentation	26.90	8.34	45.78	7.79
CNN-based speaker change detection	27.12	8.31	46.14	7.77
Track 1 winner (Team "JHU" [1])	23.73	8.44	37.19	8.04
Track 2 winner (Team "BUT" [2])	25.07	8.46	35.51	8.07

[1] Sell et al. (2018)
[2] Diez et al. (2018b)

Table 9.10: Official results (DER and JER [%]) on DIHARD II evaluation data, Track 1 only.

System	DER	JER
Proposed system	24.59	49.63
Kaldi system	25.17	54.94
Combined system	23.47	48.99
Phase 1 winner (Team "DI-IT" [1])	21.62	48.82
Phase 2 winner (Team "BUT" [2])	18.42	44.58

[1] Novoselov et al. (2019)
[2] Landini et al. (2020)

9.4.7 Discussion

Table 9.11 shows the results of our original system on each corpus in the DIHARD II development set, as well as the results of several experimental modifications: with simple VAD-based segmentation instead of speaker change detection, with denoised input data, and with information about overlapping speech.

- Alternative segmentation

  The alternative VAD-based segmentation was the same "fixed-length" option which we had previously used for DIHARD I. From the results here we can see that the new combined approach provides better results in almost all cases.

- Denoising

  Some of the DIHARD data contain a relatively large amount of noise. During DIHARD I, denoising was succesfully used by team USTC-iFLYTEK (Sun et al., 2018b) and their speech enhancement tool[5] was later made available to all participants of DIHARD II.

  We have explored the use of this tool in our own system. However, the tool itself appears to be primarily intended for improving the accuracy of VAD, and as shown in Table 9.11, did not prove beneficial for our system in Track 1.

- Overlapping speech

  Challenge data contained a large amount of overlapping speech. This was a significant source of error and accurate detection could improve the results, as shown during DIHARD I by teams such as BUT (Diez et al., 2018b).

  To investigate the potential benefits for our system, we first tested it with reference overlap labels and the following simple modification: Intervals with overlapping speech are excluded from clustering. During resegmentation, these intervals are assigned the labels of the two most similar GMMs, rather than only one.

[5] https://github.com/staplesinLA/denoising_DIHARD18

As shown in Table 9.11, we have found that with perfect overlap detection, we could decrease our DER on the development set of DIHARD II from 20.78% to 16.16%.

Additionally, the final column of the table shows baseline results, which would be achieved by assigning the same label to all speech. We can see that for most of the corpora, the system achieves a significantly better DER than the baseline. The only exceptions are SEEDLingS and VAST, where the difference is rather small, suggesting that the system does not work well on these data. This likely has to do with the character of these two corpora – SEEDLingS contains recordings of children between 6 and 18 months old, who have very different voices from adult speakers. VAST originates from unrelated web videos, recorded under different conditions, so there does not appear to be a single system configuration which would work well for all recordings.

Overall, our system placed fairly well in the challenge, but as the results of the winning teams show, there is still room for improvement.

Table 9.11: Average DER [%] on individual corpora of the DIHARD II development set (Track 1), for our system (without Kaldi) and for several different modifications – with Kaldi VAD-based segmentation instead of speaker change detection (SCD), with denoised data, and with reference overlap labels – as well as baseline results where all speech is given the same label.

Corpus	**our system**	**without SCD**	**with de-noise**	**with ref. overlap**	**all one speaker**
LibriVox	0.00	0.00	0.00	0.00	0.00
SEEDLingS	31.32	31.22	32.30	24.56	36.38
CIR	45.83	47.88	46.70	37.71	66.49
ADOS	14.06	13.26	14.25	10.73	39.64
SCOTUS	6.92	10.67	8.01	5.99	47.56
DCIEM	8.88	8.66	8.74	6.24	29.39
RT-04S	33.14	36.38	34.53	25.69	54.70
SLX	17.56	19.14	17.36	13.64	37.98
MIXER6	9.42	9.29	9.93	5.02	33.08
VAST	38.00	38.91	38.61	30.09	41.11
YouthPoint	4.55	5.26	5.49	3.89	24.61
All	20.78	21.52	21.31	16.16	37.47

9.5 Overlap Detection Using a Convolutional Neural Network (CNN)

This section describes a separate set of experiments which focused on the detection of overlapping speech. As discussed in chapter 8, overlapping speech is a major source of error in speaker diarization, and its correct handling can lead to significant improvements. This was also observed in Table 9.11, where oracle overlap labeling reduced the proposed system's DER from 20.78 % to 16.16 %.

The main idea of these experiments was inspired by the CNN-based speaker change detection used in (Hrúz and Kunešová, 2016), which is also mentioned in section 9.3.3. The proposed overlap detector uses a very similar network architecture and the same general approach.

As the overlap detector processes the input data sequentially, with a simple sliding window, it can be used for both online and offline diarization.

Parts of this section have also been published in (Kunešová, 2018) and (Kunešová et al., 2019).

9.5.1 The Overlap Detector

The overlap detector is implemented as a Convolutional Neural Network. Its architecture is summarized in Table 9.12.

The input of the network is a spectrogram of a short window of the acoustic signal. The output of the last layer is a value between 0 and 1, indicating the probability of overlapping speech in the middle of the window. Training references use a fuzzy labeling function, with a linear slope at the boundaries between overlap and non-overlap, as shown in Figure 9.7. The sliding window has a length of 1 s and is shifted with a step of 0.05 s.

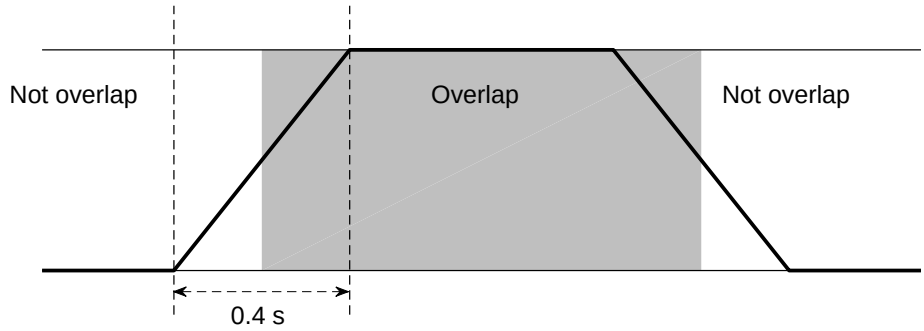

Figure 9.7: Reference signal for CNN training. Gray area denotes the interval with overlapping speech.

A median filter with a window length of 5 samples is used to smooth the raw network output, then a threshold is applied to obtain overlap / non-overlap classification. Additionally, the post-processing fills in any gaps (non-overlaps within a longer overlap) which are shorter than 0.1 s, and then discards overlaps under

Table 9.12: Summary of the architecture of the CNN.

Layer	Kernels	Size	Shift
Convolution	128	8 x 16	2 x 2
Max pooling		2 x 2	2 x 2
Batch Norm			
Convolution	256	4 x 4	1 x 1
Max pooling		2 x 2	2 x 2
Batch Norm			
Convolution	512	3 x 3	1 x 1
Max pooling		2 x 2	2 x 2
Batch Norm			
Fully Connected	1024		
Fully Connected	256		
Fully Connected	1		

0.5 s, as these are unlikely to be included in the reference labeling (as discussed in section 8.3).

9.5.2 Synthetic Training Data for Overlap Detection

Given the lack of sufficient real data (as mentioned in section 8.3), training data for experiments were artificially created from two corpora of read English speech – LibriSpeech (Panayotov et al., 2015) and TIMIT (Garofolo et al., 1993) – using an automated and randomized process. Some of the ideas used in the creation of this synthetic dataset were inspired by those used by Edwards et al. (2018) and Sajjan et al. (2018).

Both of the corpora consist of a large amount of short recordings from many different speakers.

TIMIT The TIMIT corpus consists of the recordings of single English sentences, approx. 2–5 s long. The data from 320 speakers were used for training.

To obtain overlapped data, all utterances from a single speaker are first concatenated into one file of approx. 30 s, with random-length pauses (up to 2 s) in-between. In order to avoid noticeable seams, the silence at the beginning and end of each utterance is linearly tapered. Then, files from two random speakers are combined at different volumes and with added background noise and, for 50 % of the files, also reverberation (see the end of this section for details). The result is illustrated in Figure 9.8.

Reference labels were created with the use of the original phone-level transcripts - so that only the intervals where both speakers are truly active are labeled as overlap.

The same process was also used to generate *test* data from the LibriSpeech

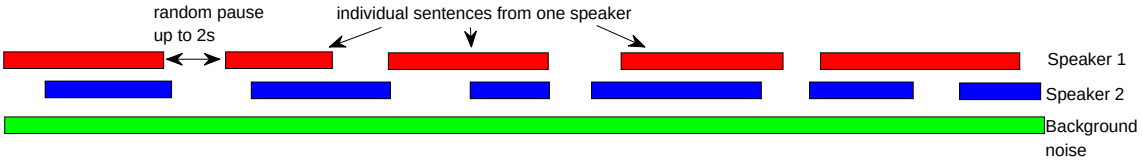

Figure 9.8: Illustration of the creation of artificial overlapped data from the TIMIT corpus.

corpus (but with pauses of 5–10 s between utterances).

LibriSpeech The second part of the training data was created from the "train-other-500" set of the LibriSpeech corpus – this consists of approx. 500 hours of speech from over 1000 speakers, in the form of 10–15 s long recordings derived from audiobooks.

Given the very large amount of available LibriSpeech data, it was possible to create several different types of overlaps, in order to better represent the various possibilities which may occur in real data (see Fig. 9.9):

a) Two full length (approx. 10–15 s) utterances, with a long overlap of 1/2 the length of the first one

b) Two shortened utterances with a brief overlap (up to 2 s) or pause (up to 1 s) in-between

c) A single utterance with an inserted word or phrase from another speaker:

   Utterance 1 is split on pauses and a randomly selected speech interval (0.25–2 s) is placed over utterance 2:

   - fully overlapping speech
   - fully inside a pause
   - random placement

In the case of b) and c), the resulting file is shortened to 5 s of non-overlap data on each side of the overlap or pause, as seen in Fig. 9.9. The is done to keep a better ratio between non-overlaps and overlaps.

As with TIMIT, the silence at the beginning and end of each utterance is linearly tapered to avoid discernible seams.

Reference labels were created with the use of a VAD on the original single speaker data without added noise – so that only the intervals where both speakers are truly active are labeled as overlap.

Data augmentation For both of the corpora, the data were also augmented with additional effects:

- A very small amount of added white noise – serves merely to avoid exact zeros in artificial pauses

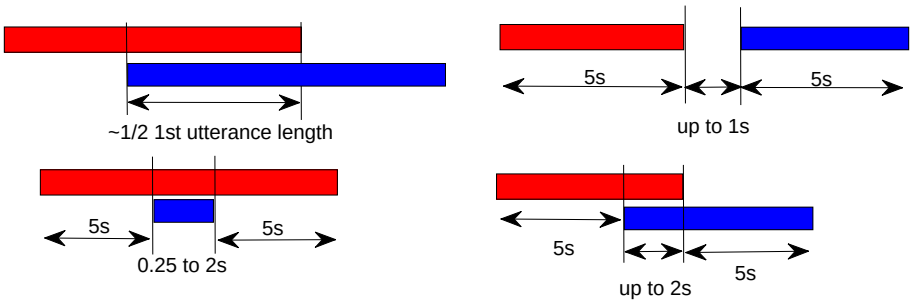

Figure 9.9: Illustration of the creation of training data with different types of overlap from the LibriSpeech corpus. (Additional noise not shown.)

- Additive noise ("office", "hallway" and "meeting") from the DEMAND database (Thiemann et al., 2013a)
- Reverberation – via convolution with room impulse response from the AIR database (Jeub et al., 2009). This was applied for 50 % of the resulting files.

9.5.3 Test Data

The overlap detector was primarily evaluated on three different sets of data: one artificially created dataset and two corpora of real conversations.

LibriSpeech Test Data

Synthetic test data were created from the "test-other" subset of the LibriSpeech corpus, in a very similar way to the TIMIT training data – but with longer pauses between a single speaker's utterances (5–10 s) on account of greater utterance length.

Figure 9.10 shows an example of the system's output on this corpus.

SSPNet Conflict Corpus

The SSPNet Conflict Corpus[6] (Kim et al., 2014) is a dataset of French-language political debates, consisting of 1430 clips of 30 s each, cut from 45 separate debates. Each clip usually involves between 2–5 people and, as these are spontaneous discussions, there are frequent instances of overlapping speech. The same corpus was also used for overlap detection by Kazimirova and Belyaev (2018).

Clips from 5 debates (06-05-31, 06-09-20, 06-10-11, 07-05-16, and 08-01-15; 161 files total = 80.5 minutes of audio data) were used as development data for tuning the decision threshold, the remainder (1269 files = 10.6 hours) was used for evaluation.

[6] https://web.archive.org/web/20180313145831/http://www.dcs.gla.ac.uk/vincia/?p=270 (archived webpage)

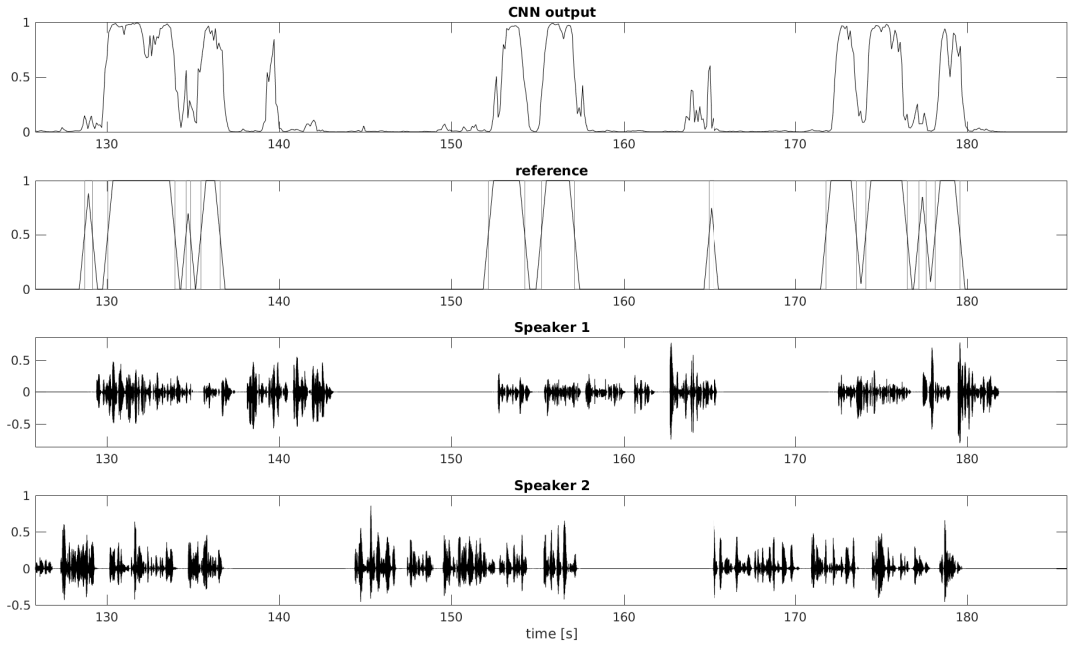

Figure 9.10: Example of the CNN's output for a LibriSpeech test file with unseen speakers (top), the corresponding reference labels (second plot) and each speaker's soundwave.

As the corpus hadn't been created with overlaps in mind, the original reference labels are relatively rough in this regard – they do not include very short overlaps at speaker changes or during isolated backchannel responses (e.g. "Oui, ... oui."), nor shorter non-overlap intervals within a longer overlap region (e.g. pauses in the speech of one speaker). However, the network proved capable of detecting all of the above. For this reason, the labels of a small number of audio clips were manually corrected[7] to better correspond to the audio data (see Fig. 9.11 for an example). Due to the time-consuming nature of such precise corrections, it was limited to 30 semi-randomly selected files, or a total of 15 minutes. These 30 files were then evaluated separately, using both the original and corrected labels, to illustrate how labeling quality affects the reported results (see Table 9.13).

AMI Meeting Corpus

The AMI Meeting Corpus, described in section 9.1.3, is a set of recordings from meetings between 3–5 people. The overlap detector was tested on the "headset mix" variant of the data, using the same train/validation/test split as Sajjan et al. (2018). Ground truth labels were generated from the original transcripts, rather than using Sajjan et al. (2018)'s force-aligned labels[8], as initial testing found the latter to be less accurate in some regards, but both versions have errors – in particular, there are many instances where overlaps with non-speech such as laughter are not labeled.

The corpus consists of several subsets of meetings which were recorded at dif-

[7]The corrected labels can be found at: https://github.com/mkunes/CNN-overlap-detection
[8]Available on: https://github.com/BornInWater/Overlap-Detection

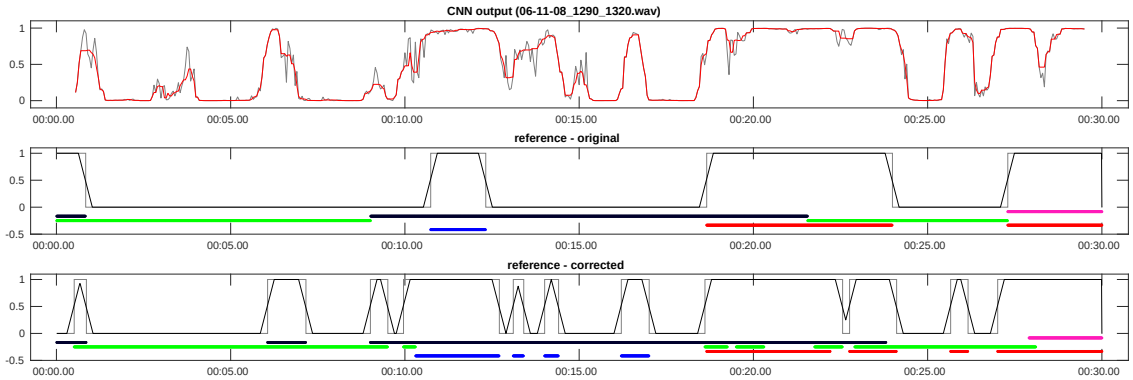

Figure 9.11: Example output (raw + median filter) for dereverberated SSPNet data (top) and the corresponding reference labels – original (middle) and manually corrected (bottom). Individual speakers are shown as colored bars at the bottom of the latter two plots.

ferent sites and vary in audio and transcription quality. The Idiap scenario meetings (IS) in particular appear to have very different optimal settings from the rest of the test set, so they are also used separately.

For this Idiap-only evaluation, the development and test data included additional AMI files, which were not used by Sajjan et al. The full list of the used "IS" files was as follows:

Development set: IS1001, IS1003 and IS1004 (a,b,c,d)

Evaluation set: IS1005 (a,b,c); IS1006, IS1007, IS1008 and IS1009 (a,b,c,d)

9.5.4 Evaluation

As seen in Table 8.1, previous works on overlap detection use a variety of different evaluation metrics, such as frame-level precision, recall and F-score, or per-overlap miss and false alarm rate. Others evaluate their systems indirectly, by measuring the improvement in the DER of a diarization system.

As the main motivation here is to improve speaker diarization, it seems to be more suitable to evaluate the overlap detector primarily in terms of the improvement in diarization performance. However, using only the actual difference in Diarization Error Rate is not ideal, as it depends not only on the overlap detector, but also on the diarization system itself.

Thus it may also be useful to calculate merely the *potential* or expected gain, which could be theoretically achieved given an otherwise perfect diarization system (i.e. one which does not make any mistakes):

As described in section 7.1, DER consists of three types of error: *missed speech* (including missing speakers in overlaps), *false alarm* (silence mislabeled as speech or non-overlap as overlap), and *speaker confusion* (wrong speaker). In a perfect diarization system with no overlap handling, false alarm and speaker confusion will be zero, while missed speech will correspond to the amount of overlapping

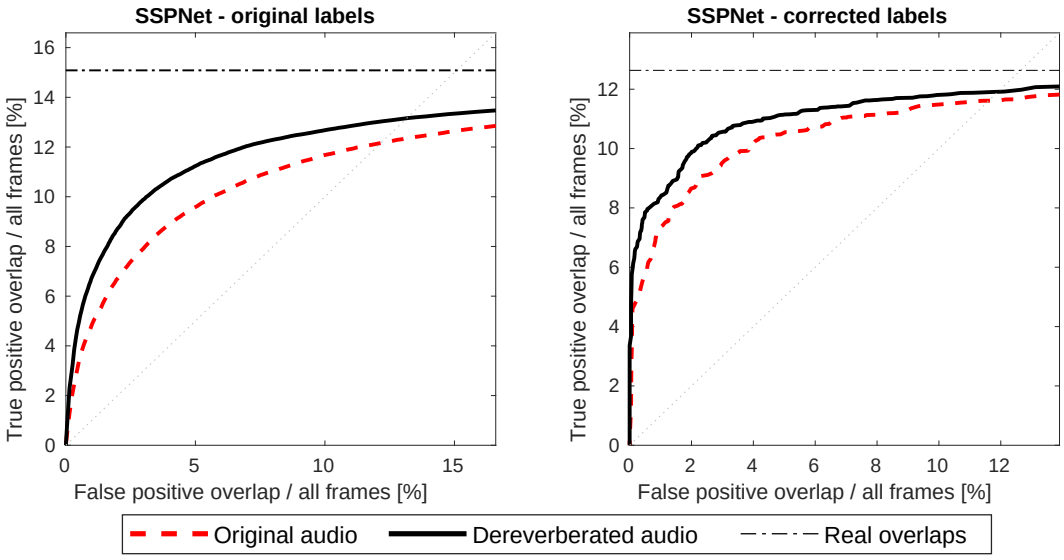

Figure 9.12: False Positive vs True Positive for SSPNet data (frame-level percentage of all audio). Original labels (all 1269 test files) on the left, corrected labels (30 files, 15 minutes total) on the right. "Real overlaps" denotes the overlap percentage in the ground truth.

speech in the data.

For calculating the potential gain from overlap detection, the following assumptions will be used:

- The diarization system assigns two speaker labels to every detected region of overlapping speech, regardless of the true number speakers
- For correctly detected overlap and non-overlap, the system assigns speaker labels perfectly – the *speaker confusion* is zero.

In such a scenario, correctly detected overlaps will directly decrease the amount of *missed speech* compared to the baseline system, while false overlaps will increase the *false alarm*. Thus, we can estimate the potential improvement as the difference between the two values.[9]

9.5.5 Results

The results achieved on the different corpora are shown in Table 9.13 and in Figures 9.12 and 9.13.

The overlap detector appears to work very well on clean audio, such as the synthetic LibriSpeech data and the SSPNet Conflict Corpus. The network also seems to be very sensitive and capable of detecting even very short overlaps and non-overlaps, down to the level of individual words – a much greater precision

[9]By the correct definition, DER is calculated as a ratio of total speech (excluding silence), with overlaps being counted multiple times – once for each speaker. However, for simplicity, we calculate the potential improvements here as relative to the total length of the audio data.

Table 9.13: Results of overlap detection on evaluation data. Overlap percentages are relative to total audio length, precision and recall are calculated per frame. (Ref. = Reference, TP = True Positive, FP = False Positive, Δ = TP - FP). Updated since the publication of (Kunešová et al., 2019).

	Overlaps [% of all frames]						
Dataset	Ref.	TP	FP	Δ	Prec.	Rec.	Thresh.
LibriSpeech test mix	16.32	11.99	2.82	**9.18**	0.81	0.73	0.25
SSPNet – original labels (all)	14.77	7.86	2.94	4.92	0.73	0.52	0.80
+ dereverberation		9.58	2.68	**6.90**	0.78	0.63	0.70
SSPNet – corrected l. (30 files)	12.62	8.05	1.42	6.63	0.85	0.65	0.80
+ dereverberation		8.90	1.41	**7.49**	0.86	0.71	0.70
SSPNet – original l. (30 files)	12.86	7.47	2.00	5.47	0.79	0.59	0.80
+ dereverberation		8.60	1.71	**6.89**	0.83	0.68	0.70
AMI test (all subsets - 16 files)	12.21	2.25	0.96	**1.30**	0.70	0.19	0.50
+ dereverberation		1.94	0.70	1.25	0.74	0.16	0.40
AMI test ("IS" - 10 files)	10.43	3.62	1.65	1.97	0.69	0.35	0.80
+ dereverberation		4.62	1.87	**2.76**	0.71	0.45	0.60
Retrained network – with added AMI training data:							
AMI test (all subsets)	12.21	5.48	2.08	**3.40**	0.72	0.46	0.50
+ dereverberation		4.92	1.61	3.31	0.75	0.41	0.50
AMI test ("IS" files)	10.43	4.21	1.61	2.60	0.72	0.41	0.70
+ dereverberation		4.29	1.40	**2.90**	0.75	0.42	0.70

than typically found in the reference annotations (as illustrated by the example output in Figure 9.11).

On the other hand, the detector had issues with the AMI corpus. This may be in part due to errors in the reference labels – a closer inspection revealed instances of missing speech, or long unlabeled intervals where multiple people are laughing, which the network also considers to be overlaps. However, the lower performance is likely also caused by the higher level of noise in the these recordings, as well as the sometimes very large differences in the voice volumes of individual speakers. This is evidenced by the fact that the results improved to some extent after including the training set of the AMI corpus in the training data – this suggests that the synthetic dataset may need more improvements.

Initial experiments also suggested that the network had problems with reverberant speech, which was often incorrectly labeled as overlap. This effect has been partly mitigated by adding reverberation to the training data (as described in section 9.5.2). However, there have also been some additional experiments with dereverberation of the test data – the potential benefits were evaluated with the use

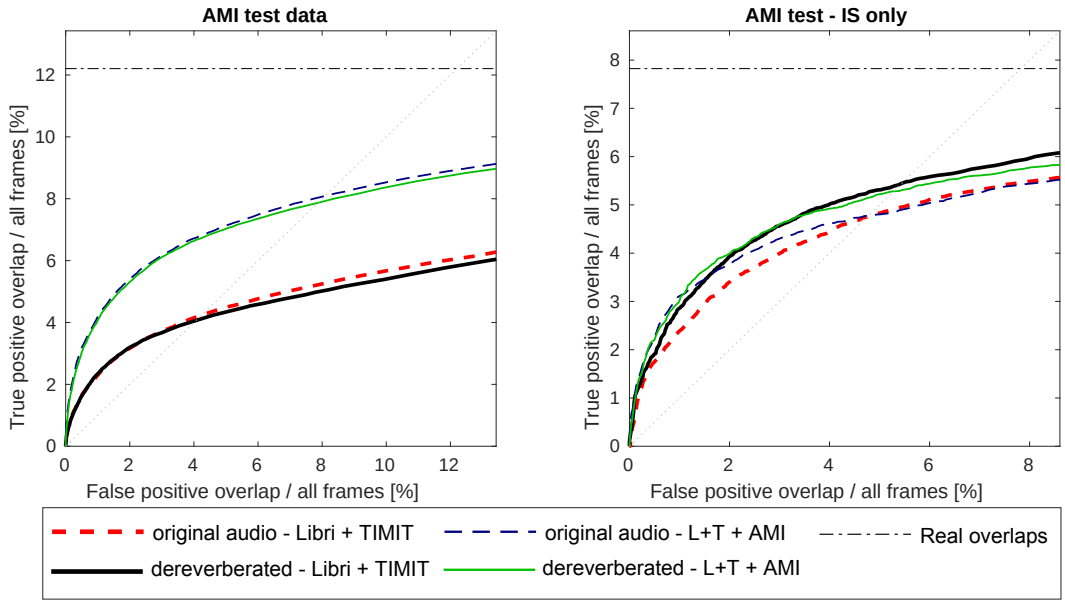

Figure 9.13: False Positive vs True Positive for AMI data (frame-level percentage of all audio), with overlap detector trained only on synthetic LibriSpeech + TIMIT data or with the addition of AMI training data. Results are for all test files (left) and only for the Idiap scenario meetings (right).

of the WPE Speech Dereverberation package[10] created by Nakatani et al. (2010). Even with the default settings without any adjustments, this has proven to be clearly beneficial for SSPNet data, but for AMI the difference is negligible (with the exception of the Idiap scenario (IS) meetings).

Finally, Table 9.14 presents a comparison of the overlap detector with some of the other works on the topic which were listed in Table 8.1. This comparison is somewhat complicated by the fact that other authors have used many different combinations of datasets (or their parts) and metrics to evaluate their systems. For instance, while 4 other systems in the table used the AMI corpus, each of them selected different files. Similarly, the results of Kazimirova and Belyaev (2018) on the SSPNet Conflict Corpus are not directly comparable with those of

[10] Available from: http://www.kecl.ntt.co.jp/icl/signal/wpe/index.html

the proposed system, as their system was evaluated only on voiced frames.

Table 9.14: Comparison of the proposed overlap detection system (selected results from Table 9.13, trained only on the synthetic corpus) with similar works. With the exception of the proposed "all subsets" and Sajjan et al.'s "original labels" AMI results, no two systems used identical test data and ground-truth labeling. However, the final set of results, included at the bottom, also shows the performance of the version of the system with added AMI training data when evaluated under identical conditions to system [6].

System	Dataset	Prec.	Rec.	F-score	Accuracy
proposed	SSPNet (original labels)	0.73	0.52	0.61	0.90
	+ dereverberation	0.78	0.63	0.70	0.92
	AMI (all subsets – 16 files)	0.70	0.19	0.30	0.89
	+ dereverberation	0.74	0.16	0.27	0.89
	AMI (19 "IS" files)	0.69	0.35	0.47	0.92
	+ dereverberation	0.71	0.45	0.55	0.92
[1]	Custom dataset	0.81	0.78	0.8	0.802
[2]	SSPNet (voiced frames only)	0.71	0.78	0.75	0.92
[3]	AMI (12 "IS" files, force aligned)	0.67	0.26	0.38	–
[4]	AMI (16 files, original labels)	–	–	–	76.0 / 60.6*
	AMI (16 files, force aligned)	–	–	–	87.9 / 71.0*
[5]	AMI (25 files)	–	–	0.51	–
[6]	AMI (24 files)	0.868	0.658	–	–
proposed	AMI (identical to [6])**				
	+ AMI train data, dereverb.	0.758	0.446	–	–

[1] Andrei et al. (2017)
[2] Kazimirova and Belyaev (2018)
[3] Boakye et al. (2008b)
[4] Sajjan et al. (2018)
[5] Yella and Bourlard (2014)
[6] Bullock et al. (2020)

* overlap-detection accuracy / single-speaker detection accuracy [%]

** used identical reference labels, test set and evaluation script as in [6]. Obtained with the help of Hervé Bredin and also seen as the baseline in the aforementioned paper.

The overall results achieved here appear to be very promising, particularly those on relatively clean and noise-free data, although some more work may be required in order to improve the performance on data with higher levels of noise.

Chapter 10

Conclusion

This thesis focused on the topic of speaker diarization. It explored several diarization approaches, both online and offline, and also proposed a new method for detecting overlapping speech, which is very relevant for diarization.

As an additional contribution, chapter 7 of the thesis also included an extensive overview of many of the previous diarization systems which have appeared in literature and compared their reported results.

Speaker Diarization

The work on speaker diarization focused on two very different principles: the earliest experiments involved a GMM-based online diarization system, while later work progressed to the more recent i-vectors and x-vectors.

The GMM-based online system, presented in section 9.2, initially started as a re-implementation of a sequential clustering approach used by several other authors (see section 4.3.1), but subsequent work focused on providing improvements. The main contribution here was the introduction of a new merging process which can identify similar speaker models on the fly and combine them back into a single label. This helps to alleviate a common issue where the system would create several different models for the same speaker.

The system was specifically intended for the diarization of recordings from the Czech parliament sessions and the results on these data were also published as part of two conference papers (Campr et al., 2014; Kunešová and Radová, 2015). Later, the system was also tested on a very different dataset of conversational data (the AMI corpus), though this was unfortunately significantly less successful. Finally, one more experiment with the AMI data also considered the possibility of acquiring speaker models in advance and then performing diarization via speaker identification, akin to the systems mentioned in section 4.3.2. Despite its inherent limitation, this approach showed promise, achieving significantly lower error rates than the original system. These same experiments, described in section 9.2.4, also explored the impact that overlapping speech has on speaker diarization, and showed that its proper handling could provide a noticeable benefit.

Overall, the GMM-based system still offers room for improvement, particularly in situations where there is mismatch between the system's UBM and the test data, such as was in the case of the AMI corpus. However, in the time since this part of the research was first initiated, the field of speaker diarization has gradually shifted from the use of GMMs as speaker models and towards more modern

solutions, and as such, there appears to be little point in following this particular direction any further. Instead, later work was focused on the more recent i-vectors and x-vectors.

Thus, the second part of the diarization research mostly involved an *i-vector based diarization system*, which was created in collaboration with Z. Zajíc (Zajíc et al., 2016). The initial system, described in section 9.3.1, performed offline diarization, but a modified online version was implemented as well (Kunešová et al., 2017) and was presented in section 9.3.2.

Some of the experimental work with the system focused on the segmentation step of speaker diarization. In section 9.3.3 and 9.3.4, we investigated the commonly held belief that conventional speaker change detection is largely unnecessary in modern offline diarization systems. The results we obtained from comparing several different segmentation methods suggest this to be mostly true, though there is still clearly room for improvement if a more accurate speaker change detector could be obtained.

Besides the basic offline and online versions of the system, a hybrid online/offline variant (see section 4.3.3) was also independently explored. As described in section 9.3.5, this was implemented as standard sequential clustering with a periodic reclustering process applied to all past data. While the initial implementation was relatively simple, it showed clear improvements compared to the baseline system without reclustering.

Finally, as part of the shared work on the offline version of the system, we also participated in the DIHARD Speaker Diarization Challenge (section 9.4, also published as Zajíc et al. (2018) and Zajíc et al. (2019)). In the first iteration of this international competition, we managed to achieve 5th place out of the 14 participating teams, in part thanks to our unique domain classifier which allowed us to better fine-tune our system. However, as the second run of the competition showed, there is still much to improve if we want to catch up to the winners.

During most of the above-mentioned experiments, the system used i-vectors for speaker representation. However, the majority of the methods can be equally easily applied to x-vectors (or similar embeddings) as well. In the final version which was used during the DIHARD challenge (section 9.4.3), we employed both i-vectors and x-vectors, and the combination of the two offered a better performance than either option alone.

Overlapping Speech Detection

Several of the experiments with speaker diarization also explored the influence of overlapping speech on the results by using oracle overlap labels. In all cases, there was a stark difference between the diarization performance with and without overlap handling. Thus, the final part of the thesis focused on the detection of such overlaps.

The main contribution here was a newly proposed approach to overlapping speech detection, based on the use of a convolutional neural network with a spectrogram as its input (section 9.5, also published as Kunešová et al., 2019). The

initial idea arose from the CNN-based speaker change detection method which previously appeared in section 9.3.3 as one of the segmentation options in the i-vector diarization system.

In order to train the CNN, it was necessary to obtain sufficiently accurate training data. As there were no appropriate available corpora, this also led to the creation of a new synthetic dataset, composed of artificially created overlaps of several different types. As described in section 9.5.2, the dataset was generated from single-speaker recordings from the LibriSpeech and TIMIT corpora, with added noise and reverberation effects.

The overlap detector was tuned specifically with the goal of lowering the diarization error rate, and was tested both on synthetic overlaps and on real data from the SSPNet Conflict Corpus and the AMI corpus. The overlap detector appears to be sensitive to reverberation, although this can be improved by applying dereverberation techniques to the test data. Overall, the detector achieves similar results to those of other recent works, although it is difficult to directly compare.

Summary of Contributions

In summary, this thesis provides the following contributions:

- Overview of current literature on the topic of speaker diarization and detection of overlapping speech, including an extensive comparison of the reported performance of recent state-of-the-art systems
- Implementation of a GMM-based online diarization system and its further improvements
- Experimental work with an offline i-vector based system and its extension to online diarization

  The system in question also placed relatively well in the DIHARD Speaker Diarization Challenge
- Comparison of several segmentation options for speaker diarization
- Implementation of a hybrid online/offline diarization approach
- Proposed approach for detecting overlapping speech using a CNN
- Creation of a synthetic dataset for training the aforementioned overlap detector

Future Work

The work in this thesis still offers several possible avenues for further improvement.

In particular, the hybrid diarization concept which was briefly explored in section 9.3.5 appears both promising and relatively uncommon, and any further research would likely pursue a similar direction. Future work could also aim at

applying this principle to some of the cutting-edge diarization methods which appeared during the late stages of work on this thesis and could not be included here – such as end-to-end diarization.

Likewise, the overlap detector (section 9.5) could be further improved, such as by providing a greater variety of training data and improving the training process, or even by extending its functionality to also act as a voice activity detector. The network's weakness to reverberant speech is also an aspect which could hopefully be alleviated in the future.

Bibliography

Ajmera, J., Bourlard, H., Lapidot, I., and McCowan, I. A. (2002). Unknown-multiple speaker clustering using HMM. In: *Proc. ICSLP 2002*, pp. 573–576. URL: https://www.isca-speech.org/archive/icslp_2002/i02_0573.html.

Andrei, V., Cucu, H., and Burileanu, C. (2017). Detecting overlapped speech on short timeframes using deep learning. In: *Proc. Interspeech 2017*, pp. 1198–1202. DOI: 10.21437/Interspeech.2017-188.

Anguera, X. and Bonastre, J.-F. (2010). A novel speaker binary key derived from anchor models. In: *Proc. Interspeech 2010*, pp. 2118–2121. URL: https://www.isca-speech.org/archive/interspeech_2010/i10_2118.html.

Anguera, X., Wooters, C., and Hernando, J. (2007). Acoustic beamforming for speaker diarization of meetings. In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.7, pp. 2011–2022. DOI: 10.1109/TASL.2007.902460.

Anguera, X. et al. (2012). Speaker diarization: a review of recent research. In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.2, pp. 356–370. DOI: 10.1109/TASL.2011.2125954.

Anguera Miró, X. (2007). Robust Speaker Diarization for Meetings. PhD thesis. Barcelona: Universitat Politècnica de Catalunya. HDL: 2117/94212.

Aronowitz, H., Zhu, W., Suzuki, M., Kurata, G., and Hoory, R. (2020). New advances in speaker diarization. In: *Proc. Interspeech 2020*, pp. 279–283. DOI: 10.21437/Interspeech.2020-1879.

Atal, B. S. and Hanauer, S. L. (1971). Speech analysis and synthesis by linear prediction of the speech wave. In: *The Journal of the Acoustical Society of America* 50.2B, pp. 637–655. DOI: 10.1121/1.1912679.

Bergelson, E. (2016). *Bergelson Seedlings HomeBank Corpus*. DOI: 10.21415/T5PK6D.

Boakye, K., Trueba-Hornero, B., Vinyals, O., and Friedland, G. (2008a). Overlapped speech detection for improved speaker diarization in multiparty meetings. In: *Proc. ICASSP 2008*. IEEE, pp. 4353–4356. DOI: 10.1109/ICASSP.2008.4518619.

Boakye, K., Vinyals, O., and Friedland, G. (2008b). Two's a crowd: improving speaker diarization by automatically identifying and excluding overlapped speech. In: *Proc. Interspeech 2008*, pp. 32–35. URL: https://www.isca-speech.org/archive/interspeech_2008/i08_0032.html.

ten Bosch, L., Oostdijk, N., and Boves, L. (2005). On temporal aspects of turn taking in conversational dialogues. In: *Speech Communication* 47.1-2, pp. 80–86. DOI: 10.1016/j.specom.2005.05.009.

Bozonnet, S., Evans, N. W. D., and Fredouille, C. (2010). The LIA-EURECOM RT'09 speaker diarization system: enhancements in speaker modelling and cluster purification. In: *Proc. ICASSP 2010*. IEEE, pp. 4958–4961. DOI: 10.1109/ICASSP.2010.5495088.

Bozonnet, S., Vipperla, R., and Evans, N. (2012). Phone adaptive training for speaker diarization. In: *Proc. Interspeech 2012*, pp. 494–497. URL: https://www.isca-speech.org/archive/interspeech_2012/i12_0494.html.

Bozonnet, S., Wang, D., Evans, N., and Troncy, R. (2011). Linguistic influences on bottom-up and top-down clustering for speaker diarization. In: *Proc. ICASSP 2011*. IEEE, pp. 4424–4427. doi: 10.1109/ICASSP.2011.5947335.

Bredin, H. (2017a). pyannote.metrics: a toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems. In: *Proc. Interspeech 2017*, pp. 3587–3591. doi: 10.21437/Interspeech.2017-411.

Bredin, H. (2017b). TristouNet: triplet loss for speaker turn embedding. In: *Proc. ICASSP 2017*, pp. 5430–5434. doi: 10.1109/ICASSP.2017.7953194.

Bredin, H. and Gelly, G. (2016). Improving speaker diarization of TV series using talking-face detection and clustering. In: *Proc. 24th ACM Int. Conf. on Multimedia*, pp. 157–161. doi: 10.1145/2964284.2967202.

Broux, P.-A. et al. (2018). S4D: speaker diarization toolkit in Python. In: *Proc. Interspeech 2018*, pp. 1368–1372. doi: 10.21437/Interspeech.2018-1232.

Bullock, L., Bredin, H., and Garcia-Perera, L. P. (2020). Overlap-aware diarization: resegmentation using neural end-to-end overlapped speech detection. In: *Proc. ICASSP 2020*, pp. 7114–7118. doi: 10.1109/ICASSP40776.2020.9053096.

Campbell, E. L., Hernandez, G., and Calvo de Lara, J. R. (2018). CENATAV voicegroup systems for Albayzin 2018 speaker diarization evaluation campaign. In: *Proc. IberSPEECH 2018*, pp. 227–230. doi: 10.21437/IberSPEECH.2018-47.

Campbell, W. M., Sturim, D. E., and Reynolds, D. A. (2006). Support vector machines using GMM supervectors for speaker verification. In: *IEEE Signal Processing Letters* 13.5, pp. 308–311. doi: 10.1109/LSP.2006.870086.

Campr, P., Kunešová, M., Vaněk, J., Čech, J., and Psutka, J. (2014). Audio-video speaker diarization for unsupervised speaker and face model creation. In: *Proc. TSD 2014*. Springer, pp. 465–472. doi: 10.1007/978-3-319-10816-2_56.

Canavan, A., Graff, D., and Zipperlen, G. (1997). *CALLHOME American English Speech*. Linguistic Data Consortium. ldc: LDC97S42.

Carletta, J. et al. (2006). The AMI meeting corpus: a pre-announcement. In: *Machine Learning for Multimodal Interaction*. Ed. by S. Renals and S. Bengio. Springer, pp. 28–39. doi: 10.1007/11677482_3.

Castaldo, F., Colibro, D., Dalmasso, E., Laface, P., and Vair, C. (2008). Stream-based speaker segmentation using speaker factors and eigenvoices. In: *Proc. ICASSP 2008*. IEEE, pp. 4133–4136. doi: 10.1109/ICASSP.2008.4518564.

Castan, D., McLaren, M., and Nandwana, M. K. (2018). The SRI international STAR-LAB system description for IberSPEECH-RTVE 2018 speaker diarization challenge. In: *Proc. IberSPEECH 2018*, pp. 208–210. doi: 10.21437/IberSPEECH.2018-42.

Charlet, D., Barras, C., and Lienard, J.-S. (2013). Impact of overlapping speech detection on speaker diarization for broadcast news and debates. In: *Proc. ICASSP 2013*. IEEE, pp. 7707–7711. doi: 10.1109/ICASSP.2013.6639163.

Chen, S. S. and Gopalakrishnan, P. S. (1998). 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.

Cornell, S., Omologo, M., Squartini, S., and Vincent, E. (2020). Detecting and counting overlapping speakers in distant speech scenarios. In: *Proc. Interspeech 2020*, pp. 3107–3111. doi: 10.21437/Interspeech.2020-2671.

Davis, S. B. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.4, pp. 357–366. doi: 10.1109/TASSP.1980.1163420.

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., and Ouellet, P. (2011). Front-end factor analysis for speaker verification. In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.4, pp. 788–798. doi: 10.1109/TASL.2010.2064307.

Delacourt, P. and Wellekens, C. J. (2000). DISTBIC: a speaker-based segmentation for audio data indexing. In: *Speech communication* 32.1, pp. 111–126. doi: 10.1016/S0167-6393(00)00027-3.

Delgado, H., Anguera, X., Fredouille, C., and Serrano, J. (2015). Fast single- and cross-show speaker diarization using binary key speaker modeling. In: *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 23.12, pp. 2286–2297. doi: 10.1109/TASLP.2015.2479043.

Diez, M., Burget, L., and Matejka, P. (2018a). Speaker diarization based on Bayesian HMM with eigenvoice priors. In: *Proc. Odyssey 2018*, pp. 147–154. doi: 10.21437/Odyssey.2018-21.

Diez, M. et al. (2018b). BUT system for DIHARD speech diarization challenge 2018. In: *Proc. Interspeech 2018*, pp. 2798–2802. doi: 10.21437/Interspeech.2018-1749.

Dighe, P., Ferras, M., and Bourlard, H. (2014). Detecting and labeling speakers on overlapping speech using vector Taylor series. In: *Proc. Interspeech 2014*, pp. 592–596. url: https://www.isca-speech.org/archive/interspeech_2014/i14_0592.html.

Dimitriadis, D. and Fousek, P. (2017). Developing on-line speaker diarization system. In: *Proc. Interspeech 2017*, pp. 2739–2743. doi: 10.21437/Interspeech.2017-166.

Edwards, E. et al. (2018). A free synthetic corpus for speaker diarization research. In: *Proc. SPECOM 2018*. Springer, pp. 113–122. doi: 10.1007/978-3-319-99579-3_13.

Fischerová, P. (2007). Detekce změny řečníka v řečovém signálu. [Speaker change detection]. [In Czech]. PhD thesis. Plzeň: Západočeská univerzita v Plzni.

Fredouille, C., Bozonnet, S., and Evans, N. (2009). The LIA-EURECOM RT'09 speaker diarization system. In: *RT'09, NIST Rich Transcription Workshop*. url: https://www.eurecom.fr/publication/2763.

Fredouille, C. and Evans, N. (2008). New implementations of the E-HMM-based system for speaker diarization in meeting rooms. In: *Proc. ICASSP 2008*. IEEE, pp. 4357–4360. doi: 10.1109/ICASSP.2008.4518620.

Friedland, G. (2012). *Using a GPU, online diarization = offline diarization*. Tech. rep. TR-12-004. International Computer Science Institute, Berkeley, CA, USA. url: https://www.icsi.berkeley.edu/pubs/techreports/TR-12-004.pdf.

Friedland, G. et al. (2012). The ICSI RT-09 speaker diarization system. In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.2, pp. 371–381. doi: 10.1109/TASL.2011.2158419.

Galibert, O. and Kahn, J. (2013). The first official REPERE evaluation. In: *Proc. SLAM 2013*, pp. 43–48. url: https://www.isca-speech.org/archive/slam_2013/slm3_043.html.

Galibert, O., Leixa, J., Adda, G., Choukri, K., and Gravier, G. (2014). The ETAPE speech processing evaluation. In: *Proc. LREC 2014*, pp. 3995–3999. isbn: 978-2-9517408-8-4. url: http://www.lrec-conf.org/proceedings/lrec2014/summaries/1027.html.

Gallardo-Antolín, A., Anguera, X., and Wooters, C. (2006). Multi-stream speaker diarization systems for the meetings domain. In: *Proc. ICSLP 2006*, pp. 2186–2189. url: https://www.isca-speech.org/archive/interspeech_2006/i06_1620.html.

Galliano, S., Gravier, G., and Chaubard, L. (2009). The ESTER 2 evaluation campaign for the rich transcription of French radio broadcasts. In: *Proc. Interspeech 2009*, pp. 2583–2586. url: https://www.isca-speech.org/archive/interspeech_2009/i09_2583.html.

Ganesh, S., Bharat P, Sharma, N., Singh, P., and Ganapathy, S. (2018). *LEAP Submission for DIHARD 2018*. url: https://dihardchallenge.github.io/dihard1/system_descriptions/leap_systems.pdf.

Garcia Perera, L. P. et al. (2020). Speaker detection in the wild: lessons learned from JSALT 2019. In: *Proc. Odyssey 2020*, pp. 415–422. doi: 10.21437/Odyssey.2020-59.

Garcia-Romero, D. and Espy-Wilson, C. Y. (2011). Analysis of i-vector length normalization in speaker recognition systems. In: *Proc. Interspeech 2011*, pp. 249–252. url: https://www.isca-speech.org/archive/interspeech_2011/i11_0249.html.

Garcia-Romero, D., Snyder, D., Sell, G., Povey, D., and McCree, A. (2017). Speaker diarization using deep neural network embeddings. In: *Proc. ICASSP 2017*. IEEE, pp. 4930–4934. doi: 10.1109/ICASSP.2017.7953094.

Garofolo, J. S. et al. (1993). *TIMIT Acoustic-Phonetic Continuous Speech Corpus*. Linguistic Data Consortium. ldc: LDC93S1.

Gauvain, J.-L., Lamel, L. F., and Adda, G. (1998). Partitioning and transcription of broadcast news data. In: *Proc. ICSLP 1998*. paper 0084. url: https://www.isca-speech.org/archive/icslp_1998/i98_0084.html.

Gebre, B. G., Wittenburg, P., Heskes, T., and Drude, S. (2014a). Motion history images for online speaker/signer diarization. In: *Proc. ICASSP 2014*. IEEE, pp. 1537–1541. doi: 10.1109/ICASSP.2014.6853855.

Gebre, B. G., Wittenburg, P., Drude, S., Huijbregts, M., and Heskes, T. (2014b). Speaker diarization using gesture and speech. In: *Proc. Interspeech 2014*, pp. 582–586. url: https://www.isca-speech.org/archive/interspeech_2014/i14_0582.html.

Gebru, I. D., Ba, S., Li, X., and Horaud, R. (2017). Audio-visual speaker diarization based on spatiotemporal Bayesian fusion. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.5, pp. 1086–1099.

Geiger, J. T., Eyben, F., Evans, N., Schuller, B., and Rigoll, G. (2013a). Using linguistic information to detect overlapping speech. In: *Proc. Interspeech 2013*, pp. 690–694. url: https://www.isca-speech.org/archive/interspeech_2013/i13_0690.html.

Geiger, J. T., Eyben, F., Schuller, B., and Rigoll, G. (2013b). Detecting overlapping speech with long short-term memory recurrent neural networks. In: *Proc. Interspeech 2013*, pp. 1668–1672. url: https://www.isca-speech.org/archive/interspeech_2013/i13_1668.html.

Geiger, J. T., Wallhoff, F., and Rigoll, G. (2010). GMM-UBM based open-set online speaker diarization. In: *Proc. Interspeech 2010*, pp. 2330–2333. url: https://www.isca-speech.org/archive/interspeech_2010/i10_2330.html.

Ghahabi, O. and Fischer, V. (2018). EML submission to Albayzin 2018 speaker diarization challenge. In: *Proc. IberSPEECH 2018*, pp. 216–219. doi: 10.21437/IberSPEECH.2018-44.

Ghahabi, O. and Fischer, V. (2019). Speaker-corrupted embeddings for online speaker diarization. In: *Proc. Interspeech 2019*, pp. 386–390. doi: 10.21437/Interspeech.2019-2756.

Giraudel, A. et al. (2012). The REPERE corpus: a multimodal corpus for person recognition. In: *Proc. LREC 2012*, pp. 1102–1107. isbn: 978-2-9517408-7-7. url: http://www.lrec-conf.org/proceedings/lrec2012/summaries/707.html.

Gračič, M., Kos, M., and Kačič, Z. (2010). Online speaker segmentation and clustering using cross-likelihood ratio calculation with reference criterion selection. In: *IET Signal Processing* 4.6, pp. 673–685. doi: 10.1049/iet-spr.2009.0235.

Gupta, V. (2015). Speaker change point detection using deep neural nets. In: *Proc. ICASSP 2015*. IEEE, pp. 4420–4424. doi: 10.1109/ICASSP.2015.7178806.

Gupta, V. and Alam, J. (2018). *CRIM's Speaker Diarization System for the DIHARD Diarization Challenge*. url: https://dihardchallenge.github.io/dihard1/system_descriptions/crim_systems.pdf.

Hagerer, G., Pandit, V., Eyben, F., and Schuller, B. (2017). Enhancing LSTM RNN-based speech overlap detection by artificially mixed data. In: *Proc. AES Int. Conf. on Semantic Audio*. Audio Engineering Society. url: https://www.aes.org/e-lib/browse.cfm?elib=18764.

Hasan, T., Saeidi, R., Hansen, J. H. L., and van Leeuwen, D. A. (2013). Duration mismatch compensation for i-vector based speaker recognition systems. In: *Proc. ICASSP 2013*. IEEE, pp. 7663–7667. doi: 10.1109/ICASSP.2013.6639154.

Heldner, M. and Edlund, J. (2010). Pauses, gaps and overlaps in conversations. In: *Journal of Phonetics* 38.4, pp. 555–568. doi: 10.1016/j.wocn.2010.08.002.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. In: *The Journal of the Acoustical Society of America* 87.4, pp. 1738–1752. doi: 10.1121/1.399423.

Himawan, I., Kanagasundaram, A., Ghaemmaghami, H., Sridharan, S., and Fookes, C. (2018). *The QUT speaker diarization system for the First DIHARD challenge*. url: https://dihardchallenge.github.io/dihard1/system_descriptions/saivt_systems.pdf.

Hoffer, E. and Ailon, N. (2015). Deep metric learning using triplet network. In: *International Workshop on Similarity-Based Pattern Recognition. SIMBAD 2015*, pp. 84–92. doi: 10.1007/978-3-319-24261-3_7.

Horiguchi, S., Fujita, Y., Watanabe, S., Xue, Y., and Nagamatsu, K. (2020). End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors. In: *Proc. Interspeech 2020*, pp. 269–273. doi: 10.21437/Interspeech.2020-1022.

Hrúz, M. and Kunešová, M. (2016). Convolutional neural network in the task of speaker change detection. In: *Proc. SPECOM 2016*. Springer, pp. 191–198. doi: 10.1007/978-3-319-43958-7_22.

Hrúz, M. and Zajíc, Z. (2017). Convolutional neural network for speaker change detection in telephone speaker diarization system. In: *Proc. ICASSP 2017*, pp. 4945–4949. DOI: 10.1109/ICASSP.2017.7953097.

Huang, Z., García-Perera, L. P., Villalba, J., Povey, D., and Dehak, N. (2018). JHU diarization system description. In: *Proc. IberSPEECH 2018*, pp. 236–239. DOI: 10.21437/IberSPEECH.2018-49.

Huang, Z. et al. (2020). Speaker diarization with region proposal network. In: *Proc. ICASSP 2020*, pp. 6514–6518. DOI: 10.1109/ICASSP40776.2020.9053760.

Huijbregts, M., van Leeuwen, D., and Hain, T. (2009). The AMI RT09s Speaker Diarization System. Presented at the 2009 Rich Transcription Evaluation Conference, May 28-29 2009, Melbourne. URL: https://web.archive.org/web/20170119114303/https://www.itl.nist.gov/iad/mig/tests/rt/2009/workshop/ami-diarization.pdf (archived on: 2017-01-19).

Huijbregts, M. and Wooters, C. (2007). The blame game: performance analysis of speaker diarization system components. In: *Proc. Interspeech 2007*, pp. 1857–1860. URL: https://www.isca-speech.org/archive/interspeech_2007/i07_1857.html.

Ioffe, S. (2006). Probabilistic linear discriminant analysis. In: *Proc. Eur. Conf. on Computer Vision*. Springer-Verlag, pp. 531–542. DOI: 10.1007/11744085_41.

Ito, N., Makino, T., Araki, S., and Nakatani, T. (2018). Maximum-likelihood online speaker diarization in noisy meetings based on categorical mixture model and probabilistic spatial dictionary. In: *Proc. ICASSP 2018*, pp. 546–550. DOI: 10.1109/ICASSP.2018.8462104.

Jeub, M., Schafer, M., and Vary, P. (2009). A binaural room impulse response database for the evaluation of dereverberation algorithms. In: *Proc. Int. Conf. on Digital Signal Processing (DSP'09)*. IEEE, pp. 1–5. DOI: 10.1109/ICDSP.2009.5201259.

Kanagasundaram, A., Vogt, R., Dean, D., Sridharan, S., and Mason, M. (2011). i-vector based speaker recognition on short utterances. In: *Proc. Interspeech 2011*, pp. 2341–2344. URL: https://www.isca-speech.org/archive/interspeech_2011/i11_2341.html.

Kanda, N. et al. (2020). Joint speaker counting, speech recognition, and speaker identification for overlapped speech of any number of speakers. In: *Proc. Interspeech 2020*, pp. 36–40. DOI: 10.21437/Interspeech.2020-1085.

Kapsouras, I. et al. (2017). Multimodal speaker clustering in full length movies. In: *Multimedia Tools and Applications* 76.2, pp. 2223–2242. DOI: 10.1007/s11042-015-3181-5.

Kazimirova, E. and Belyaev, A. (2018). Automatic detection of multi-speaker fragments with high time resolution. In: *Proc. Interspeech 2018*, pp. 1388–1392. DOI: 10.21437/Interspeech.2018-1878.

Kenny, P. (2005). *Joint factor analysis of speaker and session variability: Theory and algorithms*. Tech. rep. CRIM-06/08-13. Montreal: CRIM. URL: https://www.crim.ca/perso/patrick.kenny/FAtheory.pdf.

Kenny, P. (2008). *Bayesian Analysis of Speaker Diarization with Eigenvoice Priors*. Tech. rep. Montreal: CRIM. URL: https://www.crim.ca/perso/patrick.kenny/BayesCluster.pdf.

Kenny, P. and Dumouchel, P. (2004). Experiments in speaker verification using factor analysis likelihood ratios. In: *Proc. Odyssey 2004*, pp. 219–226. URL: https://www.isca-speech.org/archive_open/odyssey_04/ody4_219.html.

Kenny, P., Reynolds, D., and Castaldo, F. (2010). Diarization of telephone conversations using factor analysis. In: *IEEE Journal of Selected Topics in Signal Processing* 4.6, pp. 1059–1070. DOI: 10.1109/JSTSP.2010.2081790.

Khemiri, H., Petrovska-Delacrétaz, D., and Chollet, G. (2013). Speaker diarization using data-driven audio sequencing. In: *Proc. ICASSP 2013*. IEEE, pp. 7736–7740. DOI: 10.1109/ICASSP.2013.6639169.

Khosravani, A., Glackin, C., Dugan, N., Chollet, G., and Cannings, N. (2018a). *The Intelligent Voice System Description for the First DIHARD Challenge*. URL: https://dihardchallenge.github.io/dihard1/system_descriptions/intelligentvoice_systems.pdf.

Khosravani, A., Glackin, C., Dugan, N., Chollet, G., and Cannings, N. (2018b). The intelligent voice system for the IberSPEECH-RTVE 2018 speaker diarization challenge. In: *Proc. IberSPEECH 2018*, pp. 231–235. DOI: 10.21437/IberSPEECH.2018-48.

Kim, S., Valente, F., Filippone, M., and Vinciarelli, A. (2014). Predicting continuous conflict perception with Bayesian Gaussian processes. In: *IEEE Transactions on Affective Computing* 5.2, pp. 187–200. DOI: 10.1109/TAFFC.2014.2324564.

Kotti, M., Moschou, V., and Kotropoulos, C. (2008). Speaker segmentation and clustering. In: *Signal Processing* 88.5, pp. 1091–1124. DOI: 10.1016/j.sigpro.2007.11.017.

Kounades-Bastian, D., Girin, L., Alameda-Pineda, X., Gannot, S., and Horaud, R. (2017). An EM algorithm for joint source separation and diarisation of multichannel convolutive speech mixtures. In: *Proc. ICASSP 2017*, pp. 16–20. DOI: 10.1109/ICASSP.2017.7951789.

Kounadis-Bastian, D. (2017). New contributions to audio source separation and diarisation of Multichannel Convolutive Mixtures. PhD thesis. Grenoble: Université Grenoble Alpes. HAL: tel-01681361.

Kunešová, M. (2017). Speaker Diarization. PhD thesis report. Plzeň: Západočeská univerzita v Plzni.

Kunešová, M. (2018). Detection of overlapping speech using a convolutional neural network: first experiments. In: *SVK FAV 2018 – magisterské a doktorské studijní programy*. Západočeská univerzita v Plzni, pp. 56–57. ISBN: 978-80-261-0790-3.

Kunešová, M., Hrúz, M., Zajíc, Z., and Radová, V. (2019). Detection of overlapping speech for the purposes of speaker diarization. In: *Proc. SPECOM 2019*, pp. 247–257. DOI: 10.1007/978-3-030-26061-3_26.

Kunešová, M. and Radová, V. (2015). Ideas for clustering of similar models of a speaker in an online speaker diarization system. In: *Proc. TSD 2015*. Springer, pp. 225–233. DOI: 10.1007/978-3-319-24033-6_26.

Kunešová, M., Zajíc, Z., and Radová, V. (2017). Experiments with segmentation in an online speaker diarization system. In: *Proc. TSD 2017*. Springer, pp. 429–437. DOI: 10.1007/978-3-319-64206-2_48.

Landini, F. et al. (2020). BUT system for the second DIHARD speech diarization challenge. In: *Proc. ICASSP 2020*, pp. 6529–6533. DOI: 10.1109/ICASSP40776.2020.9054251.

Larcher, A. et al. (2012). I-vectors in the context of phonetically-constrained short utterances for speaker verification. In: *Proc. ICASSP 2012*, pp. 4773–4776. doi: 10.1109/ICASSP.2012.6288986.

Le, V. B., Barras, C., and Ferràs, M. (2010). On the use of GSV-SVM for speaker diarization and tracking. In: *Proc. Odyssey 2010*, pp. 146–150. url: https://www.isca-speech.org/archive_open/odyssey_2010/od10_026.html.

Le Lan, G., Charlet, D., Larcher, A., and Meignier, S. (2017). A triplet ranking-based neural network for speaker diarization and linking. In: *Proc. Interspeech 2017*, pp. 3572–3576. doi: 10.21437/Interspeech.2017-270.

Li, Z. and Whitehill, J. (2020). Compositional embedding models for speaker identification and diarization with simultaneous speech from 2+ speakers. In: *arXiv preprint*. arXiv: 2010.11803v1 [cs.SD].

Lin, Q., Yin, R., Li, M., Bredin, H., and Barras, C. (2019). LSTM based similarity measurement with spectral clustering for speaker diarization. In: *Proc. Interspeech 2019*, pp. 366–370. doi: 10.21437/Interspeech.2019-1388.

Lin, Q. et al. (2020). DIHARD II is still hard: experimental results and discussions from the DKU-LENOVO team. In: *Proc. Odyssey 2020*, pp. 102–109. doi: 10.21437/Odyssey.2020-15.

Liu, D. and Kubala, F. (2004). Online speaker clustering. In: *Proc. ICASSP 2004*. Vol. 1, pp. 333–336. doi: 10.1109/ICASSP.2004.1325990.

Lleida, E. et al. (2019). Albayzin 2018 evaluation: the IberSpeech-RTVE challenge on speech technologies for Spanish broadcast media. In: *Applied Sciences* 9.24, p. 5412. doi: 10.3390/app9245412.

Lozano-Diez, A., Labrador, B., de Benito, D., Ramirez, P., and T. Toledano, D. (2018). DNN-based embeddings for speaker diarization in the AuDIaS-UAM system for the Albayzin 2018 iberspeech-rtve evaluation. In: *Proc. IberSPEECH 2018*, pp. 224–226. doi: 10.21437/IberSPEECH.2018-46.

Luque, J. and Hernando, J. (2009). Speaker Diarization for Conference Room: The UPC RT09 Evaluation System. Presented at the 2009 Rich Transcription Evaluation Conference, May 28-29 2009, Melbourne. url: https://web.archive.org/web/20161222084459/http://itl.nist.gov/iad/mig/tests/rt/2009/workshop/UPC_SPKR_RT09.pdf (archived on: 2016-12-22).

Machlica, L. (2012). High Dimensional Spaces and Modelling in the Task of Speaker Recognition. PhD thesis. Plzeň: Západočeská univerzita v Plzni.

Machlica, L. and Zajíc, Z. (2012). Factor analysis and nuisance attribute projection revisited. In: *Proc. Interspeech 2012*, pp. 1570–1573. url: https://www.isca-speech.org/archive/interspeech_2012/i12_1572.html.

Maciejewski, M., Snyder, D., Manohar, V., Dehak, N., and Khudanpur, S. (2018). Characterizing performance of speaker diarization systems on far-field speech using standard methods. In: *Proc. ICASSP 2018*, pp. 5244–5248. doi: 10.1109/ICASSP.2018.8461546.

Madikeri, S., Himawan, I., Motlicek, P., and Ferras, M. (2015). Integrating online i-vector extractor with information bottleneck based speaker diarization system. In: *Proc. Interspeech 2015*, pp. 3105–3109. url: https://www.isca-speech.org/archive/interspeech_2015/i15_3105.html.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Online PDF edition. Cambridge University Press. isbn: 978-0-521-86571-5. url: https://nlp.stanford.edu/IR-book/.

Mansfield, P. A., Wang, Q., Downey, C., Wan, L., and Moreno, I. L. (2018). *Links: A High-Dimensional Online Clustering Method*. arXiv: 1801.10123.

Markov, K. and Nakamura, S. (2007). Never-ending learning system for on-line speaker diarization. In: *IEEE Workshop on Automatic Speech Recognition & Understanding, 2007. ASRU*. IEEE, pp. 699–704. doi: 10.1109/ASRU.2007.4430197.

Markov, K. and Nakamura, S. (2008). Improved novelty detection for online GMM based speaker diarization. In: *Proc. Interspeech 2008*, pp. 363–366. url: https://www.isca-speech.org/archive/interspeech_2008/i08_0363.html.

Matějů, L. (2020). Speech Activity and Speaker Change Point Detection for Online Streams. PhD thesis. Liberec: Technická univerzita v Liberci.

McLaren, M., Castan, D., Graciarena, M., and Ferrer, L. (2018). *The SRI International STAR-LAB DiHard Challenge System Description*. url: https://dihardchallenge.github.io/dihard1/system_descriptions/star-lab_systems.pdf.

McLaren, M., Lei, Y., and Ferrer, L. (2015). Advances in deep neural network approaches to speaker recognition. In: *Proc. ICASSP 2015*. IEEE, pp. 4814–4818. doi: 10.1109/ICASSP.2015.7178885.

Meignier, S. et al. (2013). *LIUM Speaker Diarization Wiki*. Internet Archive. url: https://web.archive.org/web/20131105011107/http://lium3.univ-lemans.fr/diarization/doku.php/overview (archived on: 2013-11-05).

Meignier, S., Moraru, D., Fredouille, C., Bonastre, J.-F., and Besacier, L. (2006). Step-by-step and integrated approaches in broadcast news speaker diarization. In: *Computer Speech & Language* 20.2, pp. 303–330. doi: 10.1016/j.csl.2005.08.002.

Miasato Filho, V. A., Silva, D. A., and Depra Cuozzo, L. G. (2018). Joint discriminative embedding learning, speech activity and overlap detection for the DIHARD speaker diarization challenge. In: *Proc. Interspeech 2018*, pp. 2818–2822. doi: 10.21437/Interspeech.2018-2304.

Minotto, V. P., Jung, C. R., and Lee, B. (2015). Multimodal multi-channel on-line speaker diarization using sensor fusion through SVM. In: *IEEE Transactions on Multimedia* 17.10, pp. 1694–1705. doi: 10.1109/TMM.2015.2463722.

Nakatani, T., Yoshioka, T., Kinoshita, K., Miyoshi, M., and Juang, B.-H. (2010). Speech dereverberation based on variance-normalized delayed linear prediction. In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.7, pp. 1717–1731. doi: 10.1109/TASL.2010.2052251.

National Institute of Standards and Technology (2009). *The 2009 (RT-09) Rich Transcription Meeting Recognition Evaluation Plan*. url: https://web.archive.org/web/20170126064326/http://www.itl.nist.gov/iad/mig/tests/rt/2009/docs/rt09-meeting-eval-plan-v2.pdf (archived on: 2017-01-26).

National Institute of Standards and Technology (2018). *SCTK, the NIST Scoring Toolkit*. Version 2.4.11. url: https://github.com/usnistgov/SCTK.

Neri, L. V., Pinheiro, H. N. B., Tsang, I. R., da C. Cavalcanti, G. D., and Adami, A. G. (2017). Speaker segmentation using i-vector in meetings domain. In: *Proc. ICASSP 2017*, pp. 5455–5459. doi: 10.1109/ICASSP.2017.7953199.

von Neumann, T. et al. (2019). All-neural online source separation, counting, and diarization for meeting analysis. In: *Proc. ICASSP 2019*, pp. 91–95. doi: 10.1109/ICASSP.2019.8682572.

Ng, A., Jordan, M., and Weiss, Y. (2001). On spectral clustering: analysis and an algorithm. In: *Advances in neural information processing systems* 14, pp. 849–856.

Nguyen, T. H. et al. (2009). The IIR-NTU speaker diarization systems for RT 2009. Presented at the 2009 Rich Transcription Evaluation Conference, May 28-29 2009, Melbourne. URL: https://web.archive.org/web/20161222095340/http://www.itl.nist.gov/iad/mig/tests/rt/2009/workshop/IIR-NTU-presentation.pdf (archived on: 2016-12-22).

Ning, H., Liu, M., Tang, H., and Huang, T. S. (2006). A spectral clustering approach to speaker diarization. In: *Proc. ICSLP 2006*, pp. 2178–2181. URL: https://www.isca-speech.org/archive/interspeech_2006/i06_1607.html.

Noulas, A., Englebienne, G., and Kröse, B. J. A. (2012). Multimodal speaker diarization. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.1, pp. 79–93. DOI: 10.1109/TPAMI.2011.47.

Noulas, A. and Krose, B. J. A. (2007). On-line multi-modal speaker diarization. In: *Proc. Int. Conf. on Multimodal Interfaces*. ACM, pp. 350–357. DOI: 10.1145/1322192.1322254.

Novoselov, S. et al. (2019). Speaker diarization with deep speaker embeddings for DIHARD Challenge II. In: *Proc. Interspeech 2019*, pp. 1003–1007. DOI: 10.21437/Interspeech.2019-2757.

Oku, T., Sato, S., Kobayashi, A., Homma, S., and Imai, T. (2012). Low-latency speaker diarization based on Bayesian information criterion with multiple phoneme classes. In: *Proc. ICASSP 2012*. IEEE, pp. 4189–4192. DOI: 10.1109/ICASSP.2012.6288842.

Otterson, S. and Ostendorf, M. (2007). Efficient use of overlap information in speaker diarization. In: *Proc. ASRU 2007*. IEEE, pp. 683–686. DOI: 10.1109/ASRU.2007.4430194.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). LibriSpeech: an ASR corpus based on public domain audio books. In: *Proc. ICASSP 2015*, pp. 5206–5210. DOI: 10.1109/ICASSP.2015.7178964.

Pardo, J. M., Anguera, X., and Wooters, C. (2006). Speaker diarization for multiple distant microphone meetings: mixing acoustic features and inter-channel time differences. In: *Proc. Interspeech 2006*, pp. 2194–2197. URL: https://www.isca-speech.org/archive/interspeech_2006/i06_1337.html.

Pardo, J. M., Barra, R., and Martínez, B. (2009). The UPM RT09 Meetings Evaluation System. Presented at the 2009 Rich Transcription Evaluation Conference, May 28-29 2009, Melbourne. URL: https://web.archive.org/web/20170128084728/http://www.itl.nist.gov/iad/mig/tests/rt/2009/workshop/upmrt09-presentation-2009-5-29-2.pdf (archived on: 2017-01-28).

Park, T. J. et al. (2019a). Speaker diarization with lexical information. In: *Proc. Interspeech 2019*, pp. 391–395. DOI: 10.21437/Interspeech.2019-1947.

Park, T. J. et al. (2019b). The Second DIHARD Challenge: System Description for USC-SAIL Team. In: *Proc. Interspeech 2019*, pp. 998–1002. DOI: 10.21437/Interspeech.2019-1903.

Park, T. J. et al. (2021). *A Review of Speaker Diarization: Recent Advances with Deep Learning*. arXiv: 2101.09624 [eess.AS].

Patino, J., Delgado, H., and Evans, N. (2017). Speaker change detection using binary key modelling with contextual information. In: *Proc. SLSP 2017*, pp. 250–261. doi: 10.1007/978-3-319-68456-7_21.

Patino, J., Delgado, H., and Evans, N. (2018a). The EURECOM submission to the first DIHARD challenge. In: *Proc. Interspeech 2018*, pp. 2813–2817. doi: 10.21437/Interspeech.2018-2172.

Patino, J. et al. (2018b). Low-latency speaker spotting with online diarization and detection. In: *Proc. Odyssey 2018*, pp. 140–146. doi: 10.21437/Odyssey.2018-20.

Patino, J. et al. (2018c). ODESSA at Albayzin speaker diarization challenge 2018. In: *Proc. IberSPEECH 2018*, pp. 211–215. doi: 10.21437/IberSPEECH.2018-43.

Peddinti, V., Povey, D., and Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In: *Proc. Interspeech 2015*, pp. 3214–3218. url: https://www.isca-speech.org/archive/interspeech_2015/i15_3214.html.

Pelleg, D. and Moore, A. W. (2000). X-means: extending k-means with efficient estimation of the number of clusters. In: *Proc. Int. Conf. on Machine Learning*, pp. 727–734.

Pfau, T., Ellis, D. P. W., and Stolcke, A. (2001). Multispeaker speech activity detection for the ICSI meeting recorder. In: *IEEE Workshop on Automatic Speech Recognition and Understanding, 2001. ASRU'01.* IEEE, pp. 107–110. doi: 10.1109/ASRU.2001.1034599.

Plchot, O., Burget, L., Aronowitz, H., and Matějka, P. (2016). Audio enhancing with DNN autoencoder for speaker recognition. In: *Proc. ICASSP 2016*, pp. 5090–5094. doi: 10.1109/ICASSP.2016.7472647.

Prasad, R. V. et al. (2002). Comparison of voice activity detection algorithms for VoIP. In: *Proc. Int. Symp. on Computers and Communications*. IEEE, pp. 530–535. doi: 10.1109/ISCC.2002.1021726.

Prince, S. J. D. and Elder, J. H. (2007). Probabilistic linear discriminant analysis for inferences about identity. In: *Proc. Int. Conf. on Computer Vision*. IEEE, pp. 1–8. doi: 10.1109/ICCV.2007.4409052.

Psutka, J., Müller, L., Matoušek, J., and Radová, V. (2006). *Mluvíme s počítačem česky*. [Talking to the Computer in Czech]. [In Czech]. Prague: Academia. 752 pp. isbn: 80-200-1309-1.

Raj, D., Huang, Z., and Khudanpur, S. (2020a). Multi-class spectral clustering with overlaps for speaker diarization. In: *arXiv preprint*. To appear at IEEE SLT 2021. arXiv: 2011.02900 [eess.AS].

Raj, D. et al. (2020b). DOVER-Lap: a method for combining overlap-aware diarization outputs. In: *arXiv preprint*. To appear at IEEE SLT 2021. arXiv: 2011.01997 [eess.AS].

Ramirez, J., Górriz, J. M., and Segura, J. C. (2007). Voice activity detection. fundamentals and speech recognition system robustness. In: *Robust Speech Recognition and Understanding*. Ed. by M. Grimm and K. Kroschel. InTech Open Access Publisher. isbn: 978-3-902613-08-0.

Ramos-Muguerza, E., Docío-Fernández, L., and Alba-Castro, J. L. (2018). The GTM-UVIGO system for audiovisual diarization. In: *Proc. IberSPEECH 2018*, pp. 204–207. doi: 10.21437/IberSPEECH.2018-41.

Ravanelli, M. and Bengio, Y. (2018). Speaker recognition from raw waveform with sincnet. In: *Proc. IEEE Spoken Language Technology Workshop (SLT) 2018*, pp. 1021–1028. doi: 10.1109/SLT.2018.8639585.

Reynolds, D. A. et al. (1998). Blind clustering of speech utterances based on speaker and language characteristics. In: *Proc. ICSLP 1998*. Paper 0610. url: https://www.isca-speech.org/archive/icslp_1998/i98_0610.html.

Reynolds, D. A. (1995). Speaker identification and verification using Gaussian mixture speaker models. In: *Speech communication* 17.1, pp. 91–108. doi: 10.1016/0167-6393(95)00009-D.

Rouvier, M. and Meignier, S. (2012). A global optimization framework for speaker diarization. In: *Proc. Odyssey 2012*, pp. 146–150. url: https://www.isca-speech.org/archive/odyssey_2012/od12_146.html.

Rouvier, M. et al. (2013). An open-source state-of-the-art toolbox for broadcast news diarization. In: *Proc. Interspeech 2013*, pp. 1477–1481. url: https://www.isca-speech.org/archive/interspeech_2013/i13_1477.html.

Ryant, N. et al. (2018a). *DIHARD Corpus*. Linguistic Data Consortium.

Ryant, N. et al. (2018b). *First DIHARD Challenge Evaluation Plan. (version 1.3)*. doi: 10.5281/zenodo.1199638.

Ryant, N. et al. (2019). *Second DIHARD Challenge Evaluation Plan. (version 1.2)*. url: https://dihardchallenge.github.io/dihard2/docs/second_dihard_eval_plan_v1.2.pdf.

Sahidullah, M. et al. (2019). *The Speed Submission to DIHARD II: Contributions & Lessons Learned*. arXiv: 1911.02388 [eess.AS].

Sajjan, N., Ganesh, S., Sharma, N., Ganapathy, S., and Ryant, N. (2018). Leveraging LSTM models for overlap detection in multi-party meetings. In: *Proc. ICASSP 2018*. IEEE, pp. 5249–5253. doi: 10.1109/ICASSP.2018.8462548.

Salmun, I., Opher, I., and Lapidot, I. (2016). On the use of PLDA i-vector scoring for clustering short segments. In: *Proc. Odyssey 2016*, pp. 407–414. doi: 10.21437/Odyssey.2016-59.

Sato, M.-A. and Ishii, S. (2000). On-line EM algorithm for the normalized Gaussian network. In: *Neural computation* 12.2, pp. 407–432. doi: 10.1162/089976600300015853.

Schmalenstroeer, J., Kelling, M., Leutnant, V., and Haeb-Umbach, R. (2009). Fusing audio and video information for online speaker diarization. In: *Proc. Interspeech 2009*, pp. 1163–1166. url: https://www.isca-speech.org/archive/interspeech_2009/i09_1163.html.

Sell, G. and Garcia-Romero, D. (2014). Speaker diarization with PLDA i-vector scoring and unsupervised calibration. In: *Proc. IEEE Spoken Language Technology Workshop*. IEEE, pp. 413–417. doi: 10.1109/SLT.2014.7078610.

Sell, G. and Garcia-Romero, D. (2015). Diarization resegmentation in the factor analysis subspace. In: *Proc. ICASSP 2015*. IEEE, pp. 4794–4798. doi: 10.1109/ICASSP.2015.7178881.

Sell, G., Garcia-Romero, D., and McCree, A. (2015). Speaker diarization with i-vectors from DNN senone posteriors. In: *Proc. Interspeech 2015*, pp. 3096–3099. url: https://www.isca-speech.org/archive/interspeech_2015/i15_3096.html.

Sell, G. et al. (2018). Diarization is hard: some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge. In: *Proc. Interspeech 2018*, pp. 2808–2812. doi: 10.21437/Interspeech.2018-1893.

Senoussaoui, M., Kenny, P., Stafylakis, T., and Dumouchel, P. (2014). A study of the cosine distance-based mean shift for telephone speech diarization. In: *IEEE/ACM Transactions o Audio, Speech, and Language Processing* 22.1, pp. 217–227. doi: 10.1109/TASLP.2013.2285474.

Shokouhi, N., Ziaei, A., Sangwan, A., and Hansen, J. H. L. (2015). Robust overlapped speech detection and its application in word-count estimation for Prof-Life-Log data. In: *Proc. ICASSP 2015*. IEEE, pp. 4724–4728. doi: 10.1109/ICASSP.2015.7178867.

Shum, S., Dehak, N., Chuangsuwanich, E., Reynolds, D., and Glass, J. (2011). Exploiting intra-conversation variability for speaker diarization. In: *Proc. Interspeech 2011*. Vol. 11, pp. 945–948. url: https://www.isca-speech.org/archive/interspeech_2011/i11_0945.html.

Shum, S., Dehak, N., and Glass, J. (2012). On the use of spectral and iterative methods for speaker diarization. In: *Proc. Interspeech 2012*, pp. 482–485.

Shum, S. H., Dehak, N., Dehak, R., and Glass, J. R. (2013). Unsupervised methods for speaker diarization: an integrated and iterative approach. In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.10, pp. 2015–2028. doi: 10.1109/TASL.2013.2264673.

Siegler, M. A., Jain, U., Raj, B., and Stern, R. M. (1997). Automatic segmentation, classification and clustering of broadcast news audio. In: *Proc. DARPA Speech Recognition Workshop*, pp. 97–99.

Silovský, J. (2011). Generativní a diskriminativní klasifikátory v úlohách textově nezávislého rozpoznávání a diarizace mluvčích. [Generative and discriminative classifiers in the tasks of text-independent speaker recognition and diarization]. [In Czech]. PhD thesis. Liberec: Technická univerzita v Liberci.

Singh, P. and Ganapathy, S. (2020). Deep self-supervised hierarchical clustering for speaker diarization. In: *Proc. Interspeech 2020*, pp. 294–298. doi: 10.21437/Interspeech.2020-2297.

Singh, P., M.A., H. V., Ganapathy, S., and Kanagasundaram, A. (2019). LEAP diarization system for the Second DIHARD Challenge. In: *Proc. Interspeech 2019*, pp. 983–987. doi: 10.21437/Interspeech.2019-2716.

Snyder, D., Garcia-Romero, D., Povey, D., and Khudanpur, S. (2017). Deep neural network embeddings for text-independent speaker verification. In: *Proc. Interspeech 2017*, pp. 999–1003. doi: 10.21437/Interspeech.2017-620.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. (2018). X-vectors: robust DNN embeddings for speaker recognition. In: *Proc. ICASSP 2018*, pp. 5329–5333. doi: 10.1109/ICASSP.2018.8461375.

Soldi, G., Beaugeant, C., and Evans, N. (2015). Adaptive and online speaker diarization for meeting data. In: *Proc. EUSIPCO 2015*. IEEE, pp. 2112–2116. doi: 10.1109/EUSIPCO.2015.7362757.

Soldi, G., Bozonnet, S., Alegre, F., Beaugeant, C., and Evans, N. (2014). Short-duration speaker modelling with phone adaptive training. In: *Proc. Odyssey 2014*, pp. 208–215. url: https://www.isca-speech.org/archive/odyssey_2014/abstracts.html#abs36.

Soldi, G., Todisco, M., Delgado, H., Beaugeant, C., and Evans, N. (2016). Semi-supervised on-line speaker diarization for meeting data with incremental maximum a-posteriori adaptation. In: *Proc. Odyssey 2016*, pp. 377–384. doi: 10.21437/Odyssey.2016-55.

Song, H., Willi, M., Thiagarajan, J. J., Berisha, V., and Spanias, A. (2018). Triplet network with attention for speaker diarization. In: *Proc. Interspeech 2018*, pp. 3608–3612. doi: 10.21437/Interspeech.2018-2305.

Stafylakis, T. and Katsouros, V. (2011). A review of recent advances in speaker diarization with Bayesian methods. In: *Speech and Language Technologies*. Ed. by I. Ipsic. InTech. doi: 10.5772/20521.

Sun, L. et al. (2018a). A novel LSTM-based speech preprocessor for speaker diarization in realistic mismatch conditions. In: *Proc. ICASSP 2018*. IEEE, pp. 5234–5238. doi: 10.1109/ICASSP.2018.8462311.

Sun, L. et al. (2018b). Speaker diarization with enhancing speech for the first DIHARD challenge. In: *Proc. Interspeech 2018*, pp. 2793–2797. doi: 10.21437/Interspeech.2018-1742.

Sundar, H., Sreenivas, T. V., and Kellermann, W. (2013). Identification of active sources in single-channel convolutive mixtures using known source models. In: *IEEE Signal Processing Letters* 20.2, pp. 153–156. doi: 10.1109/LSP.2012.2236314.

Thiemann, J., Ito, N., and Vincent, E. (2013a). *DEMAND: a collection of multi-channel recordings of acoustic noise in diverse environments*. Version 1.0. Zenodo. doi: 10.5281/zenodo.1227121.

Thiemann, J., Ito, N., and Vincent, E. (2013b). The Diverse Environments Multi-channel Acoustic Noise Database (DEMAND): a database of multichannel environmental noise recordings. In: *Proc. Mtgs. Acoust. ICA2013*. Vol. 19. 1. Acoustical Society of America, p. 035081. doi: 10.1121/1.4799597.

Thomas, S., Ganapathy, S., Saon, G., and Soltau, H. (2014). Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions. In: *Proc. ICASSP 2014*. IEEE, pp. 2519–2523. doi: 10.1109/ICASSP.2014.6854054.

Tranter, S. E., Gales, M. J. F., Sinha, R., Umesh, S., and Woodland, P. C. (2004). The development of the Cambridge University RT-04 diarisation system. In: *Proc. Fall 2004 Rich Transcription Workshop (RT-04)*.

Tranter, S. E. and Reynolds, D. A. (2006). An overview of automatic speaker diarization systems. In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.5, pp. 1557–1565. doi: 10.1109/TASL.2006.878256.

Tsai, W.-H. and Lee, H.-C. (2010). Identification of simultaneous speakers. In: *2nd International Conference on Computer Engineering and Technology (ICCET), 2010*. Vol. 3. IEEE, pp. 582–586. doi: 10.1109/ICCET.2010.5485796.

Valente, F. and Wellekens, C. (2006). Variational Bayesian methods for audio indexing. In: *Machine Learning for Multimodal Interaction: Second International Workshop, MLMI 2005*. Ed. by S. Renals and S. Bengio. Springer, pp. 307–319. doi: 10.1007/11677482_27.

Vaquero, C., Vinyals, O., and Friedland, G. (2010). A hybrid approach to online speaker diarization. In: *Proc. Interspeech 2010*, pp. 2638–2641. url: https://www.isca-speech.org/archive/interspeech_2010/i10_2638.html.

Vijayasenan, D., Valente, F., and Bourlard, H. (2009). An information theoretic approach to speaker diarization of meeting data. In: *IEEE Transactions on Audio, Speech, and Language Processing* 17.7, pp. 1382–1393. doi: 10.1109/TASL.2009.2015698.

Viñals, I., Gimeno, P., Ortega, A., Miguel, A., and Lleida, E. (2018a). Estimation of the number of speakers with variational Bayesian PLDA in the DIHARD diarization challenge. In: *Proc. Interspeech 2018*, pp. 2803–2807. doi: 10.21437/Interspeech.2018-1841.

Viñals, I., Gimeno, P., Ortega, A., Miguel, A., and Lleida, E. (2018b). In-domain adaptation solutions for the RTVE 2018 diarization challenge. In: *Proc. IberSPEECH 2018*, pp. 220–223. doi: 10.21437/IberSPEECH.2018-45.

Viñals, I., Gimeno, P., Ortega, A., Miguel, A., and Lleida, E. (2019). ViVoLAB speaker diarization system for the DIHARD 2019 challenge. In: *Proc. Interspeech 2019*, pp. 988–992. doi: 10.21437/Interspeech.2019-2462.

Vinciarelli, A. et al. (2013). *SSPNet Conflict Corpus*. url: https://web.archive.org/web/20180313145831/http://www.dcs.gla.ac.uk/vincia/?p=270 (archived on: 2018-03-13).

Vinyals, O. and Friedland, G. (2008). Towards semantic analysis of conversations: a system for the live identification of speakers in meetings. In: *Proc. Int. Conf. on Semantic Computing 2008*. IEEE, pp. 426–431. doi: 10.1109/ICSC.2008.58.

Walsh, J. M., Kim, Y. E., and Doll, T. M. (2007). Joint iterative multi-speaker identification and source separation using expectation propagation. In: *2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, pp. 283–286. doi: 10.1109/ASPAA.2007.4393034.

Wan, L., Wang, Q., Papir, A., and Lopez Moreno, I. (2018). Generalized end-to-end loss for speaker verification. In: *Proc. ICASSP 2018*, pp. 4879–4883. doi: 10.1109/ICASSP.2018.8462665.

Wang, Q., Downey, C., Wan, L., Mansfield, P. A., and Lopez Moreno, I. (2018). Speaker diarization with LSTM. In: *Proc. ICASSP 2018*, pp. 5239–5243. doi: 10.1109/ICASSP.2018.8462628.

Xu, C., Rao, W., Chng, E. S., and Li, H. (2018). A shifted delta coefficient objective for monaural speech separation using multi-task learning. In: *Proc. Interspeech 2018*, pp. 3479–3483. doi: 10.21437/Interspeech.2018-1150.

Yella, S. H. and Bourlard, H. (2014). Overlapping speech detection using long-term conversational features for speaker diarization in meeting room conversations. In: *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 22.12, pp. 1688–1700. doi: 10.1109/TASLP.2014.2346315.

Yella, S. H. and Stolcke, A. (2015). A comparison of neural network feature transforms for speaker diarization. In: *Proc. Interspeech 2015*, pp. 3026–3030. url: https://www.isca-speech.org/archive/interspeech_2015/i15_3026.html.

Yella, S. H. and Valente, F. (2011). Information bottleneck features for HMM/GMM speaker diarization of meetings recordings. In: *Proc. Interspeech 2011*, pp. 953–956. url: https://www.isca-speech.org/archive//interspeech_2011/i11_0953.html.

Yin, R., Bredin, H., and Barras, C. (2017). Speaker change detection in broadcast TV using bidirectional long short-term memory networks. In: *Proc. Interspeech 2017*, pp. 3827–3831. doi: 10.21437/Interspeech.2017-65.

Yin, R., Bredin, H., and Barras, C. (2018). Neural speech turn segmentation and affinity propagation for speaker diarization. In: *Proc. Interspeech 2018*, pp. 1393–1397. doi: 10.21437/Interspeech.2018-1750.

Zajíc, Z., Hrúz, M., and Müller, L. (2017). Speaker diarization using convolutional neural network for statistics accumulation refinement. In: *Proc. Interspeech 2017*, pp. 3562–3566. doi: 10.21437/Interspeech.2017-51.

Zajíc, Z., Kunešová, M., Hrúz, M., and Vaněk, J. (2019). UWB-NTIS speaker diarization system for the DIHARD II 2019 challenge. In: *Proc. Interspeech 2019*, pp. 993–997. doi: 10.21437/Interspeech.2019-1385.

Zajíc, Z., Kunešová, M., and Radová, V. (2016). Investigation of segmentation in i-vector based speaker diarization of telephone speech. In: *Proc. SPECOM 2016*. Springer, pp. 411–418. doi: 10.1007/978-3-319-43958-7_49.

Zajíc, Z., Kunešová, M., Zelinka, J., and Hrúz, M. (2018). ZCU-NTIS speaker diarization system for the DIHARD 2018 challenge. In: *Proc. Interspeech 2018*, pp. 2788–2792. doi: 10.21437/Interspeech.2018-1252.

Zazo, R., Sainath, T. N., Simko, G., and Parada, C. (2016). Feature learning with raw-waveform CLDNNs for voice activity detection. In: *Proc. Interspeech 2016*, pp. 3668–3672. doi: 10.21437/Interspeech.2016-268.

Zelenák, M., Segura, C., Luque, J., and Hernando, J. (2012). Simultaneous speech detection with spatial features for speaker diarization. In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.2, pp. 436–446. doi: 10.1109/TASL.2011.2160167.

Zelinka, J. (2018). Deep learning and online speech activity detection for Czech radio broadcasting. In: *Proc. TSD 2018*. Springer International Publishing, pp. 428–435. doi: 10.1007/978-3-030-00794-2_46.

Zewoudie, A. W., Luque, J., and Hernando, J. (2018). The use of long-term features for GMM- and i-vector-based speaker diarization systems. In: *EURASIP Journal on Audio, Speech, and Music Processing* 2018.1, p. 14. doi: 10.1186/s13636-018-0140-x.

Zhang, A., Wang, Q., Zhu, Z., Paisley, J., and Wang, C. (2019). Fully supervised speaker diarization. In: *Proc. ICASSP 2019*, pp. 6301–6305. doi: 10.1109/ICASSP.2019.8683892.

Zheng, R., Zhang, C., Zhang, S., and Xu, B. (2014). Variational Bayes based i-vector for speaker diarization of telephone conversations. In: *Proc. ICASSP 2014*. IEEE, pp. 91–95. doi: 10.1109/ICASSP.2014.6853564.

Zhou, X., Garcia-Romero, D., Duraiswami, R., Espy-Wilson, C., and Shamma, S. (2011). Linear versus mel frequency cepstral coefficients for speaker recognition. In: *2011 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, pp. 559–564. doi: 10.1109/ASRU.2011.6163888.

Zhu, W. and Pelecanos, J. (2016). Online speaker diarization using adapted i-vector transforms. In: *Proc. ICASSP 2016*. IEEE, pp. 5045–5049. doi: 10.1109/ICASSP.2016.7472638.

Zochová, P. and Radová, V. (2005). Modified DISTBIC algorithm for speaker change detection. In: *Proc. Interspeech 2005*, pp. 3073–3076. url: https://www.isca-speech.org/archive/interspeech_2005/i05_3073.html.

Dataset References

AIR

Aachen Impulse Response Database. Available from: https://www.iks.rwth-aachen.de/en/research/tools-downloads/databases/aachen-impulse-response-database/. See also: Jeub et al. (2009). Section: 9.5

AMI

AMI Meeting Corpus. Available from: https://groups.inf.ed.ac.uk/ami/corpus/. See also: Carletta et al. (2006). Sections: 7.2, 9.1, 9.2, 9.5

CALLHOME

CALLHOME Corpus. Includes CALLHOME American English Speech (LDC Catalog No.: LDC97S42) and others. Sections: 7.2, 9.1, 9.3

Czech Parliament Meetings

Recorded television broadcasts of June 2012 sessions of Czech Parliament. Not publicly available. Sections: 9.1, 9.2

DEMAND

Diverse Environments Multi-channel Acoustic Noise Database (DEMAND), DOI: 10.5281/zenodo.1227121. See also: Thiemann et al. (2013b). Section: 9.5

DIHARD

First DIHARD Challenge development and evaluation data, Ryant et al. (2018a) and Bergelson (2016). LDC Catalog No.: LDC2019S09, LDC2019S10, LDC2019S12 and LDC2019S13. See also: Ryant et al. (2018b). Sections: 7.2, 9.1, 9.4

ESTER 2

ESTER 2 Corpus. ELRA ID: ELRA-S0338. See also: Galliano et al. (2009). Section: 7.2

ETAPE

ETAPE Evaluation Package, ELRA ID: ELRA-E0046. See also: Galibert et al. (2014). Section: 7.2

LibriSpeech

LibriSpeech ASR corpus. Available from: https://www.openslr.org/12/. See also: Panayotov et al. (2015). Sections: 9.2, 9.5

NIST RT

NIST Rich Transcription Evaluations (includes RT-04S, RT-06, RT-07, RT-09 and others). Official website: https://www.nist.gov/itl/iad/mig/rich-transcription-evaluation. See also: NIST (2009). Section: 7.2

REPERE

REPERE Evaluation Package, ELRA ID: ELRA-E0044. See also: Giraudel et al. (2012) and Galibert and Kahn (2013). Section: 7.2

RTVE2018

RTVE2018 Database, originally used in the IberSpeech-RTVE 2018 Challenge. http://catedrartve.unizar.es/rtvedatabase.html. Section: 7.2

SEEDLingS

Bergelson Seedlings HomeBank Corpus, DOI: 10.21415/T5PK6D, Bergelson (2016). Used as part of the DIHARD Corpus. See: DIHARD

SSPNet Conflict Corpus

SSPNet Conflict Corpus. Previously available from: https://web.archive.org/web/20180313145831/http://www.dcs.gla.ac.uk/vincia/?p=270 (archived webpage), Vinciarelli et al. (2013). See also: Kim et al. (2014) Section: 9.5

TIMIT

TIMIT Acoustic-Phonetic Continuous Speech Corpus, LDC Catalog No.: LDC93S1, Garofolo et al. (1993). Section: 9.5

Software References

BeamformIt

BeamformIt acoustic beamforming software. Available from: https://github.com/xanguera/BeamformIt, see also: Anguera et al. (2007). Sections: 3.2, 3.3, 7.2

Kaldi

Speech recognition toolkit. Available from: https://kaldi-asr.org/. Section: 9.4

md-eval.pl

NIST `md-eval.pl` diarization tool. Available as part of the NIST Scoring Toolkit (SCTK): https://github.com/usnistgov/SCTK. See also: NIST (2009). Section: 7.1

pyannote.metrics

pyannote.metrics: A toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems. Available from: https://github.com/pyannote/pyannote-metrics. See also: Bredin (2017a). Section: 7.1

WPE dereverberation package

A Matlab tool for WPE speech dereverberation. Available from: http://www.kecl.ntt.co.jp/icl/signal/wpe/index.html. See also: Nakatani et al. (2010). Section: 9.5

Authored and Co-authored Publications

Zajíc, Z., Kunešová, M., and Müller, L. (2021). *Applying EEND Diarization to Telephone Recordings from a Call Center*. Submitted to SPECOM 2021.

Kunešová, M., Hrúz, M., Zajíc, Z., and Radová, V. (2019). Detection of overlapping speech for the purposes of speaker diarization. In: *Proc. SPECOM 2019*, pp. 247–257. DOI: 10.1007/978-3-030-26061-3_26.

Zajíc, Z., Kunešová, M., Hrúz, M., and Vaněk, J. (2019). UWB-NTIS speaker diarization system for the DIHARD II 2019 challenge. In: *Proc. Interspeech 2019*, pp. 993–997. DOI: 10.21437/Interspeech.2019-1385.

Kunešová, M. (2018). Detection of overlapping speech using a convolutional neural network: first experiments. In: *SVK FAV 2018 – magisterské a doktorské studijní programy*. Západočeská univerzita v Plzni, pp. 56–57. ISBN: 978-80-261-0790-3.

Zajíc, Z., Kunešová, M., Zelinka, J., and Hrúz, M. (2018). ZCU-NTIS speaker diarization system for the DIHARD 2018 challenge. In: *Proc. Interspeech 2018*, pp. 2788–2792. DOI: 10.21437/Interspeech.2018-1252.

Kunešová, M., Zajíc, Z., and Radová, V. (2017). Experiments with segmentation in an online speaker diarization system. In: *Proc. TSD 2017*. Springer, pp. 429–437. DOI: 10.1007/978-3-319-64206-2_48.

Hrúz, M. and Kunešová, M. (2016). Convolutional neural network in the task of speaker change detection. In: *Proc. SPECOM 2016*. Springer, pp. 191–198. DOI: 10.1007/978-3-319-43958-7_22.

Zajíc, Z., Kunešová, M., and Radová, V. (2016). Investigation of segmentation in i-vector based speaker diarization of telephone speech. In: *Proc. SPECOM 2016*. Springer, pp. 411–418. DOI: 10.1007/978-3-319-43958-7_49.

Kunešová, M. and Radová, V. (2015). Ideas for clustering of similar models of a speaker in an online speaker diarization system. In: *Proc. TSD 2015*. Springer, pp. 225–233. DOI: 10.1007/978-3-319-24033-6_26.

Campr, P., Kunešová, M., Vaněk, J., Čech, J., and Psutka, J. (2014). Audio-video speaker diarization for unsupervised speaker and face model creation. In: *Proc. TSD 2014*. Springer, pp. 465–472. DOI: 10.1007/978-3-319-10816-2_56.

Kunešová, M. (2014). Online speaker diarization. In: *SVK 2014 – magisterské a doktorské studijní programy, sborník rozšířených abstraktů*. Západočeská univerzita v Plzni, pp. 75–76. ISBN: 978-80-261-0365-3.