Using Deep Neural Networks for Speaker Diarisation

Rosanna Margaret Milner

Department of Computer Science
University of Sheffield

Submitted in partial fulfilment of the requirements
for the degree of
Doctor of Philosophy

Speech and Hearing Research Group

March 2017

"*In the beginning there was nothing, which exploded.*"
~Terry Pratchett

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Rosanna Margaret Milner
March 2017

Acknowledgements

First and foremost, my appreciation and thanks goes to my supervisor Professor Thomas Hain. Without his guidance and confidence in me I would not have completed the course. Weekly meetings kept me on track and discussions on how to fix the world has kept me politically aware.

The MINI and overarching SPandH group are wonderful people to work with. I thank Oscar Saz and the other RAs for their support and patience. Discussions on life and food with Yulan Liu, board games with Mortaza Doulaty and Fariba Yousefi, making dumplings with Chenhao Wu and lunchtime chats with the rest of the group have kept me sane.

The PROGRESS postgraduate society was my refuge through my Masters and the initial PhD years. Many thanks to Joanna and Joseph Longworth-Kremer and Samuel Touchard, without their encouragement I never would have started the PhD in the first place. I greatly appreciate the many rants and excuses to go to the pub with Isabelle Augenstein and Chris Quickfall over the years.

I wish to thank my parents and family for their love and encouragement throughout my life. Since primary school, my Mum inspired me to pursue my interest in maths and my Dad encouraged my love of train journeys, leading me to York and then Sheffield. My appreciation goes to David Milner for letting me consistently score higher than him while playing Civilization V. I am indebted to my partner, Mischa van Kesteren, for his constant love, knuffels and healthy winter-warming cumin-free soups.

The Cornish Gang were always there as a reminder of home. Our mini adventures around the country allowed an escape from the norm. From heated political debates to games of Pointless, they've been a constant source of kindness and laughter. Luke Shrimpton has been a great help in pursuing the joint passion of ranting in pubs about PhD life.

I thank Sheena Clarke and my recently made friends at Hope Community Allotment who have allowed me sanctuary through gardening in my final year. I've loved planting seeds, weeding paths, shovelling manure and even the countless attempts to tidy the potting shed. From discussions on communism with Carrie to Europe with Emma, Michael R made sure we got the work done.

I would also like to thank Andreas Stolcke and Geoff Zweig for giving me the opportunity to experience working in industry through an internship at Microsoft Research. The Speech and Dialog Group, as well as fellow interns and other researchers at MSR Silicon Valley, welcomed me with open arms. I will never forget hiking at 30 something degrees Celsius (I never learnt Fahrenheit) in Wallace Falls or watching the Giants lose to the Mets.

Finally, I wish to thank Jana Eggink, the BBC, and the Department of Computer Science for the funding during my time at Sheffield.

Abstract

Speaker diarisation answers the question "who spoke when?" in an audio recording. The input may vary, but a system is required to output speaker labelled segments in time. Typical stages are Speech Activity Detection (SAD), speaker segmentation and speaker clustering. Early research focussed on Conversational Telephone Speech (CTS) and Broadcast News (BN) domains before the direction shifted to meetings and, more recently, broadcast media. The British Broadcasting Corporation (BBC) supplied data through the Multi-Genre Broadcast (MGB) Challenge in 2015 which showed the difficulties speaker diarisation systems have on broadcast media data.

Diarisation is typically an unsupervised task which does not use auxiliary data or information to enhance a system. However, methods which do involve supplementary data have shown promise. Five semi-supervised methods are investigated which use a combination of inputs: different channel types and transcripts. The methods involve Deep Neural Networks (DNNs) for SAD, DNNs trained for channel detection, transcript alignment, and combinations of these approaches. However, the methods are only applicable when datasets contain the required inputs. Therefore, a method involving a pretrained Speaker Separation Deep Neural Network (ssDNN) is investigated which is applicable to every dataset. This technique performs speaker clustering and speaker segmentation using DNNs successfully for meeting data and with mixed results for broadcast media.

The task of diarisation focuses on two aspects: accurate segments and speaker labels. The Diarisation Error Rate (DER) does not evaluate the segmentation quality as it does not measure the number of correctly detected segments. Other metrics exist, such as boundary and purity measures, but these also mask the segmentation quality. An alternative metric is presented based on the F-measure which considers the number of hypothesis segments correctly matched to reference segments. A deeper insight into the segment quality is shown through this metric.

Table of contents

List of figures

List of tables

Acronyms

AANN	Auto-Associative Neural Network.
ACP	Average Cluster Purity.
AHC	Agglomerative Hierarchical Clustering.
aIB	Agglomerative Information Bottleneck.
AMI	Augmented Multiparty Interaction.
ANN	Artificial Neural Network.
ASP	Average Speaker Purity.
ASR	Automatic Speech Recognition.
BBC	British Broadcasting Corporation.
BIC	Bayesian Information Criterion.
BN	Broadcast News.
BNDF	Boundary F-measure.
CLIPS	Communication Langagièrre et Interaction Personne-Système.
CLR	Cross Likelihood Ratio.
CMS	Cepstral Mean Subtraction.
CMU	Carnegie Mellon University.
CTS	Conversational Telephone Speech.
CVN	Cepstral Variance Normalisation.
DCT	Discrete Cosine Transform.
DER	Diarisation Error Rate.
DHC	Divisive Hierarchical Clustering.

DNN	Deep Neural Network.
DPC	Dynamic Programming Cost.
E-HMM	Evolutive HMM.
EM	Expectation Maximisation.
FA	False Alarm.
FDLP	Frequency Domain Linear Prediction.
g-SEGF	Gaussian Segment F-measure.
GLR	Generalised Likelihood Ratio.
GMM	Gaussian Mixture Model.
HDP	Hierarchical Dirichlet Process.
HDP-HMM	Hierarchical Dirichlet Process HMM.
HMM	Hidden Markov Model.
IB	Information Bottleneck.
ICR	Information Change Rate.
ICSI	International Computer Science Institute.
IHM	Individual Headset Microphone.
IIR	Institute for Infocomm Research.
ILP	Integer Linear Programming.
KL	Kullback-Leibler.
LIA	Laboratoire Informatique d'Avignon.
LIUM	Laboratoire d'Informatique de l'Université du Maine.
LPCs	Linear Predictive Coefficients.
MAP	Maximum A Posteriori.
MDE	Metadata Extraction.

MDM	Multiple Distant Microphones.
MFCCs	Mel Frequency Cepstral Coefficients.
MGB	Multi-Genre Broadcast.
ML	Maximum Likelihood.
MLP	Multilayer Perceptron.
MMM	Multiple Mixed Microphones.
MS	Missed Speech.
MVDR	Minimum Variance Distortion Less Response.
NCLR	Normalised Cross Likelihood Ratio.
NIST	National Institute of Science and Technology.
NLDA	Nonlinear Discriminant Analysis.
NMI	Normalised Mutual Information.
NN	Neural Networks.
NTU	Nanyang Technological University.
OOD	Out-Of-Domain.
pdf	Probability Density Function.
PLPs	Perceptual Linear Prediction Coefficients.
PRC	Precision.
RCL	Recall.
RT	Rich Transcription.
SAD	Speech Activity Detection.
SDM	Single Distant Microphone.
SE	Speaker Error.
SEGF	Segment F-measure.
sIB	Sequential Information Bottleneck.
SNR	Signal-to-Noise Ratio.
ssDNN	Speaker Separation Deep Neural Network.
STT	Speech-to-Text.

t-SEGF	Triangular Segment F-measure.
TDOA	Time-Delay-of-Arrival.
UBM	Universal Background Model.
UPC	Universitat Politècnica de Catalunya.
UPM	Universidad Politécnica de Madrid.
VAD	Voice Activity Detection.
VB	Variational Bayes.
WER	Word Error Rate.
ZCR	Zero Crossing Rate.

Chapter 1

Introduction

Contents

1.1 Speaker Diarisation

Within the speech technology field, speaker diarisation is a major topic which is relatively new and has been growing in popularity (Miró et al., 2012). It focuses on the question “who speaks when?” within a given audio file. Research began as a by-product or on the side of other tasks until researchers realised that it was worth its own attention. The amount of “spoken documents" (Tranter and Reynolds, 2006) is continually growing through voice mails, meetings and media broadcasts. Every document can contain an abundance of information which is currently only accessible through transcripts or by listening to them, which can be time consuming. Having this information would improve the searchability of audio similarly to how search engines explore text documents. Diarisation is useful for improving speech recognition, transcript readability, in the area of information retrieval, for audio indexing and so on.

The task is considered unsupervised, meaning no prior knowledge or supplementary data is used to enhance the performance. This is opposed to semi-supervised or supervised methods which may use the expected number of speakers or speaker models, for example. A

system consists of several stages which can be performed separately, as a step-by-step method, or simultaneously, as an integrated method. The Speech Activity Detection (SAD) stage aims to detect regions of speech and remove any silence and nonspeech. These speech segments are then split further into speaker-pure segments in the speaker segmentation stage. This is also known as speaker change detection. The last stage clusters the speaker-homogeneous segments into groups which each represent a different speaker. The final number of clusters detected is the hypothesised number of speakers in the recording. The desired output is speaker labelled segments of speech with accurate timing information, or boundaries.

Over the years, speaker diarisation has been performed on many domains. Initial work focussed on Conversational Telephone Speech (CTS) and Broadcast News (BN) before moving to meeting data with help from the National Institute of Science and Technology (NIST) Rich Transcription (RT) evaluations. Evaluations encourage systems to be created using the latest and novel techniques which leads to breakthroughs in the field. A state-of-the-art system proposed in the RT07 evaluation by the International Computer Science Institute (ICSI)[1] is still current today with many systems since being based on it. Their system used a Gaussian Mixture Model (GMM) for silence and a Hidden Markov Model (HMM) for speech in the SAD stage. Iterations of Agglomerative Hierarchical Clustering (AHC) using a HMM-GMM approach with a Bayesian Information Criterion (BIC) decision metric and stopping criterion followed by Viterbi decoding are performed. This is presented by Wooters and Huijbregts (2007) and described in more detail in Section 2.8.

Neural Networks (NN) are growing in popularity in the speech technology field and beyond and have begun to be incorporated into speaker diarisation systems. Features optimised for diarisation have been extracted from the bottleneck layer of an Speaker Separation Deep Neural Network (ssDNN) with comparable results to Mel Frequency Cepstral Coefficients (MFCCs) by Yella and Stolcke (2015). Dines et al. (2006) and Saz et al. (2015) have successfully trained a Deep Neural Network (DNN) model on speech and nonspeech data to improve the SAD stage. Model-based speaker segmentation and a clustering method using an Auto-Associative Neural Network (AANN) is shown to have competitive performance by Jothilakshmi et al. (2009).

1.2 Broadcast Media Archives

As this world heads towards a more open and social media focussed view, more and more video and audio data is being created. The need for this data to be accessible grows every day. From an audio perspective, being able to search and retrieve a relevant segment of speech

[1] ICSI: http://multimedia.icsi.berkeley.edu/speaker-diarization/

WHERE: BBC Parliament TV, House of Commons
WHEN: Tuesday 8th November 2016
LINK: http://parliamentlive.tv/event/index/a9d4ea0c-fcee-4d5d-bc1c-e4ae033be83a?in=12:28:47
SPEAKER: Caroline Lucas (Female)
START: 12:18:47
TEXT: ...government delays have meant that there's been almost two years since the last contract for different auction in support for offshore wind and that is undermining a vest of confidence...

Fig. 1.1 An example of a possible response to the search "female member of parliament environment energy". The audio search benefits from speaker diarisation, ASR, gender detection and more.

from a particular speaker quickly would help journalists, researchers and the public. This is successful for textual data, for example, search engines like Google[1] and Bing[2]. Speaker diarisation is a first step towards accessing audio databases as it detects the regions of time which contain speech and attributes speaker labels to each segment. Follow on tasks such as speaker linking (van Leeuwen, 2010) would match speakers across recordings, speaker identification would determine the real name of the speaker (Reynolds, 2002) and the gender of the speaker can be detected too (Tranter and Reynolds, 2006). Furthermore, Automatic Speech Recognition (ASR) would detect the words spoken. This will result in the ability to find audio recordings of answers to specific queries, such as "female member of parliament environment energy". A possible answer returned is seen in Figure 1.1. The UK's Parliament channel[3] currently allows for searching with keywords and for specific people.

The British Broadcasting Corporation (BBC) are world-renowned for their TV and radio programmes, broadcasting in 28 languages through the BBC World Service[4]. Decades ago data was destroyed as a way to clear the BBCs archives due to lack of space as well as a lack of a consistent archiving policy[5]. For example, the original video recordings of 97 classic Dr. Who episodes are lost forever. Fortunately, their archiving policy changed in 1978 to keep all data for recording onto videos to sell and to preserve it for historical and cultural reasons. This shift has led to a vast and unwieldy broadcast media archive spanning several decades.

[1]Google: https://www.google.co.uk/
[2]Bing: http://www.bing.com/?cc=gb
[3]ParliamentLive.tv: http://parliamentlive.tv/Commons
[4]BBC World Service: http://www.bbc.co.uk/worldserviceradio
[5]Dr. Who missing episodes: https://en.wikipedia.org/wiki/Doctor_Who_missing_episodes

Evaluations and challenges are useful methods to help improve and go beyond the state-of-the-art methods in the field. Different research groups are encouraged to participate with new research which can result in novel technologies and methods. In previous years evaluations in BN data have been run using French data (Galibert and Kahn, 2013; Galliano et al., 2006) and Spanish data (Zelenák et al., 2012a). However, it was not until 2015 that a challenge was held based on an archive of broadcast media data. The BBC supplied data to the participants of the Multi-Genre Broadcast (MGB) to compete in four tasks: transcription, alignment, longitudinal Speech-to-Text (STT) and longitudinal speaker diarisation (Bell et al., 2015). English data was supplied and the next challenge will be extended to Arabic data from Aljazeera TV. Multiple languages add challenges to the task as cultural differences in speaking styles and programmes can vary dramatically, and of course programmes from a single culture can vary widely from the scripted and professional BN shows to ad-lib comedy panel shows. The speaker diarisation task highlighted the difficulties systems have with broadcast media data given baseline results at nearly 50% Diarisation Error Rate (DER).

1.3 Challenges

Diarisation systems face many challenges which can add difficulties to the task of determining who is speaking when. Meeting data has challenges ranging from the quality of the microphones to their unscripted and ad-libbed nature. Background noises such as air-conditioners or external noises can be recorded alongside vocal noises such as coughs and heavy breathing. For the broadcast media domain, the issue of poor quality microphones is reduced as many radio and TV programmes are recorded in a studio or the speech is rerecorded after filming. However, due to the range of programmes the amount of background noises increases with more varieties possible. From laughter in comedy programmes to music in the credits for most TV shows and in the background of dramas, documentaries and more.

Part of the problem for diarisation systems is in their evaluation, and this depends on their follow on task, if any. The standard metric is known as the Diarisation Error Rate (DER) and is the sum of Missed Speech (MS), False Alarm (FA) and Speaker Error (SE). A system is evaluated and believed to be competitive or successful if its DER is similar or better to other state-of-the-art systems. This leads to systems being built with the aim to reduce the DER. Little consideration is taken to the use of the hypothesised output. For example, accurate hypothesised segments, meaning speaker-pure regions of speech, are invaluable to ASR systems. A transcription system presented by Saz et al. (2015) relied on precise performance from the speaker diarisation system. The DER is time-weighted and evaluates the performance in terms of the percentage of time being incorrect, relative to the reference

duration. This does not give information of the segmentation quality, in terms of the number of correctly detected segments. In fact the DER masks over-segmentation (detecting too many segments) and under-segmentation (detecting too few segments).

Despite the task being considered unsupervised, there is a strong argument for a more supervised approach. An unsupervised method is arguably more robust leading to acceptable data independent performance. A robust system is defined as one which performs well under any conditions and only a minimal loss of performance is seen when presented with unpredictable data (Huijbregts, 2008). A speaker diarisation system is robust when good performance is seen across different domains, however many systems, including public domain toolkits, are designed specifically for certain data types (Huijbregts, 2008; Rouvier et al., 2013; Vijayasenan and Valente, 2012). This shows how better performance can be achieved through tailored systems. Meeting corpora and broadcast media archives typically contain Individual Headset Microphone (IHM) channels and transcripts which contain additional information that may benefit a system. For datasets which lack these types of supplementary data, speaker models are often available or trainable from other large datasets. Introducing a pretrained speaker model is assumed to enhance a speaker diarisation system.

1.4 Research Objectives

Research objectives for this thesis are presented and can be grouped into two types: investigating evaluation metrics (Objective 1) and investigating semi-supervised diarisation methods (Objective 2 and Objective 3).

Objective 1: Investigate a segment-oriented evaluation metric

Speaker diarisation is seen as a prerequisite task to audio indexing, ASR and more. In the case of ASR, accurate segment boundaries is vital information. Studies have shown that segmentation closer to the reference or ground-truth provides better performance in ASR systems (Hinton et al., 2012). If a system is relying on a speaker diarisation output, it is necessary to produce an output as accurate as possible. It is also important to know how accurate different outputs are. As in all fields, there needs to be a method of judging system performance to allow for comparisons across systems. The standard metric in diarisation is the DER (Miró, 2006). This evaluates several aspects of a system in a time-weighted approach. Missed speech and false alarm speech cover the segmentation quality and a speaker error is calculated to judge the hypothesised clusters. Other metrics exist such as the Dynamic Programming Cost (DPC) (van Vuuren et al., 2013) and Boundary F-measure

(BNDF) (Ajmera et al., 2004) which evaluate the segment boundaries and speaker and cluster purity which consider the speaker labelling quality (Ajmera et al., 2002). There are two desired aspects in a diarisation hypothesis: high quality segmentation (speech and speaker boundaries) and speaker labels resulting from the clustering. In a perfect situation there would be a single metric to evaluate both of these aspects. This rules out the boundary measures and the purity measures, leaving the DER. The DER is known to have disadvantages. It prioritises large clusters (Miró et al., 2012) and as it is time-weighted it does not measure correctly detected speech segments (Pardo et al., 2012).

The first objective of this thesis is to investigate an alternative segment-oriented evaluation metric. It focuses on evaluating the segmentation quality by matching correctly detected hypothesis segments with reference segments. A speaker mapping algorithm, mapping reference speaker labels to hypothesised cluster labels, is investigated to encompass speaker error within the metric.

Objective 2: Investigate speaker diarisation methods using auxiliary data

The typical diarisation system uses an unsupervised approach, meaning no prior knowledge is used within the system. The audio or features are used as input and the output are speaker labelled segments with time information. This has been the norm and the aim for many systems for many years. The argument for this is efficiency and time as adding prior knowledge could increase the computational cost and time taken to process the audio and extra information. Additionally, this may not make the system applicable to different datasets. Unsupervised systems have the advantage of making decisions for each recording based on the data and any extracted features of that single recording. This means there are no models to be pretrained and no extra information, like the number of speakers, is used. However, the major advantage of supervised techniques is the fact that outside knowledge and information can be used to guide the system on making decisions. For example, detecting speech using a tailored pretrained DNN model enhances the SAD stage (Dines et al., 2006) and knowing the expected number of speakers leads to the clustering stage stopping at the correct point (Moraru et al., 2004a). Semi-supervised methods use additional data and information not taken directly from the recording being assessed. Auxiliary data encompasses both prior information and supplementary data. Prior information refers to metadata which is data about data. Supplementary data refers to physical data other than the recording channels. For example, meeting data corpora typically contain IHMs (Carletta et al., 2005), referred

to as speaker channels, and broadcast media data contains transcripts or subtitles of the recordings (Bell et al., 2015).

The second objective of this research is to investigate semi-supervised speaker diarisation methods using supplementary data. Acoustic information in the form of IHMs and timing information in the form of transcripts are incorporated into five proposed systems. Some of these systems are step-by-step methods and others are integrated. When a system contains elements prone to performing poorly due to negative data effects, such as overlap and crosstalk, techniques to compensate are considered. Datasets which include both transcripts and IHM channels from the meeting and broadcast media data are investigated.

Objective 3: Investigate semi-supervised DNN-based clustering

Speaker information supplementary data also exists in the form of speaker models. These have been employed in speaker identification and verification tasks (Reynolds, 2002) as well as speaker linking, diarisation across data collections (van Leeuwen, 2010). For diarisation, speaker models can be built in an unsupervised fashion. The ICSI-RT07 system (Wooters and Huijbregts, 2007) builds a GMM for each cluster during an initial clustering and uses them in Viterbi decoding to resegment the data. Proposed by Konig et al. (1998), an ssDNN is trained on speaker data and aims to classify or separate speakers. A bottleneck layer allows to constrict the information learnt through the network and contain it in a reduced dimensionality. Features have been extracted from the bottleneck layer to be used in speaker diarisation systems (Yella et al., 2014) as well as ASR systems (Liu et al., 2014). Integrated systems are preferred over step-by-step methods, as errors are known to propagate and increase through multiple stages (Ajmera and Wooters, 2003). The nature of the proposed method allows for segments to be split further, leading to the clustering and speaker segmentation stages to be integrated.

The third objective is to investigate a DNN-based clustering technique for speaker diarisation. A pretrained ssDNN makes the method semi-supervised and an investigation into the different topologies for various domains is required. Furthermore, the proposed method allows for speaker segmentation to be performed at the same time, leading to it being an integrated method. The iterative process is investigated in terms of its tunable parameters to see how each affects the two main aspects of the output, the segmentation and cluster quality. The method is applied to SAD segmentation for datasets from both the meeting and broadcast media domain.

1.5 Thesis Overview

The thesis consists of seven chapters, including this introduction. The structure of the thesis is seen in Figure 1.2 in which there are four parts: background, evaluation, semi-supervised methods and conclusions. The fundamentals of speaker diarisation techniques are reviewed in Chapter 2. Chapter 3 investigates the data applicable and the challenges a diarisation system can face. This leads to a new segment-based evaluation metric which is proposed in Chapter 4. Chapter 5 presents semi-supervised methods involving acoustic and timing information. This is followed by a semi-supervised speaker information method in Chapter 6 which involves DNNs in a clustering technique. Lastly, Chapter 7 discusses the conclusions to the thesis.

Chapter 2: Speaker Diarisation Technology

Chapter 2 is an introduction to the field of speaker diarisation. The theoretical framework is presented along with different approaches to building systems. The various stages to a speaker diarisation system are provided in terms of the current state-of-the-art or most common method. This includes feature processing, SAD, speaker segmentation (or change detection) and speaker clustering. The standard evaluation metric and other alternatives are described. It finishes with a description of the best submission in RT07 and an overview of submissions for the final evaluation, RT09.

Chapter 3: Data Analysis

Chapter 3 discusses the challenges which affect a speaker diarisation system. This includes which stage in a system is most vulnerable to errors and how the different data domains can impact a system. Overlap and crosstalk are common causes of errors and are discussed. Meeting and broadcast domain datasets are presented in the form of training and test sets. Three public domain toolkits built with different methods and designed for specific data types are described and applied in an analysis of the data. The segmentation and speaker labelling quality is investigated to see how they affect the performance. The evaluation metrics are discussed in terms of what they show and what they ignore.

Chapter 4: Segment-Oriented Evaluation

Leading on from Chapter 3, Chapter 4 discusses the disadvantages to the evaluation metrics. This results in a segment-based metric being proposed as an alternative metric for evaluating speaker diarisation systems. The metric is detailed in terms of matching segments and

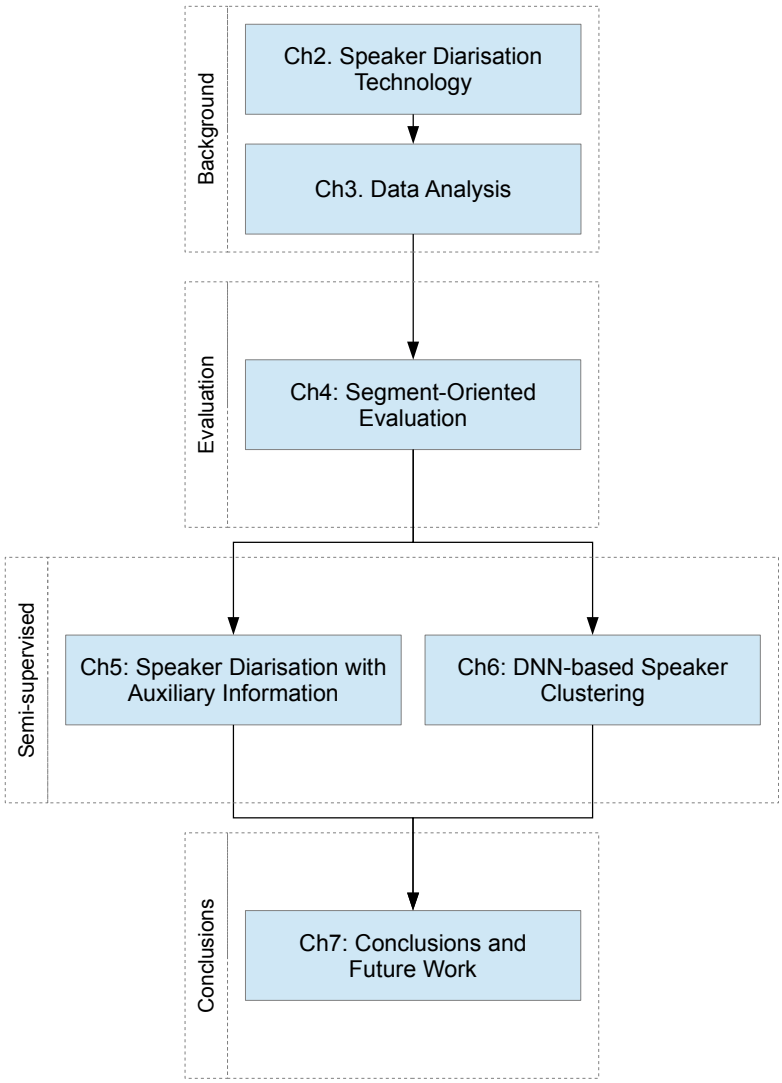

Fig. 1.2 Overview of the thesis structure in which there are four defined sections: background research, evaluation metrics, semi-supervised methods and final conclusions and future work.

mapping cluster and speaker labels. The F-measure is used as the evaluation method. The final section compares the Segment F-measure (SEGF) with the DER and other metrics. It is shown how the SEGF gives a different understanding of an hypothesised output. This chapter focusses on Objective 1.

Chapter 5: Speaker Diarisation with Auxiliary Information

Chapter 5 discusses semi-supervised diarisation as opposed to the typically unsupervised diarisation methods. A method is presented which involves timing information in the form of

transcripts. This is followed by three methods involving acoustic information in the form of IHM channels: a Single Distant Microphone (SDM) and/or IHMs. A final method is proposed combining all three types of supplementary data. These methods are evaluated on the test sets which contain IHMs. This chapter focusses on Objective 2.

Chapter 6: DNN-based Speaker Clustering

Chapter 6 implements a speaker clustering method involving DNNs. Initial research into this method is discussed followed by the method itself being presented. The main steps of the iterative method are described: training an ssDNN, reconstructing new DNNs, adaptation, decoding, applying a stopping criterion and, finally, filtering segments if the stopping criterion is not met. The method is investigated in terms of parameters using a small test set from both the meeting and broadcast media domain in which the reference segmentation is applied. Having decided on the parameters which produce the best performance for each domain, the method is given true SAD segments and evaluated. This chapter focusses on Objective 3.

Chapter 7: Conclusions and Future Work

Chapter 7 summarises the main contributions that the thesis has addressed. This is followed by a list of publications achieved related to the research. Finally, future research avenues related to the contributions is discussed.

Chapter 2

Speaker Diarisation

Contents

Speaker diarisation aims to answer the question "who spoke when?" in a spoken document. The task is to segment the audio into speaker-homogeneous segments of time which are associated to a single speaker. The input can vary depending on the system, dataset, etc, however the output or aim always remains the same, to detect speaker labelled speech segments with timing information.

Diarisation generally consists of four stages which are displayed in Figure 2.1, these are: feature processing, Speech Activity Detection (SAD), speaker segmentation and speaker clustering (Miró et al., 2012; Tranter and Reynolds, 2006). Feature processing is seen as a preprocessing stage. Features are extracted from the audio as the input for the next stage. SAD aims to detect regions of audio which contain speech and in doing so also detect other noises and silence, which are referred to as nonspeech. The speaker segmentation stage, also known as speaker change detection, further segments the speech by detecting speaker boundaries which are the points in time where the speaker changes. This results in speaker-homogeneous segments, which are next clustered together into groups representing the same speaker. An automatic stopping criterion ends the process when the predicted number of clusters, representing speakers, is determined. The different methods which are common in each of these stages are detailed in this chapter. A useful framework for the theory of diarisation has been presented by Evans et al. (2012). Equations for the segmentation and clustering stages have been formulated which provide a speaker sequence corresponding to a segment sequence. This is the desired output of a system, speaker-pure segments with timing information.

There are different approaches to building a diarisation system which can be classified as either step-by-step or integrated approaches (Meignier et al., 2006). Step-by-step refers to having distinct stages in which one must be completed before the next begins. Integrated

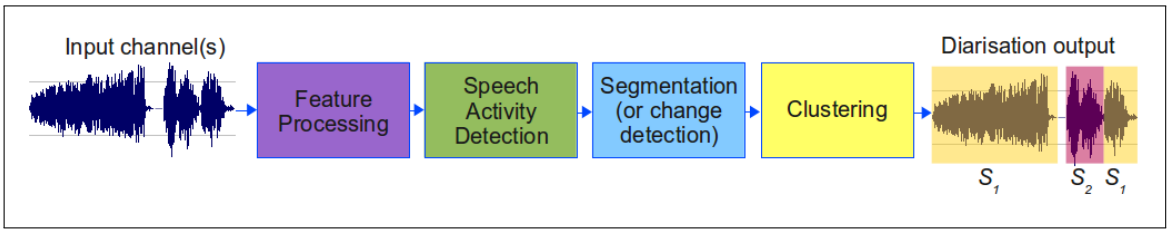

Fig. 2.1 Four stages of speaker diarisation are depicted: feature processing, SAD, speaker segmentation or change detection and finally speaker clustering. The input can vary, however, the desired outputs are speaker labelled segments with a start and end time.

means speaker segmentation and clustering are performed together in the same stage. Diarisation is typically an unsupervised task which implies no additional information or data is used in a method. For example, the number of speakers is not known a priori. In machine learning, unsupervised refers to methods without access to labelled data. Semi-supervised and supervised learning methods have been applied to diarisation systems in several ways, ranging from knowing the number of speakers in the recording (Moraru et al., 2004a) to using labelled speech data for the SAD step (Wooters and Huijbregts, 2007). Evaluations of diarisation were run by NIST until 2009. In the RT07 evaluation, the ICSI system (Wooters and Huijbregts, 2007) achieved the best performance for both Multiple Distant Microphones (MDM) and SDM channel systems. The implementation became the standard diarisation system and many later systems were based on it, including in the RT09 evaluation.

The chapter is organised as follows. A theoretical framework for diarisation is presented in Section 2.1 and different approaches to systems are discussed in Section 2.2. Next, the four steps of a system are described: feature processing in Section 2.3, SAD in Section 2.4, speaker segmentation in Section 2.5 and Section 2.6 describes speaker clustering. Finally, Section 2.7 presents the various evaluation metrics and Section 2.8 gives an overview of the NIST RT evaluations. Section 2.9 contains the chapter summary.

2.1 Theoretical Framework

It is useful to define the task mathematically and formulate the problem of diarisation. Being able to split the equation into smaller, more achievable parts will help to make the task simpler. The task of diarisation has been formalised in a probabilistic framework by Evans et al. (2012). This derivation is in compliance with the assumptions and methods used by all participants of the most recent NIST evaluation, RT09. A diarisation system should take an audio file as the observation, O, and output the speaker sequence, S, with the associative

segment sequence, G:

$$(\widetilde{S},\widetilde{G}) = \arg\max_{S,G} P(S,G|O) \tag{2.1}$$

This equation is suggested to define the task of diarisation where $\widetilde{S}$ is the optimised speaker sequence and $\widetilde{G}$ is the optimised segmentation, for example who (S) spoke when (G). It is not easy to compute $P(S,G|O)$ so the first step is to convert it into a posterior probability by applying Bayes' rule:

$$\begin{aligned}(\widetilde{S},\widetilde{G}) &= \arg\max_{S,G} \frac{P(O|S,G)P(S,G)}{P(O)} \\ &= \arg\max_{S,G} P(O|S,G)P(S,G)\end{aligned}$$

where $P(O)$ can be ignored as it is independent of S and G. The above optimisation gives acoustic models, $P(O|S,G)$, which detail attributes of speakers and speaker turn models, $P(S,G)$, which tells the probability, given a segmentation, of turns between speakers.

Gaussian Mixture Model (GMM)s are the standard for representing the acoustic model distribution:

$$P(O|S,G) = \prod_i P(O_i|\lambda_{S_i},G_i) \tag{2.2}$$

where λ_{S_i} is the GMM for speaker S_i. Speaker S_i is the ith speaker in S and the corresponding speech segment in G is labelled as O_i. This has made the acoustic models more attainable.

With regards to the speaker turn models, an assumption is made that the speaker labels either side of the turn are not important and then only take duration into account. This gives:

$$P(S,G) = P(G) \tag{2.3}$$

where it has been assumed that a uniform distribution will suffice which omits the turn model completely. Combining this with Equation 2.2 gives:

$$(\widetilde{S},\widetilde{G}) = \arg\max_{S,G} P(G)\prod_i P(O_i|\lambda_{S_i},G_i) \tag{2.4}$$

which results in a solution to the problem of speaker diarisation. The basic system process is seen in Figure 2.1 which is concluded from Equation 2.4. The segmentation is represented by $P(G)$ and clustering is represented by the GMM $\prod_i P(O_i|\lambda_{S_i},G_i)$.

2.2 Approaches

There are various ways to sort or group diarisation methods and two are considered. Firstly, methods can be grouped depending on the structure and implementation, and secondly, they can be grouped depending on how much external data is used within a system. For organising methods into their structure type, two groups are defined:

- **step-by-step:** separate stages performed sequentially.
- **integrated:** segmentation and clustering performed simultaneously.

The former performs different stages of a system separately whereas the latter performs stages simultaneously. The classical approach performs the four stages seen in Figure 2.1 one after the other, as clearly defined steps (Gauvain et al., 1998). One clear drawback is that errors made in an early stage can not be fixed later which leads to errors propagating through a system (Ajmera and Wooters, 2003). An integrated approach aims to perform diarisation in one stage or step. Although any stages can be integrated, typically in the field the speaker segmentation and clustering stages are integrated and performed simultaneously. An overview of the two defined types, specifically for the Broadcast News (BN) domain, is presented by Meignier et al. (2006).

Another way to organise approaches to diarisation methods is to consider whether they are unsupervised or involve a certain amount of external information data. The common choice for a speaker diarisation system is to be unsupervised. This means no prior knowledge is provided and models are not trained on external data. Many systems are considered unsupervised despite having a few heuristically-trained parameters, such as number of initial clusters (Sinclair and King, 2013). Many diarisation systems are designed not to contain tunable parameters meaning the system will be robust against changes in audio conditions or the data domain (Huijbregts and Wooters, 2007). In terms of machine learning, approaches to learning can be organised into three groups:

- **unsupervised:** no labelled data and algorithms must learn from the input data only.
- **semi-supervised:** some labelled data is allowed.
- **supervised:** all data is labelled and algorithms aim to predict the output from the input.

For diarisation, many systems operate in an unsupervised fashion. For example, the number of participants in the recording is not known a priori. Additionally, only the input recording is usable by a system, no external data or information is used to help enhance performance. The advantages of unsupervised methods are that there is no need for training or development

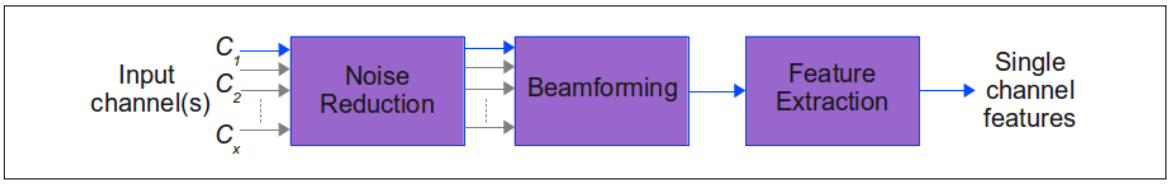

Fig. 2.2 Feature processing varies depending on the number of channels. Many channels allows for noise reduction followed by beamforming which produces a single channel. Features are then extracted for the single channel.

data and a system is portable to different datasets (Ajmera and Wooters, 2003). The desired output is the same in every case, speaker-pure segments with timing information, but varying amounts and types of data and information is applied to a system.

2.3 Feature Processing

The front-end of a speech diarisation system requires the audio channel(s) as input and creates a set of features containing the useful and important speaker information for the next stage. There are three types of channels

- **Single Distant Microphone (SDM):** a single microphone placed nearby.
- **Multiple Distant Microphones (MDM):** multiple microphones placed nearby.
- **Individual Headset Microphone (IHM):** a microphone designed to pick up a single speaker's voice.
- **Multiple Mixed Microphones (MMM):** close microphones, mostly individual which are mixed in the final signal, typical for broadcast data.

The process is displayed in Figure 2.2 and begins with a noise reduction algorithm applied to the channel(s). If more than one channel exists, beamforming is usually carried out to produce a single beamformed channel using all the given information from the MDMs. Lastly, features are extracted from the single channel in a process which recognises the most important cues from the data and removes insignificant data depending on the domain knowledge.

2.3.1 Noise Reduction

Microphones vary in quality due to different types and styles. In particular, portable microphones may be subject to low frequency noise including breathing and speaker head motion

which affects power levels (Jin and Schultz, 2004). This means the raw audio collected from one microphone may not be the same as the next. One may have no background noise whereas another in the same room could have picked up the air conditioner, for example. This poses a issue as it affects the quality of the speech which has been recorded.

Noise reduction for diarisation is commonly carried out using Wiener filtering with the goal to remove corrupting noise from the data (Adami et al., 2002). It calculates the amount of background noise or the Signal-to-Noise Ratio (SNR) and for MDMs, it can help to reduce crosstalk across the channels. Adami et al. presented the Qualcomm-ICSI-OGI Aurora toolkit[1] which was popular in several NIST RT evaluations. A modified Wiener filter algorithm is applied to the power spectra and an instantaneous filter is estimated for every frame using the equation:

$$|H_{inst}(k,m)| = \max\left(\frac{|X(k,m)|^2 - \gamma(k)|\hat{W}(k,m)|^2}{|X(k,m)|^2}, \beta\right) \tag{2.5}$$

where $|X(k,m)|^2$ and $\hat{W}(k,m)|^2$ are the power spectral estimates of noisy speech and additive noise signal respectively (k and m are the time and frequency indices). The spectral floor parameter β is set to 0.01 to avoid negative or very small transfer function components and the noise estimation factor $\gamma(k)$ is a function of the local a posteriori SNR. Equation 2.5 is smoothed in time and frequency which helps to reduce variance due to erroneous noise spectral estimates. The clean speech power spectral estimate is then obtained by multiplying the smoother filter and noisy speech power spectrum:

$$|\hat{S}(k,m)|^2 = \max(|X(k,m)|^2 \cdot |H(k,m)|^2, \alpha) \tag{2.6}$$

where $\alpha = 0.001 \cdot |\hat{W}(k,m)|^2$ is the noise floor. This method of Wiener filtering is applied to all input signals and helps to improve the performance of the SAD stage (Sun et al., 2010). Noise reduction requires an initial segmentation into speech and nonspeech parts. Systems can use SAD component of the Qualcomm-ICSI-OGI front end (Wooters and Huijbregts, 2007) for this, then a different SAD stage for the diarisation system occurs after the features are processed.

2.3.2 Acoustic Beamforming

A single channel contains all the speech from every speaker. Multiple channels means that there are various microphones in the room or perhaps each speaker is associated to a microphone. These microphones may be in front of the speaker, head-mounted or on the

[1] Qualcomm-ICSI-OGI Aurora toolkit: https://github.com/chinshr/qio

lapel. If each speaker is associated to one microphone, then one audio recording refers to one speaker so the number of speakers is known. This allows for a simple SAD method to perform on each audio file which easily detects the segments for each speaker. However, it is not as straight forward and several problems exist. Again with MDMs, there may not be an even quality if the microphones are different types. Another problem is the possibility of crosstalk across the channels which is discussed in Section 3.1.2.

For MDM channels, a beamforming technique is used which outputs a single channel of audio that combines information from all the channels provided. It produces a time domain estimate of the clean signal based on multi-channel information (Astudillo et al., 2013). The toolkit BeamformIt[1] was originally created for the RT05 evaluation. It uses the delay-and-sum technique to extract the signal output and correct misalignments due to the Time-Delay-of-Arrival (TDOA) of the speech to each microphone:

$$y[n] = \sum_{m=1}^{M} W_m[n] x_m[n - TDOA^{(m,ref)}[n]] \tag{2.7}$$

where $W_m[n]$ is the relative weight for microphone m (of M total microphones) at time n, with the sum of all weights equal to 1; $x_m[n]$ is the signal for each channel and $TDOA^{(m,ref)}[n]$ is the relative delay between each channel and the reference channel (Anguera et al., 2007). Beamforming has proven successful in many areas of speech technology including keyword recognition as shown by Astudillo et al. (2013) and the separate TDOA estimates have been used successfully to improve diarisation (Miró et al., 2012). TDOA features used in beamforming can also be used for speaker localisation (Miró et al., 2012). Tracking algorithms are used, which allows for speakers to move around a room, to create estimates of location features. Research has successfully combined standard acoustic features with inter-channel delay features (Pardo et al., 2007) which are combined at the weighted log-likelihood level, although weights were seen to vary across data. An entropy-based metric is used for automatic weighting for cluster comparison (Anguera et al., 2007) and an unsupervised discriminant analysis of inter-channel delay features is used by Evans et al. (2009).

2.3.3 Feature Extraction

Standard feature extraction in many speech technology tasks extracts Mel Frequency Cepstral Coefficients (MFCCs) (Davis and Mermelstein, 1980). To compute MFCCs, the equation is

[1] BeamformIt: http://www.xavieranguera.com/beamformit/

defined as:

$$\text{MFCC}_i = \sum_{k=1}^{K} X_k cos\left[i\left(k-\frac{1}{2}\right)\frac{\pi}{K}\right], \qquad i = 1,2,...,M \tag{2.8}$$

where M is the number of cepstrum coefficients and the log-energy output of the kth filter is represented by X_k (where $k = 1,2,...,K$). This equation is used to produce the cepstral coefficients, their derivatives and their second derivatives. Usually just the first 12 MFCCs are calculated but the higher order MFCCs (13-19) convey speaker information via the source characteristics from source-filter models of speech production. The first and second derivatives of the calculated MFCCs depict temporal differences. Also, one way to compensate for the intersection and inter channel variability is to compute the Cepstral Mean Subtraction (CMS) and Cepstral Variance Normalisation (CVN) (Jain and Hermansky, 1999).

Perceptual Linear Prediction Coefficients (PLPs) are calculated in a similar way to MFCCs and used as an alternative (Hermansky, 1990). The standard Mel-frequency filter bank is warped and used to calculate the coefficients which are then weighted and compressed. These auditory spectrum linear prediction coefficients are estimated and converted in the normal way stated above for cepstral coefficients. However, there is a disadvantage to these methods which use linear prediction envelopes as they tend to overestimate and overemphasise sparsely spaced harmonic peaks.

To reinforce the standard features, Vijayasenan et al. (2010) investigated using extra feature streams. Vijayasenan et al. proposed using MFCCs as the first stream followed by TDOA features (see Section 2.3.2), Frequency Domain Linear Prediction (FDLP) features and modulation spectrum features. Including these in the system input improved the performance by an absolute of 1.5% DER. FDLP features (Athineos and Ellis, 2003) provide an estimate of the temporal envelope. The idea is to model the temporal information instead of the spectral like the MFCCs. It is a flexible and adaptive representation which converts Linear Predictive Coefficients (LPCs) to cepstral-like temporal envelope coefficients. Modulation spectrogram features (Vinyals and Friedland, 2008) represent the slowly varying components of the short term spectrum. MFCCs are considered to be short term features as they are frame-based while modulation spectrogram features are taken from a longer segment of speech which helps to reveal information about the speaker's behaviour or voice characteristics.

MFCCs and PLPs are short term features created on a per frame basis. Long term features provide useful information for speaker discrimination (Miró et al., 2012). The features contain information about the characteristics of the speakers' voices and even some tells of their speaking behaviour. Friedland et al. (2009) carried out a systematic investigation of 70 long-term features. Most of these were prosodic and included pitch, energy, formants, harmonics-to-noise, and long-time average spectrum features. The research was extended

by Imseng and Friedland (2009, 2010) who presented an adaptive initialisation scheme combining the "adaptive seconds per Gaussian" method with a new pre-clustering method. Minimum Variance Distortion Less Response (MVDR) features are considered long-term features. MVDRs were used in speaker identification by Wölfel et al. (2009) who tried to preserve the unique characteristics of individual speakers. Warped MVDR features were also proposed as they better suit the spectral characteristics of human hearing and solved a disadvantage of the MFCCs which perform poorly in adverse conditions.

2.3.4 Discussion

Various results and comparisons with the above features are discussed, however, many papers either do not compare all the different features and of course the baselines are different from paper to paper. The relative change of performance from the baseline to the improved system is considered. Noise reduction has been tested by Wooters and Huijbregts (2007) to see at which stage in the feature extraction process it is best to apply the Wiener filtering. The baseline without using noise reduction gives 15.8% and applying to all components gives 8.51% DER. Acoustic beamforming has been used for all the systems submitted to RT09. This shows the success and performance improvements it brings. Fredouille et al. (2009) presents the RT09 submission from the Laboratoire Informatique d'Avignon (LIA) and Eurecom which included acoustic beamforming to their RT07 submission (as well as other improvements) which helped bring the MDM DER down from 24.2% to 17.7%. Successful results have been shown by Vijayasenan et al. (2010) which shows that including FDLP and MS feature streams instead of just MFCCs and TDOA features improve the system output from 11.6% to 8.3% DER.

2.4 Speech Activity Detection

The Speech Activity Detection (SAD) stage aims to remove any nonspeech and silence segments from the input audio channels. SAD, or Voice Activity Detection (VAD), is a vast and broad field in its own right (Ramirez et al., 2007). There are many approaches and methods but the most common types for diarisation systems are described.

Energy-based approaches assume the audio contains only speech and silence. The energy of short and usually overlapping windows is calculated and the silence is defined by the local minima detected. Areas containing high energy are deemed speech. This works well in high SNR conditions, but poorly when the SNR drops (Ramirez et al., 2007). The output to this approach is only speech and silence labelled segments. This is good for BN programmes

but would not be advisable for dramas and other shows involving background music or applause (Huijbregts, 2008). In these cases, applause, laughter and other background noises contain energy which would be picked up and labelled as speech.

Model-based approaches have better performance (Tranter and Reynolds, 2006) and aim to train one GMM for each class defined. Usually there are three classes, speech, nonspeech noise and silence, but in some cases more have been used such as laughter, applause, music, and music with speech (Gauvain et al., 1998; Miró et al., 2012). As models are pretrained, they are used online which means the method does not need access to the full data before it can make a decision (Kotti et al., 2008a). This means the method is semi-supervised if trained on data other than the final recordings or supervised if trained on the final recordings. The GMMs are trained and used as a Probability Density Function (pdf) for a HMM in which each state is connected to the other states. Viterbi decoding (see Section 2.5.7) is then performed using this HMM which gives the segmented labelled output. The advantage to this method is that it is straight forward to add segmentation classes (Huijbregts, 2008). However, both the GMMs and HMM involved need to be trained on some prior training data set. If the training data is not similar enough to the real audio then this can result in poor segmentation which leads to poor diarisation performance. An alternative to Viterbi decoding is a best model search using morphological rules (Meignier et al., 2006).

To make the best use of energy and model-based approaches, a hybrid version has been implemented (Wooters and Huijbregts, 2007). The first step consists of energy-based detection which labels segments of speech and silence which have high confidence. The second step uses this labelled data to train a silence and speech model which is then used in a model-based detector to produce the final segmentation output. The benefit of this detector over a model-based detector is that no other training data is used, all it takes is the data currently being used for the task.

Energy-based methods do not need any prior training to create models so are attractive in that respect. However, model-based approaches have better performance so it's worth using pretrained models to improve the SAD results. The models achieve a moderate Recall (RCL) rate and a high Precision (PRC) rate (Kotti et al., 2008b). Wooters and Huijbregts (2007) present a hybrid SAD that improves their previous system of 10.81% down to 8.51% DER.

2.5 Speaker Segmentation

The aim is to split the speech segments from the SAD stage into speaker-homogeneous segments in preparation for speaker clustering. A speech segment may contain multiple speakers talking if there are no pauses between the speech. The method of change detection

looks at smaller windows within the speech segments and checks if adjacent windows refer to the same speaker by performing hypothesis testing (Miró et al., 2012). The test considers each change point or window break and detects whether both segments belong to the same speaker or to two different speakers. The window size is not fixed but the size can dramatically affect the resulting segmentation. If duration is too short, poor Gaussian estimation, particularly of the covariance matrix, can be caused by the lack of data and this results in poor segmentation. However, windows which are too long can risk containing more than one change point which will then result in misses. A typical window size is 2 seconds which is incrementally increased (Kotti et al., 2008b). If the data is best modelled by two distributions, a change point is defined and the search begins from that point. If one distribution fits best, the window is increased and the search is repeated. Many metrics have been applied to decide whether the two windows belong to the same speaker or not and are discussed below.

2.5.1 Bayesian Information Criterion

The BIC and the related ΔBIC metrics are the popular approaches for segmenting the speech segments (Chen and Gopalakrishnan, 1998). It is an Maximum Likelihood (ML) criterion for model selection taken from statistics literature. BIC is a likelihood criterion which is penalised by the number of parameters in the model, the model complexity. The method chooses the model for which the BIC score is maximised. The BIC equation states:

$$BIC(M_i) = \log L(X_i, M_i) - \lambda \frac{m}{2} \log N_i \tag{2.9}$$

where X_i is an acoustic segment from data X at time i, M_i is a parametric model, N_i is the number of frames and m is the number of parameters. The likelihood $L(X_i, M_i)$ is maximised for the model. The quality of the match between the data and the model corresponds to the log likelihood and the second term represents the model complexity penalty where λ is tunable. A threshold of 0 is applied and positive value of $BIC(M_i)$ implies that there is a speaker change and that two GMMs best fit the data X at time i. The role of λ is seen as a threshold or penalty factor, however, the main disadvantage of BIC is the need to tune λ to the given data (Ajmera et al., 2004). When one model represents the two segments, there are more parameters which increases the likelihood. The penalty P is then applied to reduce the model complexity and penalise these likelihoods which prevents overfitting to the data (Delacourt and Wellekens, 2000). The BIC value shows how well a model fits the segment whereas the ΔBIC gives the distance between the two models:

$$\Delta BIC(i, j) = -R(i, j) + \lambda P \tag{2.10}$$

$$R(i,j) = \frac{N_{ij}}{2}\log|\Sigma_{X_{ij}}| - \frac{N_i}{2}\log|\Sigma_{X_i}| - \frac{N_j}{2}\log|\Sigma_{X_j}| \tag{2.11}$$

where the penalty for a full covariance matrix is:

$$P = \frac{1}{2}(d + \frac{1}{2}d(d+1))\log N \tag{2.12}$$

where d is the dimension of the acoustic space (Delacourt and Wellekens, 2000). The term N can be calculated in two ways. When N refers to the total number of frames in the recording, this is known as the global-BIC, whereas if N is calculated as the number of frames in both cluster i and cluster j, then this is known as the local-BIC (Tranter and Reynolds, 2004). The need to set λ is avoided when building M by merging the models M_i and M_j leading to the number of parameters before and after the merging are the same which removes the model complexities Ajmera and Wooters (2003). Aside from changing the threshold, this makes ΔBIC non-tunable and equivalent to the Generalised Likelihood Ratio (GLR), discussed in the next section (Miró, 2006). The advantage of BIC is that no prior knowledge is needed as BIC trains its own models, however, this leads to a disadvantage of computational cost as BIC uses full covariance distributions (Tranter and Reynolds, 2006). Another disadvantage is the high miss rates on short segments and poor performance when two speaker changes are less than 2 seconds apart (Kotti et al., 2008b). The problem of over-segmentation due to false alarms is less severe as it is easily fixable by clustering or merging in the next stage.

2.5.2 Generalised Likelihood Ratio

The Generalised Likelihood Ratio (GLR) is a likelihood-based metric which refers to the ratio between the two hypotheses of whether the clusters belong to the same speaker or not (Willsky and Jones, 1976). The GLR is defined as:

$$GLR(i,j) = \frac{L(X_{ij}, M_{ij})}{L(X_i, M_i)L(X_j, M_j)} \tag{2.13}$$

where X_i and X_j are two different segments with models M_i and M_j. The numerator represents the hypothesis that both clusters belong to the same speaker with the data combined as X_{ij} and the model for both as M_{ij}. Further, a distance is defined by taking the logarithm of the ratio:

$$D(i,j) = -\log(GLR(i,j)) \tag{2.14}$$

where D is the defined distance function. For diarisation, the GLR is mostly used with segments of the same length and a threshold is applied to decide whether the segments are

from the same speaker or not (Miró, 2006). The pdfs in this case are unknown and the data provided is used in their estimation.

2.5.3 Gish Distance

The Gish distance is a variation of the GLR distance and is also a likelihood ratio hypothesis test with the same goal as the GLR (Gish et al., 1991). The GLR is split into λ_{cov} and λ_{mean} and the background dependent part is ignored:

$$D_{Gish}(i,j) = -\frac{N}{j}\log\left(\frac{|S_i|^{\alpha}|S_j|^{(1-\alpha)}}{|W|})\right) \tag{2.15}$$

where S_i and S_j are the sample covariance matrices of each segment, $\alpha = \frac{N_i}{N_i+N_j}$, and the sample weighted average is $W = \frac{N_i}{N_i+N_j}S_i + \frac{N_j}{N_i+N_j}S_j$. The advantage with this metric is that no thresholds need to be defined which makes the metric robust to changes in the acoustic conditions (Kemp et al., 2000).

2.5.4 Kullback-Leibler Divergence

The Kullback-Leibler (KL) divergence is an information theoretic measure which aims to estimate the distance between two random distributions, X_i and X_j, (Siegler et al., 1997):

$$KL(X_i;X_j) = E_{X_i}(\log\frac{P_{X_i}}{P_{X_j}}) \tag{2.16}$$

where E_{X_i} is the expected value according to the pdf of X_i. The related KL2, or Gaussian divergence, is the symmetric version and an effective metric which assists the detection of long-term statistical difference in speech signals. When using GMMs the KL distance must be computed by either using sample theory or by using approximations (Miró, 2006). The KL2 is defined as:

$$KL2(X_i,X_j) = KL(X_i;X_j) + KL(X_j;X_i) \tag{2.17}$$

2.5.5 Information Change Rate

The Information Change Rate (ICR) is a distance metric which looks for changes or gains in information by merging any two segments. Unlike the above which are based on the model of each segment, it uses maximum mutual information or minimum entropy to consider the distance between segments in a space of relevance variables (Vijayasenan et al., 2009). It

describes each segment as a set of features. It is a normalised version of GLR which came from studies carried out into considering the GLR from an information-theoretic perspective. It is defined for two segments, i and j:

$$ICR(i,j) = \frac{1}{N_i + N_j} \ln GLR(i,j) \tag{2.18}$$

where N_i and N_j are the number of features in the feature vectors $x = \{x_1, x_2, ..., x_{N_i}\}$ and $y = \{y_1, y_2, ..., y_{N_j}\}$ (Han and Narayanan, 2008).

2.5.6 Hybrid

Hybrid systems attempt to combine metric and model-based approaches. The audio is pre-segmented using metric-based approaches and these resulting segments are used to create a set of speaker models. The model-based segmentation is then applied to refine the segments. The purer the initial metric-based segments are, the better the expected results will be from the model-based stage (Kim et al., 2005).

Meignier et al. (2006) combine two systems into one in an attempt to use the best aspects of both. The integrated Evolutive HMM (E-HMM) system by the Laboratoire Informatique d'Avignon (LIA) (Meignier et al., 2001) was combined with a step-by-step system which uses GLR segmentation followed by AHC using GMMs and the GLR from the Communication Langagièrre et Interaction Personne-Système (CLIPS) laboratory (Moraru et al., 2003). Two methods were tested. The first was named "hybridisation" where the CLIPS system output was fed into the LIA system. The second was called "merging" which merged preliminary outputs from both systems and used the LIA system for a final resegmentation stage.

BIC was also used alongside HMMs in a two-level clustering approach which improved metric only approaches (Kim et al., 2005). Segmentation was carried out on three levels: the segment level, the model level and the final HMM-based level. A sequential metric-based method was discussed by Wang and Cheng (2004), known as divide-and-conquer, where each speaker change had multiple chances to be detected by different analysis window pairs. An initial BIC segmentation was performed and any changes missed were determined using the different window pairs in a top-down manner. This made the method more robust than BIC and the computation cost is linear. Another hybrid metric is the DISTBIC which was used in a two pass segmentation approach (Delacourt and Wellekens, 2000). The first pass used the GLR distances to detect the speaker changes. This was followed by BIC which was used to validate or discard the previously detected changes.

2.5.7 Viterbi Decoding

This optional stage is typically performed after the clustering stage has finished or after every instance of a new cluster being created. It is referred to in several ways: realignment, resegmentation or decoding. When all the clusters (or just one new cluster) have been detected, Viterbi decoding helps to relabel the speaker homogeneous segments. Several iterations are performed until the output segments do not change. The nonspeech models from a model-based SAD stage can also be used. This helps to refine the original SAD segment boundaries and it also helps to fill in the short segments which may have been previously removed to improve robust clustering (Tranter and Reynolds, 2006).

2.5.8 Discussion

BIC and ΔBIC are the most popular choices for speaker segmentation and different research has compared several metrics using the BNDF. Mori and Nakagawa (2001) compared BIC to the GLR, along with other metrics, using Japanese BN data. The BIC gave higher performance than the GLR. GLR using full covariances as opposed to diagonal covariances gave an improved result but the highest GLR result, 75.9%, was lower than the equivalent setup with BIC which achieved 80.0%. The Gish metric was compared to the KL metric by Kemp et al. (2000). German BN data was investigated and using the BNDF, the results were close with 69% for KL and 70% for Gish. For both metrics, a higher RCL than PRC was seen. The ICR approach is computationally efficient and it was shown that when compared to a BIC-based metric, the ICR was more robust to data source variation (Han and Narayanan, 2008). Vijayasenan et al. (2010) investigated how successful a resegmentation stage is and improvements in the DERs were seen across the experiments. Combining Viterbi decoding with Expectation Maximisation (EM) training helped to further refine the segments (Evans et al., 2012).

2.6 Speaker Clustering

Speaker clustering aims to group acoustically similar segments together. The clusters produced represent the different speakers within the recording. The clustering algorithm must converge on the most apt number of speakers. Clustering refers to the acoustic model part of Equation 2.4 and a common iterative process is described simply by Tranter and Reynolds (2006):

0. Initialise clusters with speech segments

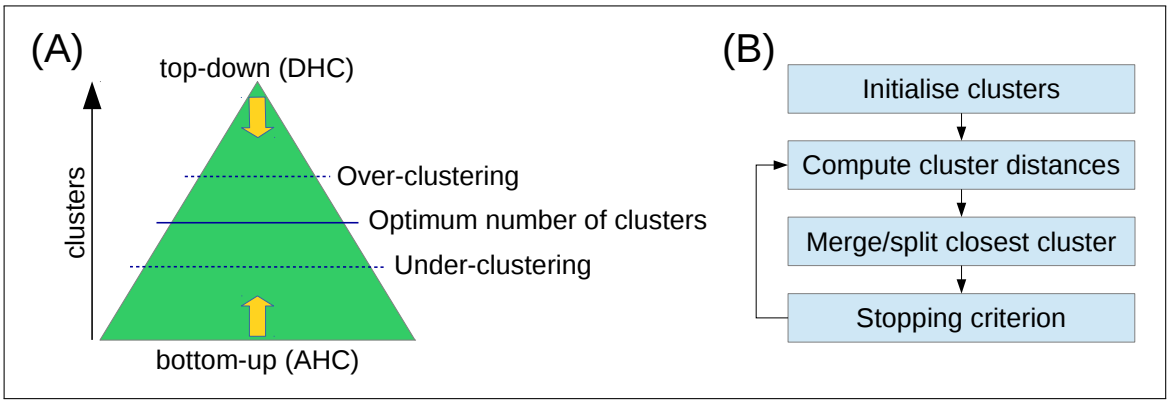

Fig. 2.3 AHC and DHC methods are shown in which (A) depicts how each approach reaches the correct number of speakers and (B) is a flowchart for the clustering process (Miró et al., 2012). AHC merges clusters to reduce the number of speakers whereas DHC splits the clusters to increase the amount of speakers.

1. Compute pair-wise distances between every cluster
2. Merge/Split the closest clusters
3. Update distances of previous clusters and the new cluster
4. Iterate steps 1. to 3. until a stopping criterion is met

Iterations of Viterbi decoding is performed and the models retrained after every new cluster is produced (Wooters and Huijbregts, 2007). The methods presented are sorted into two groups: bottom-up and top-down clustering (Evans et al., 2012). The former merges smaller clusters into larger clusters, Agglomerative Hierarchical Clustering (AHC), and the latter splits large clusters into smaller clusters, Divisive Hierarchical Clustering (DHC). An example of the two types of hierarchical clustering is shown in Figure 2.3. A simple clustering example technique is seen using a distance metric in which clusters of segments are evaluated on their similarity, and whether clusters belong to the same speaker or not. A decision to merge the clusters in the case of AHC or split the cluster in the case of DHC is iteratively carried out. The clustering method ceases when a stopping criterion is met. The aim of the stopping criterion is to stop the process when the correct number of clusters is determined. Additionally, an alternative HDP-HMM method is presented.

2.6.1 Agglomerative Hierarchical Clustering

AHC is also known as merging or bottom-up clustering and is the more popular technique in speaker diarisation. It begins by creating clusters which each contain a single segment. This

means there are more clusters than suspected speakers. A distance metric is applied to every possible pair of clusters and the pair deemed closest, or most similar, is merged to form a new cluster. This continues until a stopping criterion is met. Compared to DHC, these methods capture comparatively purer models and are therefore more sensitive to variations, making them less stable (Evans et al., 2012). Four AHC techniques are discussed: HMM-GMM with BIC (Wooters and Huijbregts, 2007), the Information Bottleneck (IB) (Vijayasenan et al., 2009), Variational Bayes (VB) (Kenny et al., 2010) and applying i-vectors (Dehak et al., 2011) from the speaker verification field.

2.6.1.1 HMM-GMM

HMMs are a common choice for AHC where each state represents a cluster and the pdf of a cluster is modelled by a speaker GMM (Ajmera and Wooters, 2003). GMMs make a popular choice for parametric pdf estimators as they predict any continuous pdf closely given a ample number of Gaussian components (Noulas et al., 2012). As it is a bottom-up approach, the data is over-clustered and the Viterbi algorithm is used to refine segments. Clusters are merged pairwise according to a likelihood ratio distance metric and the new class, GMM, is equivalent to the sum of the merged two. Parameters of the new GMM are then retrained using the EM algorithm and the segmentation is re-estimated with the new HMM topology which now has one fewer cluster. The likelihood of the data using this new segmentation is calculated and if the likelihood increases, the data in the two merged clusters are assumed to be from the same speaker. If the likelihood decreases, then they must refer to different speakers. Merging will then cease when the likelihood does not decrease any more.

Likelihood ratio distance metrics similar to those described in Section 2.5 are applied. The popular ΔBIC metric compares the two clusters X_i and X_j with the parent cluster X_{ij} to see if they should be merged. It is similar to the segmentation decision described in Section 2.5.1. The penalty factor in Equation 2.12 becomes:

$$P = (\frac{d(d+3)}{4}) \log N \tag{2.19}$$

for considering only two clusters, where d is the dimension of the feature vector and N is the total number of frames (Tranter and Reynolds, 2006). If the result of ΔBIC is below a threshold of 0 then two clusters should be merged and the two clusters imply two different speakers if it is above 0. The distance is calculated for each pair and the pair with the lowest ΔBIC is merged. The cluster merging is an iterative process which continues until a stopping criterion is met. For example, for ΔBIC, if the pair of clusters with the lowest value is above

a threshold, usually 0, the process is stopped and the current number of clusters is defined as the number of speakers detected.

The Normalised Cross Likelihood Ratio (NCLR) measure is applied as an alternative (Le et al., 2007; Rouvier and Meignier, 2012) and is calculated:

$$NCLR(M_i, M_j) = \frac{1}{N_i} log\left(\frac{L(X_i|M_i)}{L(X_i|M_j)}\right) + \frac{1}{N_j} log\left(\frac{L(X_j|M_j)}{L(X_j|M_i)}\right) \tag{2.20}$$

where M_i and M_j are two cluster models and X_i and X_j are speaker data. The Cross Likelihood Ratio (CLR), the ratio inside the first log, measures how well model M_j scores with speaker data X_i compared to model M_j with the same data. The log-likelihoods are normalised by the number of frames in the relevant data, N_i and N_j.

2.6.1.2 Information Bottleneck

The Information Bottleneck (IB) has come from rate-distortion theory and is based on an information theoretic framework. The aim is to cluster the segments using mutual information which measures the mutual dependence of two variables (Vijayasenan et al., 2009, 2011). A set of elements, or segments, X, is organised into a set of clusters, C. One GMM is trained for the entire audio and the mutual information computed is in the space of relevance variables, the set of Y, which are defined by the GMM components. The relevant variables depend on the given situation. For example, in ASR they can refer to the target sounds while in document clustering they represent the vocabulary of words. The objective function aims to minimise the loss of mutual information while preserving as much information as possible from the original dataset:

$$F = I(Y,C) - \beta I(C,X) \tag{2.21}$$

The two main techniques are Agglomerative Information Bottleneck (aIB) and Sequential Information Bottleneck (sIB). The aIB approach aims to maximise the objective function. It begins by using hard partitions of the data and $|X|$ clusters are used in the algorithm initialisation, where each data point is a cluster. These elements are then merged iteratively so that a decrease in the objective function at each step is as small as possible. The sIB approach aims to maximise the objective function in a given partition. It works with a fixed number of clusters, M, unlike the aIB, and the space is initially partitioned into M clusters. An element x is drawn out of it's now previous cluster C_{old} and is defined to be a new singleton cluster. Next, it is merged into C_{new} and the objective function is tested to see if is has either improved or stayed unchanged. This is performed for all $x \in X$ and the process is repeated several times until no change can be seen in the clustering assignment for

any input, $x \in X$. To avoid the aIB problem of finding local maxima, the approach is iterated several times with a random initialisation.

2.6.1.3 Variational Bayes

A Variational Bayes (VB) system is presented by Kenny et al. (2010) which aimed to bring large-scale factor analysis into the diarisation problem. The systems did so by using a VB framework and substituted eigenvoice and eigenchannel priors on the parameters of the speaker GMMs from the original VB work (Valente and Wellekens, 2005a). Using the rules of probability (marginalisation and conditioning) it built a hierarchical generative model of the speech which consists of three types of hidden random variable. These variables roles are to specify: assignment of segments to speakers, parameters of speaker GMMs and assignment of frames to Gaussians in the speaker GMMs. The second uses eigenvoices and eigenchannels as stated below while the third uses a Universal Background Model (UBM). A UBM is a GMM trained on lots of data from different speakers. The speakers are represented using eigenvoice models and assume the super-vectors are of the form:

$$M = m + Vy \tag{2.22}$$

where M is a randomly chosen speaker dependent super-vector, m is the speaker independent supervector, V is a rectangular matrix of low-rank in which the columns are the eigenvoices and the vector y has a normal distribution where the entries are the speaker factors (Reynolds et al., 2009). It is ended with Baum-Welch estimation of speaker GMMs along with iterative Viterbi resegmentation.

However, Kenny et al. (2010) uses CTS data where the number of speakers, two, is known as well as the speaker change detection segmentation. The initial segmentation does not have to be too accurate and a uniform segmentation of 1 second intervals (assuming silence has been removed during SAD stage) has been successful. VB has several advantages over the HMM-GMM approach with BIC as it avoids making premature hard decisions through BIC, and Bayesian methods are fully regularised, which means it is not subject to overfitting problems which maximum likelihood is prone to.

2.6.1.4 i-vectors

Features known as i-vectors have been used predominantly in speaker verification systems (Dehak et al., 2011). The i-vectors contain speaker-specific information and the most noticeable source of variability between them is expected to be attributed to speaker voice

differences (Shum et al., 2011). It is similar to the previous VB approach and Equation 2.22:

$$M = m + Tw \tag{2.23}$$

where M is a supervector, m is a stacked mean super-vector which comes from a GMM-UBM and T is a rectangular matrix of low-rank which spans the total variability subspace. As opposed to V, it is assumed in T that all segments belong to different speakers. Finally, w, also low-dimensional, has a normally distributed prior $N(0,I)$. The w is the i-vector, previously referred to as a speaker factor, which takes the place of the GMM as the speaker model. Using the segmentation output, an i-vector is extracted for each segment. To compare two i-vectors, the cosine distance is defined as:

$$CosineDistance(w_1, w_2) = \frac{(w_1)^t(w_2)}{||w_1|| \cdot ||w_2||} \tag{2.24}$$

where w_1 and w_2 are two i-vectors (Dehak et al., 2011). The i-vectors can be normalised by their magnitudes such that they all live on the unit hypersphere, this means that the angle between two i-vectors is the measure of distance (Shum et al., 2013). The clustering has been successfully performed using k-means (Shum et al., 2011) and spectral clustering (Shum et al., 2012) which are both based on the cosine distance.

The mean-shift algorithm was popularised in areas of computer vision and image processing and has been adapted for diarisation (Stafylakis et al., 2012). It is a non-parametric technique which has the advantage of not making assumptions about the shape of the data distribution (Senoussaoui et al., 2014). For any given data point, the nearest mode is determined using an iterative process to find centre-of-mass within a neighbourhood. The neighbourhood is then redefined around the centre-of-mass. The method requires some tuning, as the size of the neighbourhood must be defined. The size is necessary for determining the required size of a mode, or speaker in this case. Senoussaoui et al. defines the parameter for each conversation using the duration of the recording.

Integer Linear Programming (ILP) aims to restrict the unknown variables to be in the form of integers and the objective function and variables are linear (Dupuy et al., 2012; Rouvier and Meignier, 2012). The clustering framework is expressed as a form of a k-centre problem and the aim is to group N i-vectors into K clusters. The clustering aims to minimise the K and also aims to minimise the i-vector dispersion within each cluster. Two binary variables are defined where y_k refers to which cluster is selected and $x_{k,n}$ refers to whether

i-vector n belongs to cluster k. The objective function is:

$$z = \sum_{k=1}^{N} y_k + \frac{1}{F} \sum_{k=1}^{N} \sum_{n=1}^{N} d(w_k, w_n) x_{k,n} \tag{2.25}$$

where the first sum works out how many clusters are in the problem and the second part calculates the sum of the distances between the centre of cluster k and the i-vectors belonging to k. The function $d(w_k, w_n)$ represents the distance between the i-vector n and the centre of cluster k while F is a normalisation factor.

2.6.2 Divisive Hierarchical Clustering

The top-down, or splitting, approach is a Divisive Hierarchical Clustering (DHC) process. It begins with a single speaker model and detects new speakers from the larger speaker model. This is the opposite of the bottom-up approach. It is less popular, however, it is computationally convenient and has been improved through cluster purification (Evans et al., 2012). Initialisation happens by modelling the whole audio stream with one speaker model, S_0. A new model, S_1, is then introduced and is trained using appropriate data from the general speaker model, S_0. The appropriate segments of data can be selected using various methods, but the most consistent performance comes from using the largest segment detected during the SAD and segmentation stage (Evans et al., 2012). By repeatedly splitting the existing models, new speakers are added. Like the bottom-up method, a stopping criterion is in place to halt the process when the correct number of speakers has been determined. This could be when there is not enough data left to create a new speaker or when a preset upper speaker limit has been reached.

2.6.2.1 Evolutive HMM

This top-down method is based on an E-HMM modelling the conversation (Meignier et al., 2000, 2001, 2006) and was applied to LIA's RT09 submission (Bozonnet et al., 2010a). The states of the HMM represent the speakers and the changes between the speakers are modelled by the transitions. All the speaker changes are available at any time, making the HMM ergodic. This means it is possible for a state to transition to any other state. An iterative process creates the HMM which detects and adds a new state, a new speaker, at every iteration. The speaker detection process consists of 4 steps: initialisation, adding a new speaker, adapting speaker models, and finally, speaker model validation and assessment of the stopping criterion.

1. To initialise the process, a single speaker model, S_0, is trained on a whole recording. This means the HMM contains a single state representing a single speaker. Once all of the speakers have been detected ($n-1$ speakers), S_0 will represent a unique speaker as the last speaker, the nth speaker.

2. A new speaker can be added from the segments currently labelled as S_0, which represents the speakers which are yet to be detected. Using a 3 second region of S_0, the new speaker model is trained so that the likelihood ratio between model S_0 and a UBM (Reynolds et al., 2000) is maximised. The initial region's length must be sufficient to create a robust speaker model as it contains only one speaker. The strategy chooses the closest data to the speaker model and the 3 second region is selected empirically. A corresponding state is added to the previous HMM which is labelled S_x, where x is the number of iterations, and the transition probabilities are updated given a set of rules. The selected region is relabelled from S_0 to S_x in the segmentation.

3. Segments belonging to a new speaker S_x are detected and the data is reallocated between all the detected speakers. This is performed by adapting the speaker models to the current segments and then Viterbi decoding produces new segments. These two tasks, adaptation and decoding, are repeated until the segments do not change.

4. Lastly, the likelihood of the current solution is compared to the likelihood of the previous solution using the current HMM model as a stopping criterion. The previous solution is rescored with the current HMM where a non-emitting state is added. This makes the transition probabilities the same values for both HMMs. If there is no gain in likelihood or there is no more speech left to create a new speaker then the stopping criteria is met.

The DER can be minimised using two heuristics (Meignier et al., 2006). Firstly, the current speaker is removed if the total time in terms of segments is less than 4 seconds. Furthermore, the 3 second region which initialised the speaker is not used again in step 2. The segmentation continues from the previous iteration. Secondly, the DER puts higher importance on the large clusters. A rule to discard previous speakers from the segmentation if the length of their segments is shorter than the current one is enforceable. This rule leads to the longest speakers being detected first.

2.6.3 HDP-HMM

The "infinite HMM" method introduces a Hierarchical Dirichlet Process (HDP) on top of a HMM and is referred to as an HDP-HMM (Johnson and Willsky, 2010). This allows

for an infinite number of HMMs, or speakers. The HMM-based clustering approaches are fully non-parametric which means the observations dictate the number of parameters in the system. Systems can be converted into Bayesian and non-parametric systems using the Dirichlet process (Ferguson, 1973) and infinite Gaussian mixtures are produced by the Dirichlet process mixture model. The number of components is defined by a measure over the distribution. An improvement over classical methods is shown using Dirichlet process mixtures in (Valente and Wellekens, 2005b). This is extended by Teh et al. (2006) to a HDP which defines a prior distribution on transition matrices over infinite state spaces. Instead of assuming a fixed number of speakers, the prior measure is placed over distributions, referred to as a random measure, and is integrated out using likelihood-prior conjugacy. This results in a HDP-HMM (Johnson and Willsky, 2010), a data-driven learning algorithm. Posterior distributions are inferred over the number of states and the posterior uncertainty is integrated out when making predictions. This effectively averages over models of varying complexity. Similar results to the HMM-GMM method are seen by Fox et al. (2008) with the inclusion of the "sticky" parameter. This allows for more robust learning of smoothly varying dynamics. It helps to reduce the over-segmentation which results from creating redundant states and quickly switching between them.

2.6.4 Combination

Combining bottom-up and top-down approaches is carried out in an attempt to use the advantages from both in one system. Bozonnet et al. (2010b) apply the top-down approach as a base segmentation and the bottom-up is applied after with the aim of purifying the results. For three out of the four datasets tested, the combined system performed better than both the top-down and the bottom-up systems. Evans et al. (2012) combine the outputs of both the top-down and a bottom-up system. Each cluster, C_i, output from the top-down system is compared to the clusters from the bottom-up system. The bottom-up cluster C_n is matched to C_i if they share a sufficient proportion of frames or if C_n is the closest bottom-up cluster to C_i, where the inter-cluster distance is measured in terms of ICR. All the matched cluster pairs are accepted as valid speakers and are then retrained using only the frames detected in both C_i and C_n. Using both outputs acts to purify the models as only the best fitting data is used which implies better performance expected.

2.6.5 Discussion

Compared to the DHC approach, AHC tends to produce purer speaker models but it can be more sensitive to variation (Evans et al., 2012). Nevertheless, AHC is the most popular

approach for a speaker diarisation system (Ajot and Fiscus, 2009). Two thirds of the RT09 entrants used an AHC approach which consistently achieved the best results in the NIST RT evaluations. The IB method is compared to a BIC-based HMM-GMM method (Ajmera and Wooters, 2003) by Vijayasenan et al. (2009). The IB showed small improvements in the DER but performed significantly faster on meeting data. Vijayasenan et al. (2011) enhances the IB method with additional features leading to improvements seen in the DER when again compared to the BIC approach (Ajmera et al., 2004), with and without a realignment stage. For an i-vector approach, Shum et al. (2011) compared their method to the VB and the HMM-GMM with BIC baselines described by (Kenny et al., 2010). Considering the SE rates, the BIC system scored 3.5%, the VB scored 1.0% and the i-vector system scored 1.1%. When applying the mean-shift algorithm an improvement is seen over an HMM-GMM approach (Deléglise et al., 2005) on the ESTER-2 test set. It is noted that mean-shift algorithms tend to over estimate the speaker number (Senoussaoui et al., 2014). The ILP algorithm is compared to two methods by Rouvier and Meignier (2012) using ESTER-2 data (Galliano et al., 2009): an AHC i-vector approach and an HMM-GMM method with NCLR as opposed to BIC. The two i-vector approaches show improvement over the NCLR method, with the ILP approach showing the largest improvement. It is noted by Dupuy et al. (2012) that the i-vector ILP method is faster than the NCLR approach and implies it is more useful for larger datasets. Shum et al. (2013) compares their VB related method to the sticky HDP-HMM approach by Fox et al. (2011). Is it seen that the latter is more suitable when restricting the number of speakers detected however, the method tends to underestimate the number of speakers.

2.7 Scoring Metrics

The most common way to evaluate speaker diarisation is to use the DER. It was established in the early 2000s for the NIST RT evaluations and became the dominant metric in the field. Other metrics do exist which evaluate the segment boundaries from the SAD and speaker segmentation stages (Ajmera et al., 2004; van Vuuren et al., 2013) and the clusters created in the speaker clustering stage (Ajmera et al., 2002).

2.7.1 Diarisation Error Rate

Fiscus et al. (2006) describe the Diarisation Error Rate (DER)[1] which measures the amount of time not accurately assigned to speech, a specific speaker or nonspeech. This is the most

[1]DER scoring script: http://www.itl.nist.gov/iad/mig/tests/rt/2006-spring/code/md-eval-v21.pl

dominant evaluation metric and is calculated using the equation:

$$\text{DER} = \frac{\sum_{s=1}^{S} \text{d}(s)(max(N_{\text{ref}}(s), N_{\text{hyp}}(s)) - N_{\text{correct}}(s))}{\sum_{s=1}^{S} \text{d}(s) N_{\text{ref}}} \tag{2.26}$$

where S is the number of speaker segments in which the reference and the hypothesis contain the same speaker pair. The number of speakers in segment s in the reference is $N_{\text{ref}}(s)$ and $N_{\text{hyp}}(s)$ in the hypothesis, and $N_{\text{correct}}(s)$ represents the number of speakers in segment s which have been correctly matched between the reference and the hypothesis. The duration of a segment s is represented by $\text{d}(s)$. Miró (2006) describe the complete equations for each component, however, it is simply a combination of Missed Speech (MS), False Alarm (FA) and Speaker Error (SE) rates:

$$\text{DER} = \text{FA} + \text{MS} + \text{SE} \tag{2.27}$$

A collar parameter, or "forgiveness collar", adds a specified time in seconds to the segment boundaries allowing for leeway in the ground truth. However, it reduces the amount of total time scored in the evaluation. The standard and recommended selection is 0.25 seconds (Fiscus et al., 2006).

2.7.2 DP Cost and Boundary F-measure

Automatic speech segmentation research using a NN by van Vuuren et al. (2013) has provided a useful metric for evaluating boundaries. It uses dynamic programming to align the two sequences of boundary times (the reference and the hypothesis) and defines the absolute time difference between the two as the path cost. DPC is measured in milliseconds per reference boundary, and is calculated by:

$$\text{DPC} = \frac{pathcost}{N_{\text{ref}}} \tag{2.28}$$

where N_{ref} is the number of reference boundaries.

A Boundary F-measure (BNDF) can be calculated which gives a score involving the number of matched, inserted and deleted boundaries in terms of Precision (PRC) and Recall (RCL) (Ajmera et al., 2004). PRC refers to when a true boundary is matched and RCL refers to when a hypothesis boundary correctly corresponds to a boundary in the reference:

$$\text{PRC} = \frac{N_{\text{mat}}}{N_{\text{mat}} + N_{\text{ins}}}, \qquad \text{RCL} = \frac{N_{\text{mat}}}{N_{\text{mat}} + N_{\text{del}}}, \qquad \text{BNDF} = 2\frac{\text{PRC} * \text{RCL}}{\text{PRC} + \text{RCL}} \tag{2.29}$$

where N_{mat} is the number of reference boundaries which have matched to an equivalent hypothesis boundary. The number of hypothesis boundaries which are not matched to a reference boundary are the inserted boundaries, N_{ins}. Finally, the number of reference boundaries which have not been matched to an hypothesis boundary are the deleted boundaries, N_{del}. The F-measure is popular in the information retrieval field which measures the relevant documents that have been retrieved as an accuracy. It is a harmonic mean of the precision and recall. In terms of boundaries, precision normalises the number of matched boundaries by the reference number of boundaries and returns a fraction, or percentage. Recall normalises the number of matched boundaries by the hypothesis amount of boundaries.

2.7.3 Purity Measures

Purity measures are applied to speaker clustering in the form of cluster purity and speaker purity. Cluster purity describes how well a cluster is constrained to only one speaker and speaker purity describes how well a speaker is constrained to a single cluster. They are described by Ajmera et al. (2002) and assume the following relationships exist:

$$n_{i.} = \sum_{j=1}^{N_s} n_{ij}, \quad n_{.j} = \sum_{i=1}^{N_c} n_{ij}, \quad N = \sum_{i=1}^{N_c} n_{ij} \sum_{j=1}^{N_s} n_{ij} \tag{2.30}$$

where $n_{i.}$ is the number of segments in cluster i, $n_{.j}$ is the number of frames uttered by speaker j, n_{ij} is the number of frames in cluster i spoken by speaker j, N_c is the number of clusters, N_s is the number of speakers and, finally, N is the number of frames. Cluster purity, $p_{i.}$, of cluster i and the Average Cluster Purity (ACP) are:

$$p_{i.} = \sum_{j=1}^{N_s} \frac{n_{ij}^2}{n_{i.}^2}, \quad \text{ACP} = \frac{1}{N} \sum_{i=1}^{N_c} p_{i.} n_{i.} \tag{2.31}$$

Secondly, the speaker purity, $p_{.j}$, of speaker j and Average Speaker Purity (ASP), are:

$$p_{.j} = \sum_{i=1}^{N_c} \frac{n_{ij}^2}{n_{.j}^2}, \quad \text{ASP} = \frac{1}{N} \sum_{j=1}^{N_s} p_{.j} n_{.j} \tag{2.32}$$

An overall purity calculation combines both cluster and speaker purity measures:

$$\text{K} = \sqrt{\text{ACP} * \text{ASP}} \tag{2.33}$$

which is used as a method to evaluate different systems.

2.7.4 Discussion

The DER is a time-weighted metric which gives a percentage of time that has been incorrectly labelled, be it MS, FA or SE. The MS and FA rates roughly show how the segmentation has performed without penalising where the boundaries are wrong. The boundary measures are more appropriate in that respect, however, the BNDF has to be tweaked to consider whether the boundary is the start or end of a speech segment. Lastly, purity measures help to show how well the speakers and clusters have separated. Each metric has its disadvantage and none explicitly evaluate the segmentation quality. These disadvantages are investigated further in the following chapters.

2.8 NIST Rich Transcription Evaluations

NIST held Rich Transcription (RT) evaluations[1] until 2009. It encouraged research groups and companies to participate in different tasks to promote advances in speech technology. It covered tasks such as ASR, STT and Metadata Extraction (MDE). The task of diarisation is defined as an MDE task and began by focussing on BN and CTS data. In 2005 meeting data was introduced in the style of either conference or lecture room recorded meetings and 2007 saw the introduction of "coffee-break" recordings. For scoring the submissions it is assumed that small pauses within speech segments which are less than 0.3s are not considered segmentation breaks and are merged into the surrounding speech segment. This is known as smoothing. This minimum duration for a pause becoming an segment boundary was determined to be a good approximation. The submissions were evaluated using the DER with a collar of 0.25s. Specified portions of time were evaluated as opposed to the entire recording. This is referred to in this thesis as the NIST scoring setup.

The current state-of-the-art was a system designed by Wooters and Huijbregts (2007) at ICSI for RT07[2]. Many future systems have been based on this implementation (Anguera et al., 2005; Friedland et al., 2009; Huijbregts and van Leeuwen, 2012; Luque et al., 2007; Pardo et al., 2007; van Leeuwen and Konecný, 2007; Yella et al., 2014, etc.), and Figure 2.4 displays the implementation. Firstly, for the front-end acoustic processing, Wiener filtering is performed to remove any "corrupting" noise from the channels. Secondly, beamforming is applied to the separate channels to produce a single "enhanced" channel, namely MDM. MFCCs are extracted and a second stream of TDOA features are used for the MDM system only. The SAD is model-based and builds models for speech, silence and nonspeech on

[1] RT: http://www.itl.nist.gov/iad/mig/tests/rt/

[2] RT07: http://www.itl.nist.gov/iad/mig/tests/rt/2007/

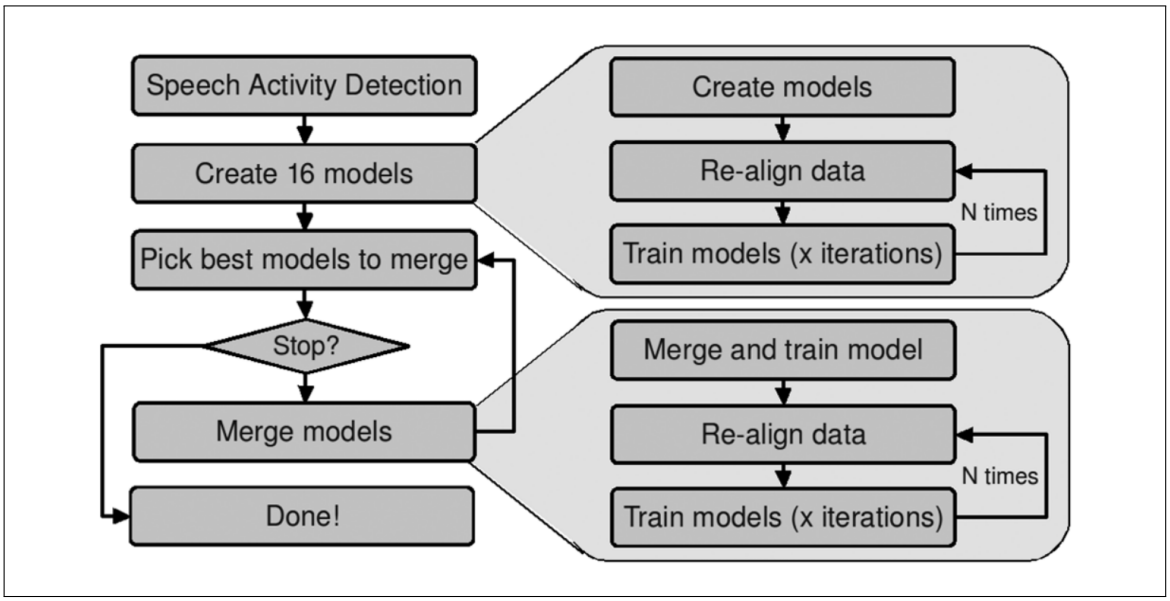

Fig. 2.4 The iterative process of the RT07 submission by ICSI is presented by Huijbregts et al. (2012). This systems remains to be the state-of-the-art today with Viterbi realignment and model retraining greatly improving the performance.

regions of high confidence for each. To calculate the confidence scores, a speech HMM and a silence GMM trained on BN data are applied. The silence is then split into two classes, one for regions with high energy and one for regions with low energy but high zero crossing rate. This results in models for speech, silence and nonspeech sound. The nonspeech sound class is then checked using BIC to see if any speech segments have been assigned to the nonspeech sound class. For speaker segmentation, linear splitting uniformly splits the audio into many clusters. Initial models are trained given these segment clusters. Iterations of Viterbi segmentation and training allow the models to be refined. For the clustering stage, iterations of pairwise merging and retraining the cluster models are performed. Viterbi decoding resegments the data from which the cluster models are retrained using the EM algorithm. The ΔBIC is applied as the clustering decision metric and stopping criterion. For the MDM system Wooters and Huijbregts achieved 8.5% and the SDM system achieved 21.7% DER.

In RT09[1], most of the submitted systems relied heavily on the ICSI-RT07 system design. ICSI themselves participated in RT09 with a system derived from their previous submission with added features including an audiovisual system, a multi-stream algorithm and a low-latency system designed for online diarisation (Friedland et al., 2012). Submissions from the

[1]RT09: http://www.itl.nist.gov/iad/mig/tests/rt/2009/

Universidad Politécnica de Madrid (UPM) (Pardo et al., 2012) and Universitat Politècnica de Catalunya (UPC) both applied the HMM-GMM approach with BIC. The UPC system determines the prior probability for a speaker talking after other participants have been speaking using a model. The UPM submission filters out feature vectors deemed not helpful for classification in a "frame purification" step. The Laboratoire d'Informatique de l'Université du Maine (LIUM) (Bozonnet et al., 2010a) presented the only submission which did not apply AHC. It performed DHC using the E-HMM approach, nevertheless it is similar to the rest in the fact that it applies the HMM-GMM topology for both segmentation and clustering as well as iteratively refining the models. The system with the lowest DER came from the Institute for Infocomm Research (IIR) and Nanyang Technological University (NTU) (Hieu, 2012; Nguyen et al., 2009). The system uses linear segmentation and the popular AHC clustering using two feature streams with Viterbi resegmentation, MFCCs and TDOA features. The SAD is model-based and trains speech and nonspeech models by looking at the energy and the zero crossing rate; it considers the MFCCs only. This is followed by bootstrap clustering within pair quantisation. This technique creates a histogram of the TDOA values for every microphone pair which has been selected. The peaks and centroids in the histograms are identified and other values are mapped to the nearest centroid. Using quantised TDOA the 9 centroids with the highest bin count are used as initial clusters and other bins are merged to the nearest centroid. This is known as smart initialisation. After merging, the centroids are given unique labels. Single-stream clustering uses a modified version of CLR as the distance metric between clusters and the T_s stopping criterion (Nguyen et al., 2008) whereas multi-stream clustering, which includes the delay features, introduces a weighting algorithm based on a distance metric.

The results for RT09 are presented by Ajot and Fiscus (2009). For MDM and SDM systems, the IIR-NTU submission achieved the best performance, 9.2% DER and 16.0% respectively (Nguyen et al., 2009). These results are including overlapping speech in the scoring, otherwise they greatly reduce to 3.8% and 10.7% respectively when ignoring overlap. The performance across the different systems proposed the final question of what is the "best performance you can get without solving overlap".

2.9 Summary

This chapter has presented an overview of the current technologies in the field of speaker diarisation. The goal of a system is to answer the question "who spoke when?" in an audio recording. Systems aim to produce the desired output of speaker labelled segments with timing information. A method can be split into four stages: feature processing, SAD,

speaker segmentation and speaker clustering. Step-by-step methods contain several distinct stages whereas integrated methods typically combine the speaker segmentation and speaker clustering steps into a single stage. The task is assumed to be unsupervised in regards to the clustering stage, in which no information on the number of speakers is provided to the system. However, model-based SAD is common which relies on pretrained speech and nonspeech models, making this stage semi-supervised. The NIST RT evaluations encourage many groups to submit different systems. RT07 resulted in the standard approach to diarisation still applied today, HMM-GMM-based AHC with BIC as the distance metric and stopping criterion, referred to as ICSI-RT07. The evaluations also gave the field the DER scoring metric.

Chapter 3

Data Analysis

Contents

Speaker diarisation systems come up against different difficulties in various ways. There are many public domain toolkits and several are designed for specific data types (Huijbregts, 2008; Meignier and Merlin, 2010; Vijayasenan and Valente, 2012). The domain of a dataset can be influential in the creation of a system. Ideally, a system would be robust to all datasets. This means a system should be adequate and usable for every dataset. However, this can be at the sacrifice of some performance when compared to systems tailored for a specific dataset. As well as the data domain being variable, the data itself has challenges too. Domain independent, crosstalk and overlapping speech have large negative effects on a speaker diarisation system.

An understanding of where the major errors come from and how they affect different stages in a system is necessary to produce a successful system (Huijbregts et al., 2012; Huijbregts and Wooters, 2007; Knox et al., 2012; Mirghafori and Wooters, 2006; Sinclair and King, 2013). This has been carried out in terms of DER, however there exists other metrics which give a different side to the story. The DPC and the BNDF investigates the accuracy of the hypothesised boundaries and the purity measures investigate the cluster and speaker labelling. The segmentation quality is considered, however there is not a metric that focuses directly on how well the system has performed in terms of detecting the correct segments.

The chapter is organised as follows. The challenges to the systems and properties of the data types are described in Section 3.1. Training and test sets are defined and described in terms of the meeting domain in Section 3.2 and broadcast media domain in Section 3.3. Next, three public domain toolkits using methods described in Chapter 2 are discussed in Section 3.4. Analysis of the toolkit results and an investigation into the segment and speaker labelling quality is carried out in Section 3.5. Lastly, the chapter is summarised in Section 3.6.

3.1 Challenges

The challenges for a successful diarisation system come from different areas. This section will discuss research that has investigated the challenges that face speaker diarisation, which system stages suffer the most and where the issues are most detrimental (Huijbregts et al., 2012; Huijbregts and Wooters, 2007; Knox et al., 2012; Mirghafori and Wooters, 2006; Sinclair and King, 2013). Further to this, the data domains are considered as each has different properties which affect the outcome in different ways. Lastly, two common negative data effects are discussed: overlap and crosstalk.

The data characteristics which caused different DERs were investigated by Mirghafori and Wooters (2006) and recordings were split into two types. Specific recordings which had

abnormally high DERs were called “nuts” and those which were overly sensitive towards tuning parameters (including number of initial clusters, number of Gaussians in each GMM and minimum duration of a cluster) were named “flakes”. BN data from the RT04 evaluation was investigated to determine which recording characteristics caused the two types. Features were extracted such as: speaker count features (males, females, ratios of each); conversation turn features (speaker changes per minute, total changes,); speaker duration features (everything labelled as one speaker, percentage of dominant speaker time); and show duration features (total duration, duration of scored regions). For “nuts”, the shows seemed to have many speakers, a large number of speaker changes and a high DER when there was only one dominant speaker labelled. For “flakes”, the correlation between high DER with a single speaker and the number of speakers was not strong, and the authors suggested this was because the recordings have fewer speakers.

Huijbregts and Wooters (2007) used their diarisation system, based on SHoUT (Huijbregts et al., 2009a) and related to ICSI-RT07 (Wooters and Huijbregts, 2007), to determine which system component contributed the most error to the final DER result. The system was described as robust to different audio conditions and data domains. This meant that tuning on external data is not necessary, however small parameters were noted which were apparently not sensitive to changes. These include number of initial clusters and number of Gaussians in each GMM. This conflicted with analysis by Mirghafori and Wooters (2006) who noticed recordings to be over sensitive to these parameters, the “flakes”. RT06s data was used and oracle experiments were carried out in which each component was replaced with the reference speaker labelled segmentation. The authors noted that they assumed each component was independent of the rest, which is not completely true as stages will impact others. In their case, for the RT06s meeting data given an ICSI-RT07 style system, the SAD stage produced the largest error at 22.1% of the DER. The research by Huijbregts and Wooters was extended by Huijbregts et al. (2012) who performed more oracle experiments using RT09 meeting data. The authors investigated how dependent one component was on another as opposed to assuming each was independent. Experiments showed that the larger errors came from the SAD component, the overlapping speech time and the merging component producing impure clusters.

Complementary to work by Huijbregts and Wooters (2007), Sinclair and King (2013) further investigated the challenges for speaker diarisation systems. They also used an ICSI-RT07 style system evaluated on a large meeting dataset, a combination of RT06/07/09 data. Oracle experiments were performed to isolate the effects of the various system components. Experiments investigated: the true number of speakers, an oracle SAD, ideal cluster initialisation, ideal models and overlap segmentation. Similar to Huijbregts and Wooters’s conclusions,

the SAD component was shown to be a large contributor to the overall DER performance. However, unlike the previous work, Sinclair and King realised the importance of building speaker models on pure data, i.e. coming from a single speaker. The authors concluded by suggesting that speaker models trained on reliably-identified pure data, even if it meant a reduction in amount of data, would greatly improve performance.

Instead of investigating system stages to determine which causes the most error, the quality of the segments was considered by Knox et al. (2012). Segments from five RT09 MDM systems were analysed to determine the types which were most detrimental to the performance (Bozonnet et al., 2010b; Friedland et al., 2012; Huijbregts, 2008; Nguyen et al., 2009; Vijayasenan et al., 2010). Two different segment types were investigated: segment duration (long/short) and speaker boundaries (distinguishing between the first segment after a change point, the last segment before a change point and the segments in between). Systems were evaluated on miss and speaker error rate, and the error contribution of the overlap. The authors showed that short segments cause more DER than long segments and segments before or after the change point gave worse performance than other segments. Furthermore, at least 40% of the error was detected within half a second of a speaker boundary.

3.1.1 Data Domains

There are four broad data domains in which speaker diarisation has been considered:

- **Conversational Telephone Speech (CTS):** Typically two speakers on separate channels.
- **Broadcast News (BN):** TV and radio recordings for news in a studio with live reports.
- **Meeting:** A limited number of people in a single room discussing a topic.
- **Broadcast Media:** TV and radio recordings of any format and genre.

Meeting data has become the most popular in the speaker diarisation field followed closely by the growing field of broadcast media. Meignier et al. (2006) discuss the main problems for each domain. The domains are discussed in terms of their typical format and the different challenges each presents to diarisation.

CTS data was the first type of data used for diarisation (Tranter and Reynolds, 2006). Microphones can be poor quality which leads to speech being unclear. Background noises are prevalent depending on the quality of the telephone line. Speech varies from informal conversations with friends to formal discussions with businesses, for example. In terms of the

number of speakers, it is usually restricted to just two speakers taking turns to talk, however overlap can be present.

BN data has been a stepping stone from CTS due to the lack of noise corrupting the speech and large amounts of data available (Miró et al., 2012). As this type of data is usually recorded in a studio setting, the quality of the microphones is high which leads to less background noise. The speech is scripted and well-spoken. It contains less overlapping speech and is more formal than a conversation. However, there may be instances of journalists recorded outside the studio where all sorts of background noise could occur. Furthermore, depending on the format of the BN programme, there might be adverts and music interspersed between news reports. There is likely to be a few speakers talking often and brief conversations with other speakers.

Meeting data is more challenging than BN data as it involves more discussions (Miró et al., 2012). This entails more overlapping speech at a faster pace. Meetings are less formal and more like a conversation than BN data as it is unscripted. Arguably, it's not completely natural speech as the participants are aware of being recorded. The microphones could be IHM, MDMs placed around the room, or an SDM usually in the middle of the room or table. The quality of the data recorded can vary depending on the type of microphones used and there may be an issue with crosstalk across channels if using IHMs. There could be many more speakers, again with a few who speak often and more who speak very little.

While transcription is the most common task in the evaluation of broadcast media systems, speaker diarisation has also been tackled in several challenges. The ESTER (Galliano et al., 2006) and REPERE (Galibert and Kahn, 2013) evaluation campaigns have used French BN data to develop diarisation systems, and Albayzin (Zelenák et al., 2012a) has used Spanish BN data. The MGB challenge, as part of its goal of improving spoken language technology for general broadcast media, has proposed the task of longitudinal speaker diarisation as one of its main components. Similar problems to BN data are seen, with background noises and music and varying audio quality from different recordings. However, the speech may not be scripted and can be spontaneous as in meeting data.

3.1.2 Data Properties

The data itself may contain elements of speech which lead to negative effects and additional challenges to a system. Two important data properties are discussed:

- **Overlap:** One or more speakers talking over another speaker.
- **Crosstalk:** Speech belonging to a different speaker detected on a speaker's IHM channel.

Overlap is widely accepted to be the largest cause of error in diarisation results (Huijbregts and Wooters, 2007; Knox et al., 2012) and overlapping speech can account for a substantial amount of the speech time in some datasets (Shriberg et al., 2001). Meeting data is seen as particularly prone to overlapping speech with participants discussing and talking over each other to make sure they have had their say. Clustering algorithms suffer when time that is in fact overlapping speech is labelled as a single speaker and models are trained on the inaccurate segments. Models have been trained on simulated overlapping speech which successfully detected simulated overlapping speech but did not generalise to real, natural overlap in meeting data (Otterson, 2008). Huijbregts et al. (2009b) trained overlapping speech models on speech surrounding speaker changes which was applied in the Viterbi decoding stage to detect overlapping speech segments. Acoustic features, such as MFCCs, energy, and spectral flatness, were determined to be useful for overlap detection from which accurate detection rates led to a reduction in DER (Boakye et al., 2011). Short-term spectral feature based overlap detection was improved by the use of TDOA features extracted from the cross-correlation of speech signals (Zelenák et al., 2012b). More recently, long-term conversational features such as silence and speaker changes have been extracted from long segments of time which again helped to reduce the DER (Yella and Bourlard, 2014). In RT09, described in Section 2.8, none of the submissions directly dealt with overlap and it concluded by posing the question what is the "best performance you can get without solving overlap?" (Ajot and Fiscus, 2009).

Crosstalk becomes an issue when considering the IHM channels. It can be caused by speakers sitting too close to each other, from breath and contact noise, and moving of heads to speak to another person as many people are untrained with using microphones. The SAD stage is already known to be the cause of the majority of the errors (Huijbregts et al., 2012; Huijbregts and Wooters, 2007) and detecting speech across IHM channels is troublesome when there is heavy crosstalk. Crosstalk features can be derived to help detect crosstalk and minimise the impact on a system. Wrigley et al. (2005) organised the speech in multichannel situations into four categories: local speech (the current speaker), crosstalk, crosstalk plus local speech and silence. Features are detailed from previous work and created by the authors to help detect crosstalk. Those described are: MFCCs, energy, Zero Crossing Rate (ZCR), Kurtosis, Fundamentalness, Spectral Autocorrelation Peak-Valley Ratio, Pitch Prediction, Genetic Programming features and Cross-Channel correlation. Two experiments were performed. Firstly, the best performing feature set for each channel classification was identified. For crosstalk it was determined to include mean cross-channel correlation and mean spherically normalised cross-channel correlation. For the local speech with crosstalk category, kurtosis and fundamentalness was additionally necessary. Secondly,

Table 3.1 Details of the AMI corpus, its large size is ideal for training models.

Dataset	Domain	#Files	#Segments	#Speakers	Time (hrs)
AMI	meeting	148	83811	182	101.6

these features were used for training an ergodic HMM for each meeting. This allowed for transition constraints between the four states to be dynamically applied. Performance improvements were seen but not for all classes. Further work combined MFCCs features with these auxiliary features for a speech/non-speech detector which led to an improvement in ASR performance (Dines et al., 2006).

3.2 Meeting Datasets

Meeting data has been used for a lot of speaker diarisation research in the last decade. The datasets can be grouped into two types: training sets and test sets. Training sets are large with many hours of speaker labelled data. This is useful for training models for speakers, speech and nonspeech. It is better for the training data to vary within the domain-type or recording setup to prevent models overfitting to the training data. Two large corpora are presented, the Augmented Multiparty Interaction (AMI) corpus (Carletta et al., 2005) and the ICSI corpus (Janin et al., 2003). Test sets are typically much smaller and do not contain recordings which appear in the training data. The test set presented is the evaluation set from the RT07 evaluation.

3.2.1 AMI: Augmented Multi-party Interaction project

The AMI Meeting Corpus[1] was created in 2006 as part of a 15-member European project, the main aim being to help improve group interaction (Carletta et al., 2005). About 100 hours of meetings were recorded in a combination of completely natural and uncontrolled situations, alongside meetings where participants are playing certain roles. A third of the meetings are natural and come from meetings which would have occurred whether they were recorded or not. As well as transcriptions for every meeting, more detailed annotations exist such as dialogue acts, topic segmentation, and head and hand gestures.

Each meeting is split into 4 sub-meetings and contains between 3 and 5 speakers. Meetings were recorded in different places: University of Edinburgh (U.K.), IDIAP (Switzerland), and the TNO Human Factors Research Institute (The Netherlands). The recording setup varied across the different recording institutions. As well as audio recordings, videos exists

[1] AMI Corpus: http://groups.inf.ed.ac.uk/ami/corpus/

Table 3.2 Details of the ICSI Meeting corpus which is ideal for training models given its large size.

Dataset	Domain	#Files	#Segments	#Speakers	Speech (hrs)
ICSI	meeting	70	102020	52	62.5

from various positions in the rooms. For all the setups, IHM channels exist for every speaker as well as SDMs for every meeting. Table 3.1 shows details of the training set which is used in this thesis.

3.2.2 ICSI: International Computer Science Institute corpus

The ICSI Corpus[1] of meeting data was created through the Meeting Recorder Project[2] in the years 2000-2002 along with other collaborators (Janin et al., 2003). The goal was to create a corpus from situations in which further speech recognition research would be useful. Automatically recorded, annotated and speaker labelled meeting transcripts are invaluable. Time is saved by humans not having to write notes which are sometimes vague and incomplete, and difficult to read later. The aim of the project was to create a portable device that records meetings and generates a searchable record.

Meeting rooms were equipped with a multichannel studio-quality recording system and a total of 70 meetings were recorded giving 62.5 hours worth of data. The audio was recorded using table-top microphones and close-talking channels depending on the number of participants in the meeting. Transcripts are available for each meeting giving words spoken as well as details of speech and nonspeech events. This is a second large dataset useful for training models and more details are seen in Table 3.2.

3.2.3 RT07: NIST Rich Transcription Evaluation in 2007

The NIST RT evaluations have been described in Section 2.8. Test sets were defined and systems are built by different groups and companies with the aim of achieving the best scores for certain tasks. These test sets are continued to be used after the evaluations have finished as a practical way to evaluate new systems, as results on the test sets exist for many different styles of systems.

For the RT07 evaluation[3], the speaker diarisation task data covered meetings which had been collected in three different situations. Firstly, conference room data was collected

[1]ICSI Corpus: http://www1.icsi.berkeley.edu/Speech/mr/
[2]Meeting Recorder Project: http://www1.icsi.berkeley.edu/Speech/mr/mtgrcdr.html
[3]RT07: http://www.itl.nist.gov/iad/mig/tests/rt/2007/

Table 3.3 Details of the RT07 evaluation set. It has been widely used after the evaluations allowing for comparisons across systems to be made.

Dataset	Domain	#Files	#Segments	#Speakers	Time (hrs)
RT07	meeting	8	11144	35	3.0

consisting of 180 minutes over 8 meetings. Two recordings from each of the following establishments: AMI, Carnegie Mellon University (CMU), NIST and Virginia Tech. Secondly, lecture room data was collected from 5 establishments totalling 160 minutes. Finally, a total of 40 minutes of coffee break meetings were recorded at the same five establishments as the lecture data. The main task was to perform diarisation on the MDM channels, and the secondary task was the SDM channels. The latter was more difficult than the former. The ICSI-RT07 system achieved 8.% DER on the MDMs and 21.7% DER on the SDMs. The SDM conference room data is used as a test set in this thesis and details are shown in Table 3.3.

3.3 Broadcast Media Datasets

The BBC archives consists of a multitude of different TV and radio shows. Different formats and genres contain different properties relating to the number of speakers, the amount of speech, the type of speech, the amount of background noise or music, and more. A dataset is first presented which consists of one TV and radio programme: “The Bottom Line”. The data is split into a train and test set. Next, data from the MGB challenge is presented which contains a training, development and evaluation set containing many different programmes covering various genres.

3.3.1 TBL: “The Bottom Line” programme

Provided by the BBC, “The Bottom Line” is a business and financial discussion TV and radio programme. There is always four speakers, a host and three different guests, some of whom feature in more than one episode. Although deemed to be broadcast media data, it is recorded in a meeting format. There are set ideas and a theme to discuss using free flowing speech, but all participants are aware that they are being recorded and will have prepared to a certain extent. It is recorded in a studio to be broadcast on both TV and radio, which means there is no background music or noises. Instead of having access to the final broadcast, the data supplied is the raw recordings from the studio.

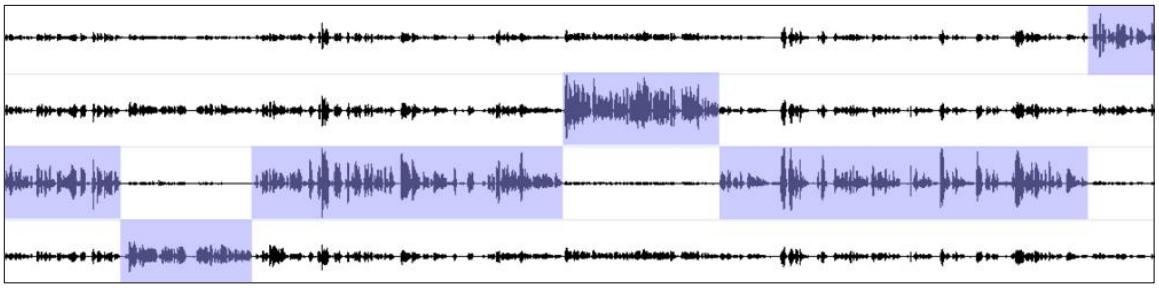

Fig. 3.1 Four IHM channels from a programme in the TBL dataset are displayed. The shaded segments represent speech on that channel. Crosstalk is seen across three channels but less is seen on the third, as this speaker sits opposite the other three.

The audio supplied is in the form of IHM and SDM channels. As is seen in Figure 3.1, crosstalk is noticeable across the channels. The microphone channels are not labelled with a speaker so these were manually determined by listening to each channel and viewing the waveforms. Transcript files are supplied and it is unknown how they were originally produced, but it seems to be manually transcribed. There is text which has been split into a new segment for every speaker turn and given a speaker label. This means for each word a speaker label is provided. However, there are two major issues with the transcripts. Firstly, as the audio is the raw edit, there are portions at the beginning and end of audio files containing non-programme related chatter and this is not included. Secondly, there is a lack of timing information. There are time stamps, but these only appear every 5 minutes, giving 5 minute chunks of different speakers talking.

Due to the issues in the transcripts, an accurate reference could not be produced by transcript alignment software. The reference was instead manually produced to an accuracy of 0.1 seconds. The speaker labels according to the transcript were used and segments were created according to pauses in speech, but the words were assigned to these segments. Diarisation is not evaluated by the words so for this task it was not needed.

The data consists of the raw edits of 26 episodes and each episode was recorded in several parts. The 26 episodes are split into a train set of 12 episodes and a test set of 10 episodes. It was discovered that 4 episodes could not be used, either because they were missing a transcript or a speaker channel. Table 3.4 depicts details of the two data sets. It is interesting to note that the percentage of segments on the third channel is much higher than the percentage of time. The third channel belongs to the host speaker, and although there are a couple of episodes which the male host is replaced by a female host, it is clear to see their dominant speaking style of pausing often while speaking, producing more segments, is shown. The TBLTRAIN and TBL datasets are used in experiments within this thesis and

Table 3.4 Additional statistics about the two TBL datasets. TBLTRAIN contains 12 programmes for training models and TBL contains 10 programmes for evaluation.

Information	TBLTRAIN	TBL
Programmes	12	10
Total time (hrs)	6.8	6.0
Instances of overlap	371	393
Boundary increase (speech to speaker)	18.0%	16.5%
Unique speakers (F and M)	8 and 30	7 and 25
Female speaker	19.7%	16.4%
Speaker channel speech %	24.0, 22.2, 25.5, 28.4	25.0, 24.2, 25.3, 25.6
Speaker channel segment %	23.7, 20.7, 30.1, 25.6	23.5, 24.1, 29.0, 23.3

Table 3.5 Details of the TBL datasets. The 22 usable programmes have been split into a training set of 12 programmes, TBLTRAIN, and a test set of 10 programmes, TBL.

Dataset	Domain	#Files	#Segments	#Speakers	Time (hrs)
TBLTRAIN	media	12	8295	48	6.8
TBL	media	10	8749	40	6.0

are formally defined in Table 3.5. Work does exist already using this data within a Master's thesis (Milner, 2012) but the results are not comparable as the manually transcribed reference was not used.

3.3.2 MGB: Multi-Genre Broadcast media evaluation

The MGB challenge was held in 2015 and open to both university and industry research groups (Bell et al., 2015). The challenge consisted of four tasks: multi-genre broadcast show transcription, lightly supervised alignment, longitudinal broadcast transcription and longitudinal speaker diarisation. The BBC supplied more than 1,500 hours of data over more than 2,000 TV programmes which were broadcast by the BBC during 6 weeks in April and May of 2008.

Task 4 covered longitudinal diarisation, which can be seen as an extension to speaker diarisation. Longitudinal diarisation is also known as speaker linking (van Leeuwen, 2010), speaker partitioning (Brümmer and de Villiers, 2010), cross-show diarisation (Tran et al., 2011; Yang et al., 2011) and large scale diarisation (Huijbregts and van Leeuwen, 2010) and refers to diarisation across a collection of connected audio recordings. For example, the recordings could be meetings held by a single group recorded over a few months or a TV series. Speaker linking aims to cluster across recordings to match speakers who appear in more than one recording. The commonly used method involves AHC without a model

retraining step and merges clusters by using the closest segment pairing distance as the score for the cluster pair (van Leeuwen, 2010; Vaquero et al., 2011). Alternatively, complete-linkage clustering works by taking the furthest distance in terms of segment pairings as the score for each cluster pair (Ghaemmaghami et al., 2012). Early work was carried out on two-speaker telephone conversations only but has since been extended to meetings (Ferras and Bourlard, 2012).

The system produced by the MINI group at the University of Sheffield consists of several stages: SAD, speaker segmentation and clustering, and speaker linking (Milner et al., 2015). The SAD is performed using DNNs trained to distinguish speech and nonspeech. Adaptation is then performed using an improved DNN output. The output is further improved by decoding using a novel duration-based language modelling approach for the speech and nonspeech states. Speaker segmentation and clustering is performed using a standard toolkit, which is unsupervised. Thus, it was suitable to use within this challenge. The second part is again adaptation using a pretrained DNN to classify or separate speakers, based on a novel approach of speaker clustering. Finally, the speaker linking stage uses BIC to test whether speakers with the largest amount of speaking time should be merged across shows.

The development data for the task was 19 shows covering 5 series broadcast by the BBC during June and July of 2008. For the training data no speaker labels were provided, and the time of speech segments was semi-automatically derived, in a lightly supervised training setting (Bell et al., 2015). The five series in the development set consisted of 3 episodes of a nature documentary show, 6 episodes of a political drama series, 2 episodes of a science fiction drama, 2 episodes of a sporting event and 6 episodes of a situation comedy series. These series had a large range of speakers, including re-occurring speakers and speakers confined to one programme. The date and time of broadcast for each show was provided as well as the series name. Diarisation across different episodes of the same series was restricted by allowing only episodes broadcast in previous dates to be used. Episodes from future dates were not allowed to affect the diarisation of the current episode. The evaluation data consisted of 19 episodes from two different programmes: a reality competition show and a documentary show. However, no speaker labelled training data was provided to train models.

The development and evaluation sets from the MGB challenge are used within this thesis and are defined in Table 3.6. SDM audio was provided, however speaker channels were not available in the challenge. Also defined in Table 3.6, is a training set. This training set is a collection of files which had been manually labelled for different speakers after the challenge took place. It was meant as a longitudinal diarisation set for the next MGB challenge which was originally intended to run in 2016 but has been postponed until 2017.

Table 3.6 Details of the MGB datasets. MGBDEV and MGBEVAL are the development set and evaluation set from the MGB Challenge in 2015. The MGBTRAIN data contains speaker labelled segments meant for use with the 2016 challenge which has been postponed. The MGBMINI is a smaller dataset of the earliest recording of each TV series contained in MGBDEV and MGBEVAL.

Dataset	Domain	#Files	#Segments	#Speakers	Time (hrs)
MGBTRAIN	media	17	12225	630	10.9
MGBDEV	media	19	9788	455	12.0
MGBEVAL	media	19	9236	569	15.0
MGBMINI	media	7	4484	179	5.2

3.4 Public Domain Toolkits

Public domain toolkits for diarisation exist freely to the community and many are tailored towards specific data types: ALIZE[1] (Bonastre et al., 2008), AudioSeg[2] (Gravier et al., 2010), CMUseg[3], DiarTk[4] (Vijayasenan and Valente, 2012), LIUM_SpkrDiarization[5] (Rouvier et al., 2013), SHoUT[6] (Huijbregts, 2008) and VB_Diarization[7] (Kenny et al., 2010). Three are described in detail: DiarTk, which uses the IB method (described in Section 2.6.1.2); LIUM_SpkrDiarization, a 6-step AHC approach; and SHoUT, which implements a model-based training regime similar to the ICSI-RT07 system (described in Section 2.8).

3.4.1 DiarTk

This toolkit was created by researchers at IDIAP using the IB principle and it was tailored for meeting data. Presented by Vijayasenan and Valente (2012), the method was initially formulated by Vijayasenan et al. (2009) based on MFCCs. This was extended to MFCCs and TDOA features in 2011 and finally extended to four feature streams in 2012. The toolkit takes up to four feature streams: MFCCs, TDOA features, modulation spectrogram features and FDLP features. It is AHC, same as the ICSI-RT07 system, however, the aIB method (described in Section 2.6.1.2) does not need to explicitly model speakers with GMMs which reduces complexity. Figure 3.2 shows the implementation. Four main aims were sought by the authors in creating this toolkit: code simplicity, the ability to handle any number of

[1] ALIZE: http://mistral.univ-avignon.fr/
[2] AudioSeg: https://gforge.inria.fr/projects/audioseg
[3] CMUSeg: https://www.nist.gov/itl/iad/mig/tools
[4] DiarTk: http://www.idiap.ch/scientific-research/resources/speaker-diarization-toolkit
[5] LIUM_SpkrDiarization: http://www-lium.univ-lemans.fr/diarization/doku.php/welcome
[6] SHoUT: http://shout-toolkit.sourceforge.net/
[7] VB_Diarization: http://speech.fit.vutbr.cz/software/vb-diarization-eigenvoice-and-hmm-priors

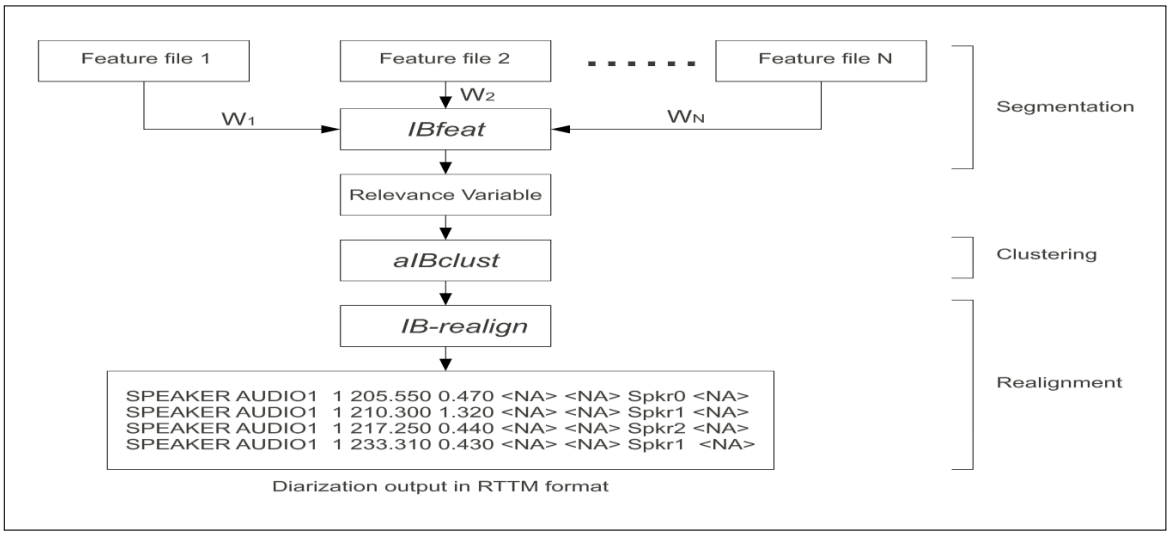

Fig. 3.2 The DiarTk toolkit is presented by Vijayasenan and Valente (2012) and consists of three main stages: segmentation, clustering and realignment.

varying feature streams, reduced computational complexity and to be able to reproduce state-of-the-art results on the NIST benchmark databases. The SAD aims to reject all nonspeech frames in the extracted features and is combined with a model-based speaker segmentation stage. The method estimates a background GMM for each feature stream and computes the relevance variables as a weighted sum of individual distributions. The approach uses AHC to cluster the relevance variables and a stopping criterion based on a threshold of Normalised Mutual Information (NMI) is enforced. The method ends with a Viterbi realignment loop to further refine the speaker boundaries detected in the segmentation stage and after the clustering stage.

3.4.2 LIUM_SpkDiarization

Based on MISTRAL_Seg from LIA, LIUM_SpkDiarization was created by LIUM and written in Java (Meignier and Merlin, 2010; Rouvier et al., 2013). Using Java removes the dependency problems with different operating systems and the toolkit is run directly without additional packages given the distributed JAR archive. It is designed for use with BN data. Figure 3.3 presents the system for both diarisation, single-show, and longitudinal diarisation, cross-show, and the authors were inspired by their winning RT04 system (Barras et al., 2006). The 6-step method begins by extracting MFCCs from the user provided audio. Secondly, the speaker segmentation initially detects the instantaneous change points using the GLR distance measure. Then BIC is calculated across the consecutive segments to fuse those

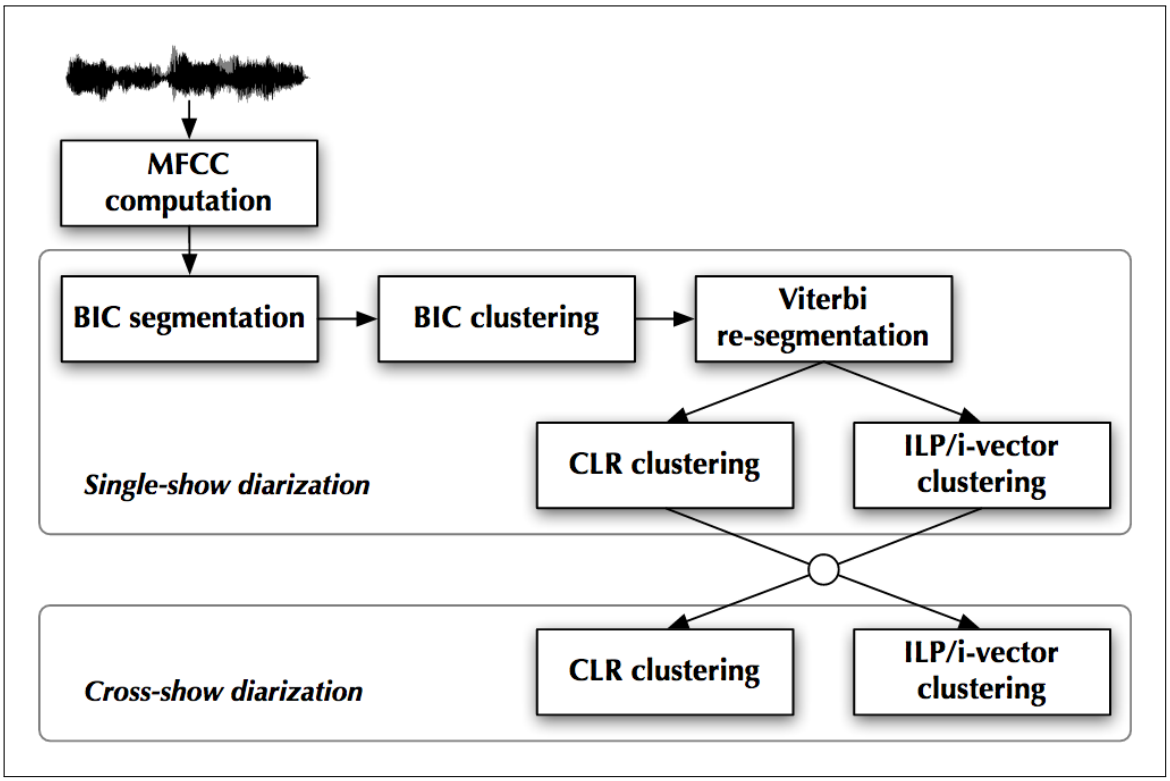

Fig. 3.3 Presented by Meignier and Merlin (2010) and more recently by Rouvier et al. (2013), the LIUM_SpkDiarization toolkit can perform diarisation (single-show) and longitudinal diarisation (cross-show).

which belong to the same speaker. Next, bottom-up clustering using a HMM-GMM model with BIC merges the closest clusters. Viterbi decoding is performed resegmenting the audio. The penultimate step performs speech/nonspeech segmentation using Viterbi decode aiming to remove music and jingles, commonly appearing in BN data. Lastly, a second clustering step is performed and two methods are presented. The first is another AHC step employing the NCLR distance measure as opposed to BIC to merge closest clusters. The second clusters acoustic i-vectors in an ILP approach described in Section 2.6.

3.4.3 SHoUT

SHoUT, an acronym for speech recognition research at the University of Twente, was part of a PhD project and written in C++. The approach is very similar to the ICSI-RT07 system described in Section 2.8, without a second GMM for delay-sum features (Anguera et al., 2006). It was designed to work well with meeting data and Figure 2.4 on page 38 shows how the 5-step algorithm is implemented. The first step performs SAD to filter out the nonspeech

Table 3.7 Overview of the training and test sets where two different reference segment annotations exist for the RT07 meeting data. The broadcast media test sets are TBL, the discussion programme, and MGBDEV and MGBEVAL which contain several episodes from different TV series.

Type	Name	Domain	#Files	#Segments	#Speakers	Time (hrs)
TRAIN	AMI	meeting	182	83811	182	101.6
	ICSI	meeting	70	102020	52	62.5
	TBLTRAIN	media	12	8295	48	6.8
	MGBTRAIN	media	17	12225	630	10.9
TEST	RT07 (NIST)	meeting	8	5425	35	3.0
	RT07 (SHEF)	meeting	8	11144	35	5.8
	TBL	media	10	8749	40	6.0
	MGBDEV	media	19	9788	455	12.0
	MGBEVAL	media	19	9236	569	15.0
	MGBMINI	media	7	4484	179	5.2

in the audio by using pretrained models built from BN data. Linear segmentation equally splits the speech segments and these are randomly divided into a number of bins. A GMM is trained for each bin, beginning with a single Gaussian before increasing to five by splitting the first with the biggest weight after each training iteration. This leads to one speaker fitting a GMM better than another so the Viterbi realignment assigns more speech from that speaker. Iterations of realignment and GMM retraining help to fit the dominant speaker on an GMM better. The third step merges the closest cluster pairs using the BIC score. This is followed by Viterbi realignment given the new merged models. These clustering and alignment steps are performed iteratively until the BIC stopping criterion is met.

3.5 Analysis

The analysis section considers two aspects of the speaker diarisation output: the speaker labels and the segments. Firstly, toolkits are applied which are performed without tuning parameters. Various evaluation metrics are considered along with the DER as they evaluate different aspects of the hypothesis. The datasets described previously are summarised in Table 3.7. There are four different evaluation sets which cover both meeting and media domain data and are used throughout the thesis.

Table 3.8 For each test set, the scoring setup is different. For RT07 (NIST) and the MGB datasets, the same scoring setup as applied in their respective evaluations is used. For RT07 (SHEF) and TBL, transcripts have been manually transcribed to a greater degree of accuracy allowing for stricter scoring.

Data	Reference	DER		
		Collar	Overlap	All time
RT07 (NIST)	provided by NIST	0.25	YES	NO
RT07 (SHEF)	manually transcribed	0.05	YES	YES
TBL	manually transcribed	0.05	YES	YES
MGBDEV, MGBEVAL	provided by MGB	0.25	NO	YES

3.5.1 Scoring

The DER is the standard metric for evaluating speaker diarisation systems and is described in Section 2.7.1. It has different settings which evaluate different aspects. The test sets described have different setups for scoring and these are seen in Table 3.8. A scoring setup requires three things: a set of reference segments, a metric, and the metric configuration. The DER metric is discussed. Two different scoring setups are seen for RT07. The NIST setup is the same as for the RT evaluations. However, the NIST evaluation only scores specific portions of time within the recordings, and not the entire recordings. The collar applied is 0.25 seconds. The second RT07 scoring setup is referred to as SHEF. The complete recordings were manually transcribed to an accuracy of 0.1 seconds[1]. This gives the alternative reference segments and requires different DER configuration. The full recordings are evaluated with a smaller collar of 0.05 seconds. Both NIST and SHEF scoring setups include overlap in the evaluation. The TBL dataset was also manually transcribed to the same accuracy allowing for the same stricter collar of 0.05 seconds to be applied. Again, overlap is included. Lastly, the references for the MGB datasets were provided by the MGB challenge which scored systems using a collar of 0.25 seconds. As opposed to the other setups, scoring is performed without overlap. This means portions of time where overlap exists is ignored. This removes some difficult areas from being evaluated.

The number of segments (#Segs) will be displayed alongside the amount normalised to the reference number of segments as a percentage (Seg%). If the system produces the exact number of expected segments, the result will be 100%. The same occurs for the amount of speakers detected (#Spk and Spk%). This chapter will display both, but following chapters will only display the percentages. Furthermore, with regards to the results in the tables, some metrics require 100% to be the goal and others required 0%. The DER and

[1]Manually transcribed RT07 test set: http://mini.dcs.shef.ac.uk/resources/dia-improvedrt07reference/

Table 3.9 Results for the public domain toolkits evaluated on both RT07 (NIST) and RT07 (SHEF). The DER is the sum of the missed speech (MS), false alarm (FA) and speaker error (SE) rate.

Data	Toolkit	MS%	FA%	SE%	DER%
RT07 (NIST)	DiarTK	3.7	15.2	19.8	38.8
	LIUM	4.9	13.9	21.3	40.1
	SHoUT	3.9	9.2	12.3	25.3
RT07 (SHEF)	DiarTK	7.1	36.7	24.9	68.7
	LIUM	8.6	34.7	23.2	66.4
	SHoUT	7.4	27.7	14.4	49.5

it's corresponding rates of MS, FA, and SE are errors meaning the goal is to minimise this towards 0%. However, the PRC, RCL, BNDF, ACP, ASP and finally the overall purity measure K are seen as accuracies so the goal is to reach 100% correct. The DPC is measured in time with the aim to achieve the lowest path cost.

3.5.2 Results

Initial investigations into the discussed public domain toolkits are performed with the DER metric. A deeper analysis considers the segmentation and boundaries followed by the speaker labelling in terms of purity measures. Lastly, the effects of overlap and crosstalk are investigated across the domains.

3.5.2.1 Analysis: Toolkits

The software described in Section 3.4 are public domain toolkits each applying a different technique to perform speaker diarisation. They are applied to the test sets defined to investigate their performance across the two domains. DiarTK uses the IB method and is tailored for meeting data, LIUM_SpkDiarization uses a AHC method desgiend for BN data and finally SHoUT is most similar to the ICSI-RT07 system (described in Section 2.8) which is also tailored for meetings.

The two scoring setups for RT07 are displayed as NIST and SHEF in Table 3.9. For NIST scoring, DiarTK and SHoUT outperform LIUM_SpkDiarization as they are designed specifically for meeting data. However, this is not the case for SHEF scoring, in which DiarTK has the highest error rate. SHoUT gives the best results except for the MS rate, which DiarTK achieves a lower error in both scoring setups. The SHEF setup is stricter than the NIST setup as it scores using a smaller collar and evaluates the whole recording instead of specified times. This is reflected in the results, in which SHoUT achieves twice as high

Table 3.10 Results for the public domain toolkits evaluated on TBL.

Data	System	MS%	FA%	SE%	DER%
TBL	DiarTK	3.8	14.2	14.5	32.5
	LIUM	3.8	14.2	9.8	27.8
	SHoUT	4.0	13.1	8.7	25.8

Table 3.11 Results for the public domain toolkits evaluated on both MGBDEV and MGBEVAL.

Data	Toolkit	MS%	FA%	SE%	DER%
MGBDEV	DiarTK	0.0	35.9	50.3	86.2
	LIUM	13.9	9.8	32.6	56.4
	SHoUT	9.3	8.1	36.5	53.9
MGBEVAL	DiarTK	0.0	46.1	56.2	102.3
	LIUM	8.2	19.2	45.7	73.1
	SHoUT	7.7	13.6	38.3	59.5

DER when scoring using the SHEF setup. This setup aims to evaluate the true performance by not hiding or obscuring difficult regions of time. It is clear that the NIST scoring does obscure the result.

The TBL results for the three toolkits are seen in Table 3.10. Again, the SHoUT toolkit achieves the lowest errors, except for the MS rate. The IB toolkit DiarTK yields the highest overall DER but the same MS and FA rates as the LIUM_SpkDiarization toolkit. An almost 5% absolute reduction in SE causes LIUM_SpkDiarization to outperform DiarTK.

Finally, the MGBDEV and MGBEVAL results for the three toolkits are displayed in Table 3.11. The scores are generally poorer when compared to the RT07 and TBL results, however the challenge itself has a baseline of 46.9% DER for MGBDEV and 47.1% for MGBEVAL (Bell et al., 2015). Again, the order of the toolkit performance is DiarTK, LIUM_SpkDiarization then SHoUT with the lowest error. DiarTk gives over 100% DER for MGBEVAL due to the high FA being detected. This is not seen in the other two toolkits. The SE is the highest error rate for all results due to the high number of expected speakers, 455 and 569 across MGBDEV and MGBEVAL respectively. This is different to the meeting datasets in which there are fewer speakers.

The following analysis is carried out using the SHoUT toolkit which is consistently better across the datasets. The DER metric does not give information on the segmentation and the speakers directly, it must be inferred from the MS and FA, and the SE. Deeper investigation into the segmentation and speaker labelling is performed, including the negative effect of the crosstalk and overlapping speech.

Table 3.12 The segmentation performance for SHoUT is evaluated using the Dynamic Programming Cost (DPC), measured in ms, and the boundary F-measure (BNDF), which is a combination of Precision (PRC) and Recall (RCL). The number of segments detected, #Seg, is compared to the percentage detected, Seg%. The DPC is measured in milliseconds.

Data	#Seg	Seg%	DPC	PRC%	RCL%	BNDF%
RT07 (SHEF)	8420	75.6	0.8	38.8	24.5	30.0
TBL	7499	85.7	0.7	31.9	22.5	26.3
MGBDEV	13801	141.0	2.7	45.1	54.5	48.9
MGBEVAL	15396	166.7	5.8	38.8	53.7	44.7

3.5.2.2 Analysis: Segmentation

To evaluate the boundaries, the DPC and BNDF are applied, along with considering the number of segments detected. The DER gives an idea of the MS and FA rates but this does not penalise for over- and under-segmenting the data. Table 3.12 displays the results for these additional metrics. The RT07 (NIST) results are not included as the metric is not set up to apply to specific regions of a recording only, thus would result in distorted results. For the meeting data, SHoUT under-segments both RT07 and TBL whereas it over-segments the MGB data. This leads to the meeting data having a higher PRC rate and a lower DPC. The broadcast media data has a higher DPC with a higher RCL rate, and a higher overall BNDF scores over the meeting data. This shows how under- and over-segmentation affects the results. However, these results show that the MGB data have a better segmentation, or boundary distribution, than the meeting data despite the DER showing the meeting data has lower error rates. This again is a miss representation by the DER of the underlying impact of the boundaries. The boundary measures do not address the segmentation in terms of how well matched a detected segment is to a reference segment.

Knox et al. (2012) saw that high DER is caused by short segments. Figure 3.4 displays the segment duration histogram for each reference. Across the references it is seen that the many segments are of a short duration. RT07 (NIST) reference contains 5424 segments and RT07 (SHEF) contains twice as many at 11144 segments. There is a similar amount of segments of duration less than 1s as segments longer than 1s for both references. For the TBL dataset, there are many short segments but it is clear than there are more segments longer than 1s. The longest segment is 19.4 seconds. The two MGB datasets contain many more longer segments, with both plots mostly to the right of 0. MGBDEV contains more shorter segments than MGBEVAL. This is despite having a similar number of reference segments, 9788 and 9236 respectively. All of the test sets contain many short segments, and therefore many boundaries, which can disrupt the DER and contribute to around 40% of the

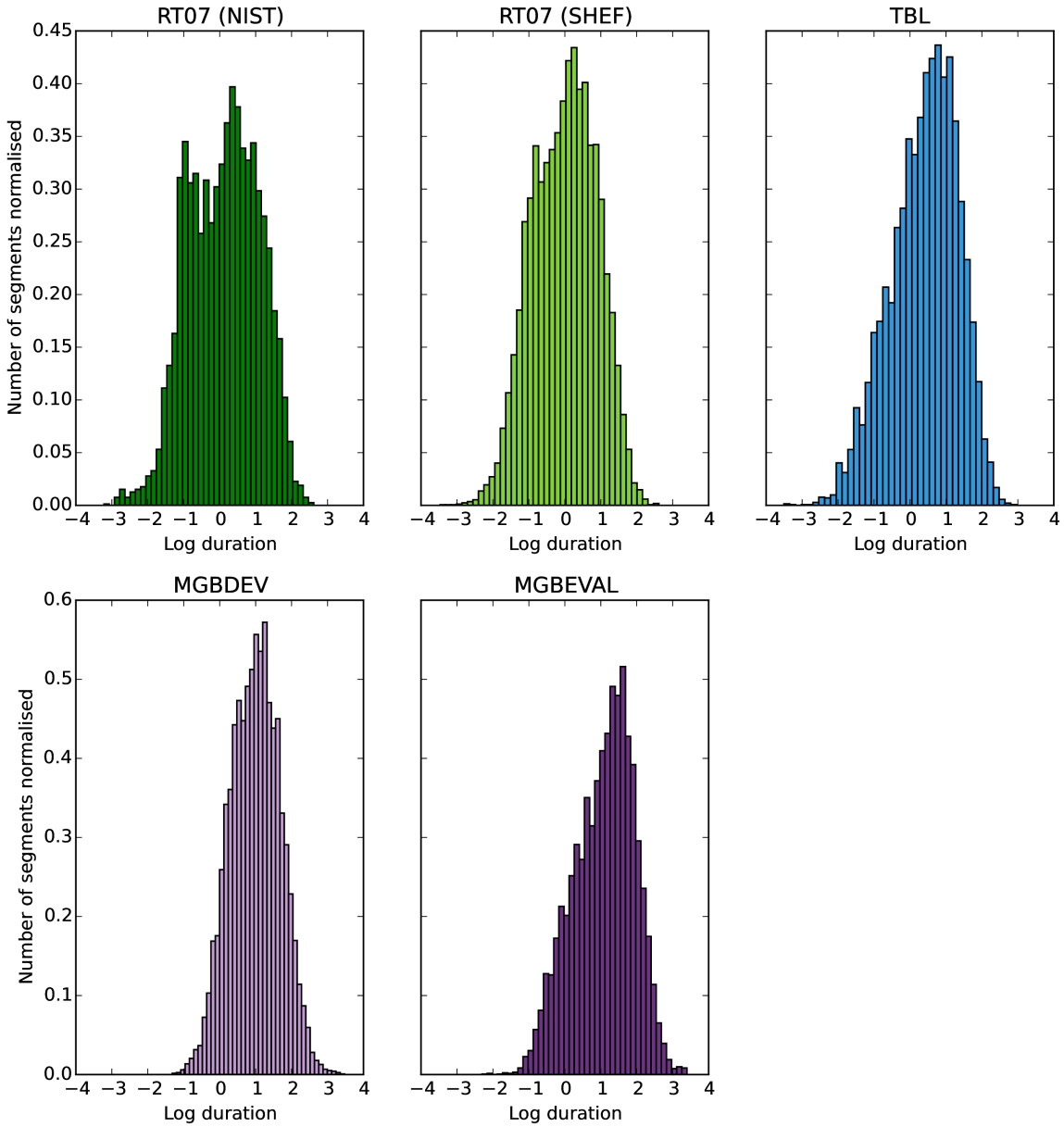

Fig. 3.4 Normalised logplot of histograms for the segment durations. The RT07 references are in green, TBL is blue and MGB references in purple.

error (Knox et al., 2012). For the meeting datasets, 50% of segments have a duration less than 2s and for the MGB datasets it is longer at 4s.

Speaker diarisation methods can make mistakes with detecting segments. Segments might be inaccurate due to detecting long or short segments and detecting too many or too few pauses within the segments. To mimic this behaviour of algorithms, the reference is

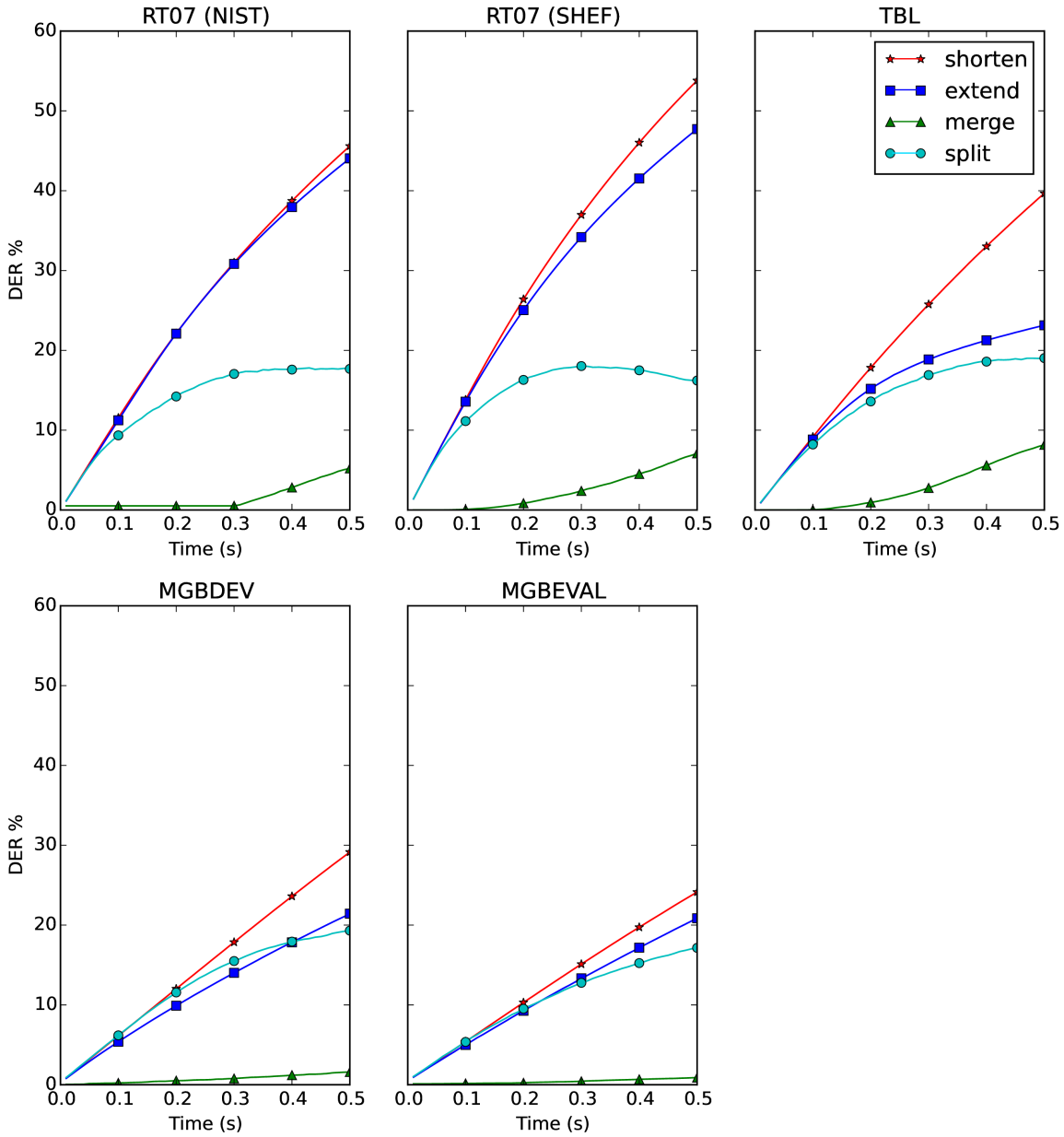

Fig. 3.5 The reference segmentation is perturbed and plotted to see how each perturbation affects the DER. The segments are extended, shortened, merged and randomly split into two parts given a specified time.

perturbed and scored against itself. This shows how the different variations of the reference affect the DER scoring. There are four defined methods of changing the reference:

- **Shorten:** every segment is reduced at the beginning and end by a specified time.
- **Extend:** every segment is extended at the beginning and end by a specified time.

- **Merge:** segments with the same speaker label are merged if the gap between the two is within a specified time.
- **Split:** segments are randomly split into two parts with a gap of a specified time.

Shortening and extending segments represents the difficulties in determining the true boundaries, and merging and splitting segments mimics under and over-clustering. The complete references are used and a collar is not applied as it is reference scoring against a perturbed reference. The boundaries may be moved and the collar would obscure the result. Figure 3.5 displays the outcome. For the first 0.1 seconds of time considered, shortening, extending and merging segments have a similarly bad effect across the datasets. When a larger time is considered, the biggest impact on the results is when the reference segments are shortened. This is followed by extending segments. Splitting segments also has a large negative effect for broadcast media but less so for the meeting domain. Merging has less effect as it is less likely to happen. This shows that merging, or under-segmenting the data, is not as detrimental to the results as splitting, or over-segmenting, when considering the DER.

3.5.2.3 Analysis: Speakers

The speaker analysis investigates the speakers in the recordings. The clustering stage aims to cluster together similar segments assuming the similarity is the speaker. Speaker error is calculated using the DER metric, however an alternative is to consider the speaker and cluster purities, as described in Section 2.7.3. Table 3.13 displays the purity results where SHoUT is run with default settings. The purity results for the meeting domain test sets are higher than the broadcast media domain. However the MGB test sets show more balance between the ASP and the ACP results. It is not consistent across the results or domains that either ASP or ACP outperforms the other. This shows the purity measures to be independent of the domain. It can be seen that the meeting test sets achieve an amount of speakers closer to the reference than the MGB test sets, and this is a reason for the higher purity scores. The results improve with a better guess of the expected number of speakers. Furthermore, the meeting test sets both under-cluster whereas the MGB datasets over-cluster. The last column depicts the SE from the DER metric. It is seen that the SE and the K are not consistent for the RT07 results. For RT07 (NIST), the K is worse than RT07 (SHEF) but the SE is better. These are calculated in different ways but are both time-weighted. This shows that one of these may not be as accurate as the other.

The SHoUT toolkit has a parameter for selecting the maximum number of clusters at initialisation. This has been varied and the outputs are scored in terms of ACP and ASP and displayed in Figure 3.6. This shows how the two purity measures trade-off. Higher

Table 3.13 The speaker clustering performance for SHoUT is evaluated using the purity metrics, Average Speaker Purity (ASP), Average Cluster Purity (ACP) and a overall purity measure K. The number of speakers detected, #Spk, is compared to the percentage detected, Spk%. These metrics are compared with Speaker Error (SE) and the reference purity, REF K%, which is the best possible result when the reference is scored against itself.

Data	#Spk	Spk%	SE%	ASP%	ACP%	K%	REF K%
RT07 (NIST)	41	117.1	12.3	71.6	65.3	68.4	70.1
RT07 (SHEF)	41	117.1	14.4	74.1	68.7	71.3	74.2
TBL	69	172.5	8.7	78.5	88.0	83.2	85.4
MGBDEV	319	70.1	36.5	53.4	56.0	54.7	84.4
MGBEVAL	438	77.0	38.3	54.6	53.8	54.2	94.8

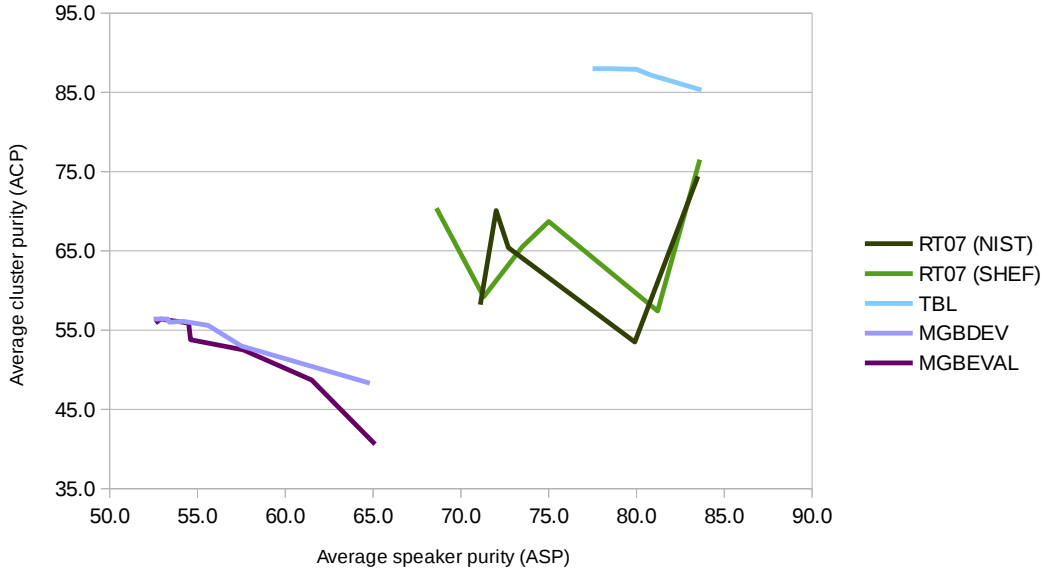

Fig. 3.6 The parameter in SHoUT for selecting the number of clusters at initialisation is varied to see how the purity measures vary when plotted against each other.

performance can be achieved over using the default settings for SHoUT. It is clear to see that the MGBDEV and MGBEVAL datasets behave very similarly, despite containing different programmes. For RT07, the two scoring setups also behave similarly. However, changing the number of initial clusters does not give smooth results for either setup and erratic curves are seen. Results on the TBL dataset see the highest purity scores no matter the number of clusters and is closer to the RT07 results. This is because it is meeting format which has been recorded in a higher quality broadcast setup.

Next the percentage of time for each speaker is investigated as it is known that the DER prioritises speakers with more data (Miró et al., 2012). Figure 3.7 displays the average speaker time across the recordings with the largest first. It is clear that there is typically a

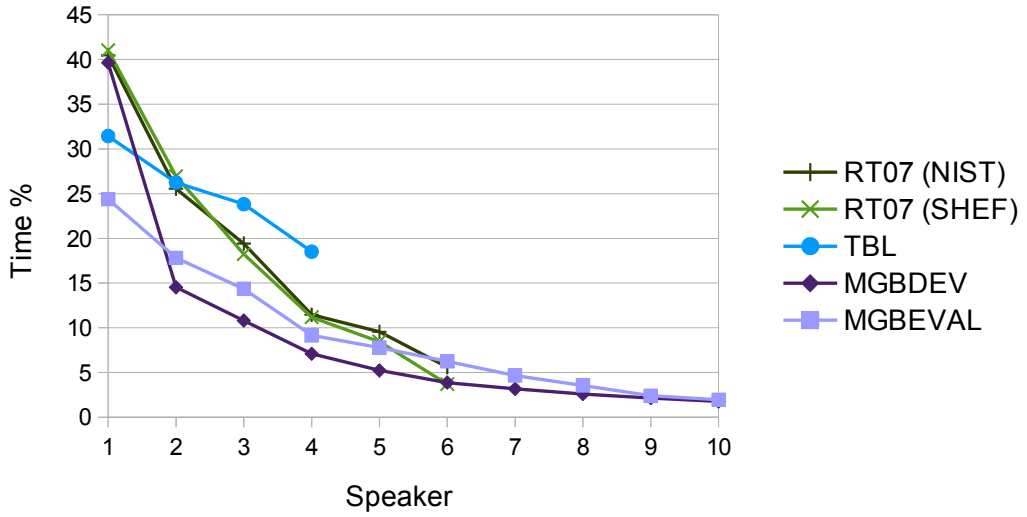

Fig. 3.7 The percentage of time spoken by each speaker across recordings and ranked in order of highest first. TBL has the smallest number of speakers at 4, whereas the number of speakers detected in the MGB datasets ranged from 10 to 62.

dominant speaker in a recording, across both domains. For meeting data like RT07, this could be the person leading the meeting or the host in terms of the TBL data. In TBL there is one host and three guests. It is assumed the three guests would have equal speaking time however this is not the case. For MGB there is a clear difference between the two datasets. MGBDEV recordings contain a single speaker talking for approximately 40% of the total speech time, with the second and following speakers at less than 15%. The MGBEVAL test set has a highest speaker at 25% with the following slowly reducing. The plot shows that it is not domain dependent having a single dominant speaker. As the DER prioritises speakers with more data, this could be miss representative of the true error in cases like the MGB data where many speakers are present.

3.5.2.4 Analysis: Overlap

As described in Section 3.1.2, overlapping speech is a problem for diarisation systems. It is investigated for the test sets using the toolkit SHoUT. The DER evaluates a system by either including overlap regions or ignoring overlap regions. Section 3.5.1 details the different scoring setups for the test sets, however, for every test set scoring with and without overlap regions are considered for this analysis. This shows how the overlap affects the different datasets and domains. Table 3.14 displays the DER results when evaluating in both ways. The last column includes the percentage of time in overlap for comparison. The results without overlap are consistently lower as the DER metric has removed the more difficult regions

Table 3.14 DER results when scoring with and without overlap regions of speech. Both methods are scored and compared to the percentage of time in which overlap occurs. The percentage of overlapping time is referred to as O%.

Data	With overlap				Without overlap				O%
	MS%	FA%	SE%	DER%	MS%	FA%	SE%	DER%	
RT07 (SHEF)	7.4	27.7	14.4	49.5	0.3	32.1	15.6	48.0	15.9
RT07 (NIST)	3.9	9.2	12.3	25.3	0.2	9.9	12.8	22.9	19.4
TBL	4.0	13.1	8.7	25.8	0.3	14.0	8.3	22.6	8.1
MGBDEV	13.1	7.4	34.7	55.1	9.3	8.1	36.5	53.9	10.9
MGBEVAL	8.3	13.4	38.0	59.7	7.7	13.6	38.2	59.5	2.8

of time from evaluation. This means time in the reference is removed from scoring which leads to a smaller amount of scored speech in the calculation of the DER. The denominator has changed which affects the results. In terms of the MS rate, this is reduced in every test set. In the meeting domain, the MS has become negligible at about 0.3% whereas the drop in error is less drastic for the broadcast media domain. Across both datasets, the FA has increased between 0.2% and 4.4% absolute. There is no clear difference between the two domains. The same is seen in the SE which increases in all datasets except for the TBL, in which there is a decrease of 0.4% absolute. The last column displays the percentage of time in overlap. Generally the meeting datasets have higher amounts of overlap than the MGB due to the format of the data. Meetings are unscripted and involve discussions which leads to participants talking over each other. Broadcast media data is more scripted and depending on the programme, overlapping speech is actively avoided. The TBL dataset, being a discussion TV and radio programme falls in both categories. It is not completely scripted however. The guests and host have prepared comments but discussions occur leading to the existence of overlap which is not removed from the programme. When considering the reduction in DER and the percentage of overlap, they are consistent. RT07 has the largest amount of overlap and the largest reduction in DER of 2.4%. This shows that a large amount of overlapping speech leads to a higher DER when scoring includes overlap. When there is overlap, it is a difficult region to detect the right amount of speakers as well as the correct speakers.

3.5.2.5 Analysis: Crosstalk

The RT07 and TBL test sets contain IHM channels, one for each speaker. This allows for SAD to be performed on each channel and it is assumed that performance would be improved over SDM channels as a single speaker is expected for each channel so no clustering is performed. Table 3.15 shows the results for both datasets when using the SAD part of the toolkit SHoUT only. The results are very poor, with over 100% DER caused by extremely

Table 3.15 Results on the IHM channels where heavy crosstalk noticeably affects the DER results, in particular the FA has risen above 100%.

Channels	Data	#Seg	Seg%	#Spk	Spk%	MS%	FA%	SE%	DER%
IHM	RT07	23173	208.0	35	100.0	16.1	103.4	0.0	119.47
	TBL	17044	194.8	40	100.0	55.1	104.9	0.0	159.98

high amounts of FA. As the DER is calculated considering the amount of time in the reference as the total time, if the hypothesis contains more than this then the overall DER can be higher than 100%. These results show, for both datasets, that too much speech is being detected across the IHM channels. This negative effect is caused by the heavy crosstalk. With the MS rates being so low, the majority of the channel time has been declared as speech. A second issue is the MS being detected. Even though too much speech is being detected, a large amount is still missed. This shows how the crosstalk has negatively affected the system.

3.5.3 Discussion

The analysis carried out has considered three public domain toolkits each applying a different technique. SHoUT, based on the ICSI-RT07 system (Wooters and Huijbregts, 2007), achieved the lowest scores across the datasets and the two domains. Each dataset is scored in a different way as described in Table 3.8, which includes two different scoring setups for RT07. The SHEF setup is stricter than the NIST setup. This helps to give a deeper understanding of the errors.

The segmentation analysis shows that different segment errors have a different impact on the results. However, the DER masks this as the MS and FA is weighted equally. There is no regard for the quality of the segmentation itself, as in how many segments were correctly detected. The speaker analysis shows that it is not domain dependent to be a single dominant speaker. However, the MGB datasets contain many more speakers with a small amount of speech time which the DER gives less preference too. The DER is the standard metric for evaluating speaker diarisation, however, it has been shown that it hides some errors and does not give detailed information on the segmentation. The boundary measures attempt this but give no consideration to how well matched the hypothesis and reference segments are. The purity measures consider the speaker labelling and ignores the segmentation completely.

In terms of the negative data effects discussed in Section 3.1.2, overlap has had a large effect on the results. It is seen that the datasets containing more overlapping speech have a higher error rate when compared to scoring without overlap, which ignores the overlapping regions. Crosstalk has a much larger impact on the results when performing diarisation on

the IHM channels. This effect is largely ignored when using SDM channels but if crosstalk exists, then the assumption that simply using the IHM channels will easily suffice is not true.

3.6 Summary

Speaker diarisation has been performed on many different domains over the years. Much of the past research has been on the meeting domain but more recently, the broadcast media domain has grown in popularity. Four training sets have been defined from both domains, however the larger training sets are meeting data. Meeting data has been more widely transcribed for speaker information though this may change in the coming years with the demand for more broadcast media data. Four test sets have also been defined covering both meeting and media data. Challenges to speaker diarisation come from the data itself. Overlapping speech and crosstalk can negatively affect the results and the system stages themselves can have difficulties handling this type of data. In terms of the segmentation, 40% of the DER occurs around the boundaries making over-segmentation an issue (Knox et al., 2012). However, the DER does not consider the segments themselves, rather the duration the segments cover. The analysis and data investigations have shown that the quality of the segments is largely ignored when scoring using the metrics discussed. This leads to the necessity of an alternative metric which evaluates the segments themselves and gives a score as to how well matched the hypothesis segments are to the reference segments.

Chapter 4

Segment-oriented Evaluation

Contents

The task of speaker diarisation was popularised by the NIST RT evaluations in which a metric was proposed to evaluate the submitted systems. The DER was established as the standard evaluation tool for the field. However, the DER has a few disadvantages which

causes it to behave in a less than ideal way. An arguably large collar around the reference boundaries is applied which removes time from evaluation due to not penalising an over- or under-segmented hypothesis. Furthermore, large clusters are prioritised leading to small clusters being ignored as they have less impact in the overall result. Most importantly, the number or quality of the segments detected does not feature in the metric. A fundamental part of speaker diarisation is to produce accurate segments. Other metrics have been proposed, such as boundary measures (Ajmera et al., 2004; van Vuuren et al., 2013) and purity measures (Ajmera et al., 2002), but these also have their disadvantages when evaluating segmentation.

When it comes to evaluating the segments, the current option is to simply state the detected number of segments alongside the DER results. Unfortunately this does not give a deep insight. It may show a similar or very different number of segments have been detected according to the reference, but no further information into the quality can be assumed. Alternatively, the DPC and BNDF measures consider the boundaries of the segments but not the segments themselves. To evaluate the diarisation performance, it is necessary to know how accurately the system has performed when compared to the reference. References are known to occasionally contain small human errors as it is difficult to precisely label boundaries. The DER combats this by using a collar around the reference boundaries. However, the reference provided is typically the only known true expected output and the user has no choice but to believe it and aim for a system that does well in comparison. If the reference contains many segments, then the system should be able to detect many segments. If the system under-segments the data, then the metric should penalise the system.

Speaker diarisation is used as a prerequisite for ASR (Saz et al., 2015). For this task it is vital to provide the ASR with accurately segmented speech. If a reference segment has been incorrectly split into many short segments prior to recognition, this could easily disrupt the system by not detecting all the words causing a higher deletion rate. It is valuable to know the quality of the segments output in the diarisation stage before passing them on to the next task.

The chapter is organised as follows. The disadvantages of the current evaluation metrics leading to the motivations are discussed in Section 4.1. An alternative evaluation metric for diarisation is presented in Section 4.2 which considers segmentation quality as the most important factor in a diarisation system. The segment-based metric is compared to the current metrics in Section 4.3 with results displayed in Section 4.3.3. Finally, Section 4.4 contains a summary of the chapter. This chapter investigates Objective 1, as discussed in Section 1.4, and is supported by a publication (Milner and Hain, 2016b).

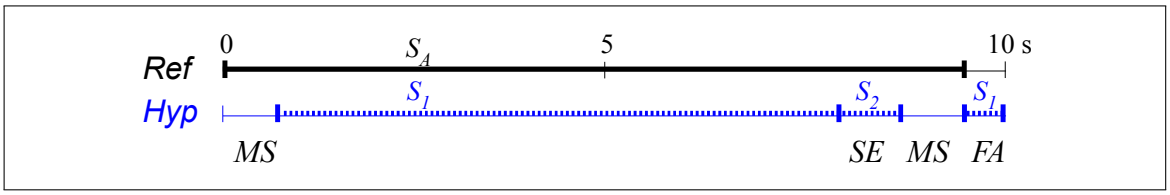

Fig. 4.1 Example of DER scoring, where MS, FA and SE are missed speech, false alarm and speaker error time segments. There is no hypothesised segment which represents the reference yet there is only small MS, FA and SE and thus low DER.

4.1 Disadvantages with current metrics

System evaluation is an important factor in determining how well a system performs compared to other systems. In an ideal world, every aspect of diarisation should be evaluated in a single scoring metric. These aspects include the segmentation quality, the speaker labels, the clusters detected, etc. However, this is not the case as is shown. The current evaluation metrics are: the DER (Miró, 2006), boundary measures (Ajmera et al., 2004; van Vuuren et al., 2013) and purity measures (Ajmera et al., 2002), which have been described in more detail in Section 2.7. Each metric focuses on particular aspects of the system and ignores others. The disadvantages of each are discussed.

4.1.1 Diarisation Error Rate

DER is the standard metric for speaker diarisation. It is the sum of three duration error values: MS, FA and SE (Miró, 2006). The first disadvantage is that the number of segments does not feature in the metric, which implies that either the introduction of short inter-segment gaps or the bridging of short gaps is hardly penalised. In Figure 4.1, multiple segments have been hypothesised for one reference segment, and, if reference speaker S_A is mapped to hypothesised speaker S_1, there is a segment with an incorrect speaker label. However, as the majority of the reference speech has been detected in the hypothesis and has the correct speaker mapped label, the DER will be a reasonable result. It measures time in error instead of error based on correctly detected speech segments (Pardo et al., 2012).

DER is a duration-based metric. This means an over-segmented output can still achieve a reasonably low DER, deceiving the user into thinking it is an acceptable speech/nonspeech segmentation close to the reference. This also implies 50 ms of speaker error is equivalent to 50 ms of missed speech. Arguably, missing a segment boundary is worse than a relatively short amount of speaker error. This implies a token-based metric is best which could be weighted to allow for greater penalisation of different and more important errors. Furthermore,

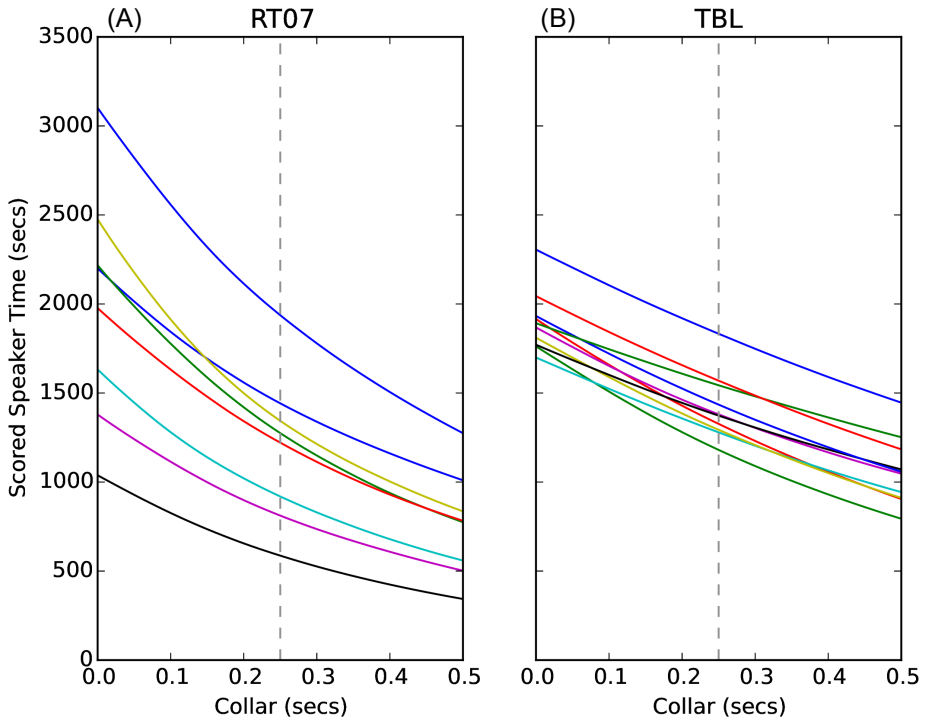

Fig. 4.2 For both meeting data, RT07 (SHEF), and broadcast media data, TBL, the SHoUT output for every recording in Section 3.5.2.1 is rescored with a varying collar for the DER. Increasing the collar leads to more regions of audio removed from the evaluated time. The dashed line represents the standard collar applied in the field.

there are occasions where the DER is over 100%. This happens when more speech is detected in the hypothesis than is contained in the reference, typically from very high false alarm rates as seen in Table 3.15.

In practice, a collar is added to either side of the reference segment boundaries. The collar compensates for uncertainty in human judgements in the reference as any error values inside the collar time are not counted. However, the standard value is ± 0.25 seconds around a segment boundary, thus allowing an hypothesis boundary to match a reference boundary within half a second. Assuming 3 words a second, this is more than one word which is a large amount of time, but is used throughout the field. The regions of time ignored when using a collar are removed from the overall scoring. Increasing the collar leads to more audio removed from evaluation, furthermore, more reference boundaries leads to more audio removed from evaluation. This can amount to around a third of the overall data at a collar of 0.25s. Figure 4.2 shows this effect by plotting the results in Tables 3.9 and 3.10 with a collar ranging from 0 to 0.5s. The recordings within the two datasets are displayed and it is clear

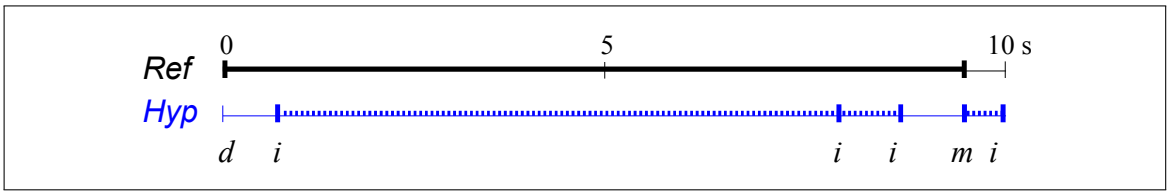

Fig. 4.3 Example of DPC and BNDF scoring, which shows the initial hypothesised boundary as a deletion, d, several inserted boundaries, i and an incorrectly matched reference end boundary, m.

that around half the amount of effective time is ignored from scoring given a collar of 0.25 seconds.

Speaker error is the amount of time labelled as the wrong speaker. For this to be calculated, the reference speaker labels must be mapped to the hypothesis cluster labels. The coinciding time between every possible pair of reference and hypothesis labels is considered and the pairs with the maximum are selected. This can give priority to large clusters and virtually ignore small clusters. This is not a problem generally for the DER as the large clusters have more impact in the overall score as they cover more time. However, the DER does not penalise detecting the incorrect number of speakers, except for indirectly within the SE.

Another point about the DER which may be considered as a weakness is that it does not allow for ambiguity in reference or hypothesis. For manual transcribers it is not completely clear where boundaries have to be placed. Decisions need to be lenient and allow for correctness ranges, for example in the form of confidence on boundary location. DER does not accommodate this and therefore leniency is expressed by deletion of data, in regards to the collar as described. Hence highly conversational speech becomes easier to detect although in fact exact segment times are harder to detect and overlap plays a big role.

4.1.2 DP Cost and Boundary F-measure

Ajmera et al. (2004) proposed applying the F-measure in the context of segment boundaries. The number of matched, inserted and deleted boundaries leads to calculating the PRC and RCL, which in turn provides the BNDF. Furthermore, an NN-based automatic speech segmentation paper (van Vuuren et al., 2013) has provided a useful metric. It uses dynamic programming to align the two sequences (the reference and the system output) of boundary times and defines the absolute time difference between the two as the path cost. The cost in seconds per reference boundary, DPC, can be calculated.

A problem with this boundary evaluation is that deletions and insertions are treated equally. Arguably in a speaker diarisation system it is worse to produce misses than false

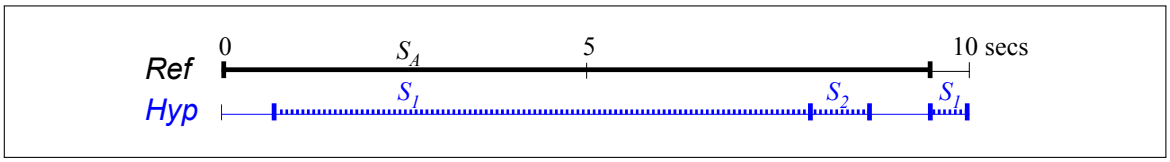

Fig. 4.4 Example of speaker and cluster purity scoring, which shows the single reference segment with no hypothesised speaker, nonspeech, at the beginning and end. The final hypothesised segment labelled S_1 also points to nonspeech. These regions are ignored in the metric.

alarms, as these are unrecoverable portions of speech. As for the DPC, the metric will give most information if the units to be assessed are of approximately equal length. However, for diarisation this is often not the case.

This method does penalise split segments in terms of increasing the number of insertions, but it does not consider what type of boundaries the matches are. For example, looking to the left and the right of the boundary, it could be NONSPEECH-SPEECH, SPEECH-NONSPEECH or SPEECH-SPEECH (different speakers, referred to as a speaker change). It detects the closest boundary in time, within a given window, without checking the type of boundary. Figure 4.3 shows an example of a match for the second reference boundary but it should be considered incorrect due to the types. However, the metric can be changed to penalise any matches which do not have the same type of boundary and this updated boundary F-measure is used for the metric comparison in this thesis.

4.1.3 Purity Measures

Purity measures are used for general clustering algorithms but can be applied to speaker clustering in the form of ACP and ASP (Ajmera et al., 2002). Cluster purity describes how a cluster is contained to only one speaker and speaker purity describes how well a speaker is constricted to only one cluster. The purity measures are duration-based and calculate how much a speaker, or cluster, is spread across clusters, or speakers. There is no consideration for the segmentation quality. This means the metric can not be used as a stand alone metric in evaluation a full diarisation system. It must be combined with another to evaluate the segmentation. For example, Figure 4.4 displays the speaker and cluster labels for a given segmentation. The reference speaker S_A has been hypothesised to be nonspeech at the beginning and the end. This is ignored when considering the speaker purity, and not flagged as missed speech as in the DER. The same happens for the cluster purity, as seen with the third hypothesised segment labelled S_1. It is not labelled as FA as in the DER, but simply ignored.

4.1.4 Motivations

There are fundamentally two aims for speaker diarisation: correct timing information and a correct speaker label. For an evaluation metric it is ideal to provide analysis for both aims. The timing information forms segments of speaker-pure speech. An issue arises from vagueness of what constitutes a segment. Typically, references are created by humans who will choose pauses where it is semantically meaningful. Therefore sentences, or "spurts" (Shriberg et al., 2001), are seen more as semantic units. This is because it makes no sense to listen to fragmented sentences of a speaker. Similarly, downstream applications such as translation or summarisation require semantically meaningful fragments. DER avoids that issue by using duration correctness rather than segmental correctness, allowing for completely fragmented output without any penalty.

This chapter focusses on an alternative speaker diarisation metric which does contain a penalty for an incorrectly segmented output. A metric similar to the BNDF is presented in Section 4.2 which attempts to evaluate the segmentation quality. Hypothesised segments are accepted as correct if they match a reference segment in time and in speaker label. This metric gives the user an insight into how closely matching their hypothesised output is to the reference. This assumes the system has detected nonspeech and speech in similar places and for similar durations as the reference.

4.2 Segment F-measure

An alternative metric is sought to evaluate segmentation quality more accurately than the current options. A speaker diarisation system aims to detect speaker-pure segments with cluster labels, so the segmentation is as important as the clusters. The F-measure score, as used for the BNDF, is calculated from the PRC and RCL rates. As it has been applied to boundaries, it can also be applied to segments themselves. This is referred to as the SEGF. PRC refers to missed or incorrectly hypothesised segments and RCL refers to hypothesised segments which do not have a corresponding reference segment. Using these metrics, it is necessary to determine which hypothesised segments match with a reference segment. An hypothesised segment is matched to a reference segment if its start and end boundaries are the same as the reference segment's start and end boundaries. It is known that errors in the boundaries exist in references, therefore a collar can be applied around the reference boundaries to allow for leniency when matching. The collar does not remove any time like in the DER.

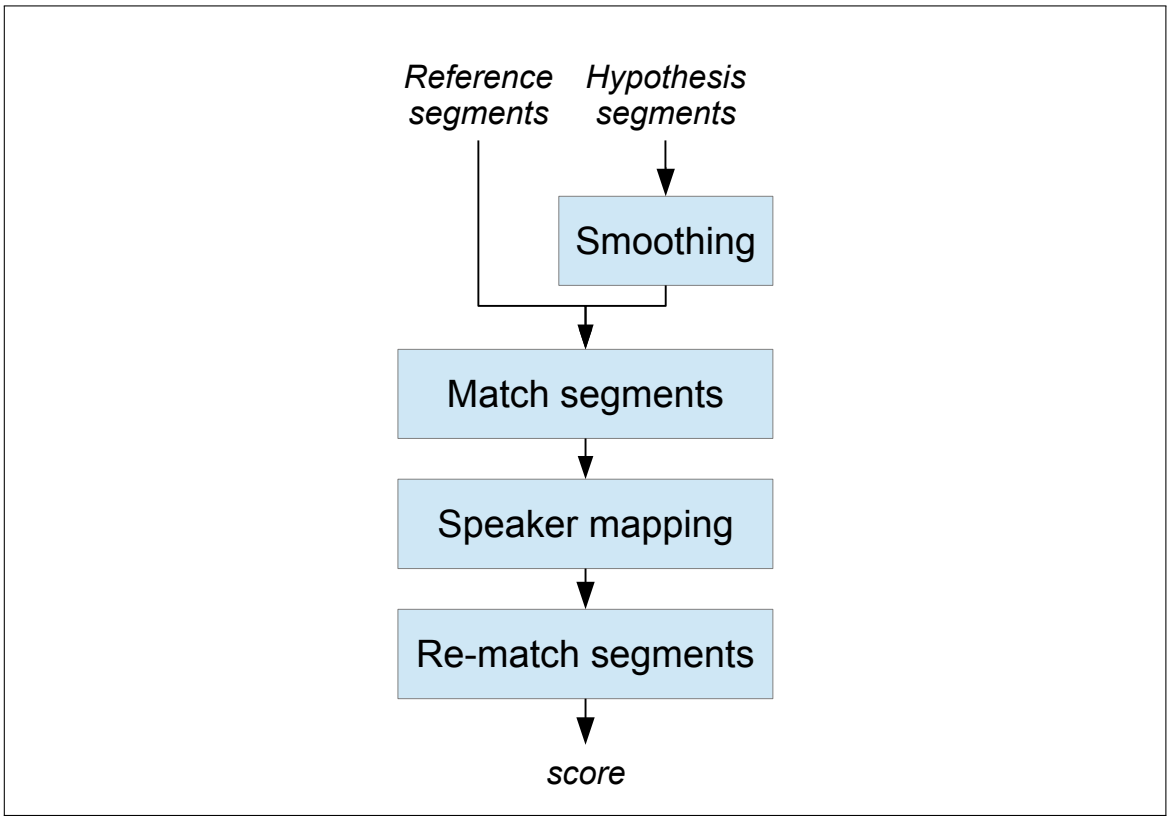

Fig. 4.5 The Segment F-measure (SEGF) algorithm requires a reference and an hypothesised segmentation and consists of four stages: smoothing, matching segments, speaker mapping and rematching segments. It outputs a score for the hypothesised segmentation in terms of correctly detected segments.

A speaker mapping method is required to determine a map between the reference speaker labels and the hypothesised cluster labels. Instead of a duration-based technique, the number of segments with matched boundaries is considered as this is a segment-based metric. The DER considers the amount of time between the speakers and the clusters, however, a full search improves this method by calculating a cost for every possible combination of pairs. The chosen combination is the set of speaker-cluster pairs which gives the lowest cost.

The stages within the scoring metric are shown in Figure 4.5. The first stage is discussed in Section 4.2.1 and considers smoothing hypothesis segments as in the NIST scoring setup. Section 4.2.2 details how the segments are matched together in time by considering a distribution around the reference boundaries. Thirdly, the speaker mapping algorithm is presented in Section 4.2.3. Lastly in Section 4.2.4, after a speaker mapping has been calculated, the segments are rematched in terms of time and speaker label. The final output is a score in terms of the F-measure of how well the hypothesis matches the reference.

4.2.1 Smoothing

The NIST RT evaluations argued for the hypothesis segments to be smoothed prior to evaluation using the DER, as described in Section 2.8. It was assumed that short pauses within the speech segments are not considered segment breaks. Consecutive speech segments in the hypothesis with the same label are merged if the gap is less than a specified time. The minimum duration of 0.3s for a segment break was deemed optimal. Smoothing is implemented as an optional step for a more accurate comparison to the NIST DER scoring. However, smoothing will have a different effect in the SEGF than in the DER. Arguably, if a hypothesis has output short pauses that are unwanted, this is an issue with the method applied.

4.2.2 Matching Segments

This step matches every hypothesis segment to a reference segment. Initially, this can only be done by considering the timing information. The speaker and cluster labels cannot be considered until after the speaker mapping stage. Matching in time implies seeking a reference segment with equivalent start and end boundaries. The hypothesised boundaries aim to match the reference boundaries. The reference boundary may not represent the real boundary due to human mistakes in transcription or mistakes in the transcript alignment. There may also be mistakes in the hypothesis boundaries. This means there are two types of uncertainty:

1. Uncertainty in the reference leading to a collar, c, around reference boundaries.
2. Uncertainty in the hypothesis leading to padding, w, around hypothesis boundaries.

This means that the true, correct boundary may not be the reference or the hypothesis boundary. This can be formalised as:

$$P(\text{correct}|\text{ref},\text{hyp}) \cong P[\text{ref} - c < \text{hyp} < \text{ref} + c] \tag{4.1}$$

where ref is the boundary in the reference, hyp is the boundary in the hypothesis and c is the collar representing uncertainty around the reference. In DER applying a collar around the reference boundaries, typically ± 0.25 s, is used to allow for reference errors. However, as previously mentioned, the time is removed from scoring which leads to a loss of scoring time (see Figure 4.2). For the SEGF, a collar is employed which does not remove this time from scoring. It defines a range around the reference boundary in which the hypothesis boundary is allowed to fall. For example, if the reference is known to have inaccuracies then

increasing the collar allows more leniency in the scoring. The collar can also be decreased if it is known that the reference was created to a high standard. The range value is an expression of reference uncertainty without loss in scoring power.

However, as both the ref and the hyp may contain uncertainty, the true, unknown boundaries are represented as r and h. Equation 4.1 is defined assuming independence:

$$P[r-c<h<r+c] = \int_{-\infty}^{\infty}\int_{r-c}^{r+c} P_{R,H}(r,h)drdh \tag{4.2}$$

$$= \int_{-\infty}^{\infty}\int_{r-c}^{r+c} P_R(r)P_H(h)drdh \tag{4.3}$$

$$= \int_{-\infty}^{\infty}\left[\int_{-\infty}^{r+c} P_R(r)P_H(h)drdh - \int_{-\infty}^{r-c} P_R(r)P_H(h)drdh\right] \tag{4.4}$$

$$= \int_{-\infty}^{\infty} P_R(r)\left[\int_{-\infty}^{r+c} P_H(h)dh - \int_{-\infty}^{r-c} P_H(h)dh\right]dr \tag{4.5}$$

$$= \int_{-\infty}^{\infty} P_R(r)\Big[F_H(r+c) - F_H(r-c)\Big]dr \tag{4.6}$$

where c is the reference collar, P_R and P_H are probability distributions of the reference and hypothesis, and F_R and F_H are functions for the reference and hypothesis. The collar is applied to the reference boundary times (on either side) allowing for the hypothesis boundaries to fall within this region. This is equivalent to the assumption that the actual boundary is represented by a uniform pdf of a certain width around the boundary. Consequently, one can estimate the probability of the hypothesis segment falling into a region using a distribution. The probabilities for start and end boundaries of a speech segment are multiplied and a threshold is applied to determine whether the segment matches or not. Distributions considered include uniform, triangular and Gaussian and are depicted in Figure 4.6. The first is a cumulative distribution function whereas the latter two are pdfs. There are seven cases of how a hypothesis boundary could fall around a reference boundary, given the collar being around the hypothesis too. The equations for each situation are detailed for the three different distributions in Appendix 1. Further, a padding variable can be applied which allows for the uncertainty in the hypothesis boundaries which introduces more leniency, however, this is not investigated.

4.2.3 Speaker Mapping

The DER scoring pairs reference speakers and hypothesised clusters based on duration, by mapping the speaker and label with the overall maximum time matched, until all reference speakers have an equivalent hypothesised label, if possible. For the proposed segment-based

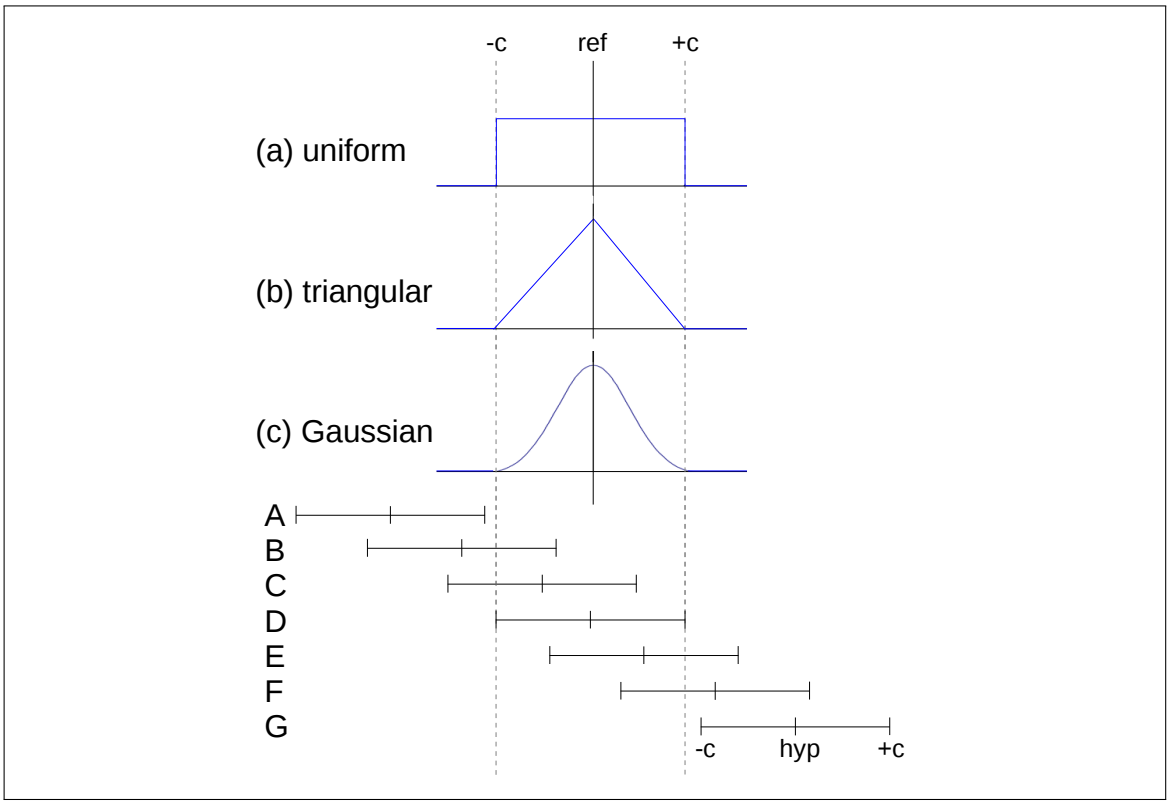

Fig. 4.6 There are seven cases (A-G) in which a segment boundary may fall around the reference when applying a collar. The distributions are (a) uniform, (b) triangular and (c) Gaussian.

metric, reference segments without matches are deemed errors in segmentation which implies the speaker labels for these segments are not reliable. Therefore, only the reference speakers and hypothesised clusters which contain at least one segment with matching start and end boundaries can be considered in the speaker mapping stage. This prevents speaker-cluster label mappings being contaminated with errors and incorrect matchings. However, a clear downside to this is when the segmentation quality is so poor that few matches are detected; this prevents the speaker mapping method from performing well.

The speaker mapping search example is shown in Equation 4.7. If reference speakers S_A and S_B exist and hypothesised speaker labels S_1 and S_2 exist, the probability, or score, that a reference speaker, S_A, is mapped to hypothesised label, S_1, given all the observations can be expanded:

$$P(rSpkr = S_A, hSpkr = S_1|O) = P(hSpkr = S_1|rSpkr = S_A, O)P(rSpkr = S_A|O) \quad (4.7)$$

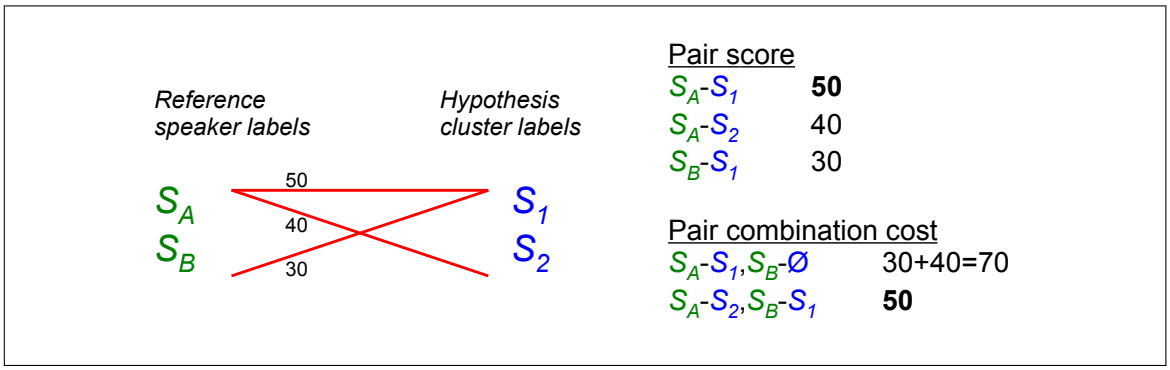

Fig. 4.7 Example of sub optimality of greedy speaker mapping. The optimal solution ($S_A - S_2$,$S_B - S_1$) gives the lowest cost as opposed to a higher score calculated for ($S_A - S_1$), which results in S_B and S_2 being unmatched.

Both parts can be represented in different ways, either by time (amount of coinciding time between matched segments) or the number of segments. As this F-measure is based on segment match, the number of segments is chosen for the speaker mapping score. As every segment is considered equal, there is no difference between an incorrect long segment and an incorrect short segment. However in time-weighted methods, an incorrect long segment will have a large impact on the error rate than an incorrect shorter segment.

Instead of the greedy search, a full search is implemented in order to determine the globally optimal mapping of reference to hypothesised speakers. Firstly, all possible matchings between reference speakers and hypothesised clusters and their scores are calculated. Next, for every pair with a score, the possible combinations of other pairs of speakers are sought and the scores of any ignored pairs are counted towards a cost for this combination of pairs. Finally, the combination of speaker and cluster label pairs which produce the lowest overall cost is chosen to be the correct speaker mapping. Figure 4.7 illustrates how this improves over methods based on greedy search. Considering the pairs depicted, the greedy method would select $S_A - S_1$ to be correct as it has the highest score, removing these two labels from further mappings meaning both S_B and S_2 would be unmatched labels (and thus both are an error). However, full search looks at all combinations and calculates a cost: where $S_A - S_1$ pairing would have a cost of 30+40=70 with two unmatched speakers, and the alternative would be $S_A - S_2$ with cost 50 and $S_B - S_1$ with cost 0, an overall cost of 50 + 0 = 50 making it the better combination.

4.2.4 Re-matching segments

With the speaker mapping, the previously matched segments are reconsidered. The initial matching determined pairs of reference and hypothesis segments given their start and end boundaries. It did not consider the speaker and cluster labels. In this stage, the segments are confirmed to match if the hypothesis cluster label corresponds to the reference speaker label in the speaker mapping. Those which do not, are counted as error.

4.2.5 Evaluation

The F-measure has been chosen as the method of evaluation as it allows detailed analysis of counting objects correct or not. It was used in the BNDF to match boundaries, as detailed in Section 2.7.2, and this method applies it in matching segments. The measure counts the insertions, deletions and matches. From these, PRC and RCL rates are calculated which provide the overall F-measure score. PRC refers to when reference segments are matched correctly and RCL refers to when hypothesis segments are matched correctly, and these give a percentage of accuracy:

$$\text{PRC} = \frac{N_{\text{mat}}}{N_{\text{mat}} + N_{\text{ins}}}, \qquad \text{RCL} = \frac{N_{\text{mat}}}{N_{\text{mat}} + N_{\text{del}}} \tag{4.8}$$

Lastly, the F-measure score is computed:

$$\text{SEGF} = 2\frac{\text{PRC} * \text{RCL}}{\text{PRC} + \text{RCL}} \tag{4.9}$$

This provides a weighted average of the PRC and RCL rates of segment matching.

4.3 Metric Comparison

The DER is the established scoring metric. However, the DPC, BNDF and K, the overall purity measure are useful for determining additional information about where the specific errors lie. The SEGF is investigated to see what extra information to the scoring the metric gives. The SEGF is compared to the other metrics and specifically to the DER. The effect of the collar is investigated in terms of SEGF and DER. Lastly, introducing leniency through different distributions around the reference boundaries is also investigated.

Table 4.1 Results comparing the different evaluation metrics to the Segment F-measure (SEGF). Scores have been calculated using outputs on RT07 (SHEF), TBL, MGBDEV and MGBEVAL from SHoUT. The DPC is measured in milliseconds.

Data	Seg%	Spk%	DER%	DPC	BNDF%	K%	SEGF%
RT07 (SHEF)	75.6	117.1	49.5	0.8	30.0	71.3	0.5
TBL	85.7	172.5	25.8	0.7	26.3	83.2	0.7
MGBDEV	141.0	70.1	53.9	2.7	48.9	54.7	1.5
MGBEVAL	166.7	77.0	59.5	5.8	44.7	54.2	0.8

4.3.1 Data

Datasets for both the meeting and broadcast media domains are considered and Table 3.7 displays the details. For RT07, the SHEF scoring setup is applied and referred to as RT07 for the rest of the chapter.

4.3.2 Setup

The systems used to produce the hypothesised segmentation outputs for scoring are from the SHoUT toolkit, and from Chapter 5 and Chapter 6. Both chapters contain semi-supervised speaker diarisation methods which use supplementary data. Five systems are presented in Chapter 5. The first uses transcript alignment and is only applicable to TBL data. Method 2 uses both SDM and IHM channels in a two-stage technique. The first stage is DNN-based SAD and the second calculates frame scores across IHM channels through alignment. Methods 3 and 4 use only IHM data by concatenating speaker channel features. The former expects a fixed number of speaker channels per recording which means it is not applicable to RT07 data. The final method uses SDM SAD segmentation, transcript speaker boundaries and labels, and finally frame posterior probabilities from the IHM channels. The method in Chapter 6 uses the best DNN-based SAD model from Method 2 in Chapter 5. The presented DNN-based clustering technique performs clustering and resegmentation with 20 fixed iterations.

4.3.3 Results

Experiments analyse the performance of the SEGF when compared to other metrics, such as the DER, DPC, BNDF and the overall purity measure, K. For the SEGF, a collar of 0.1 seconds is applied as minimal improvement is seen with higher collars as is shown later. Table 4.1 displays the results using the different metrics on the output from SHoUT. However,

Table 4.2 Results comparing the different evaluation metrics to the Segment F-measure (SEGF). Scores have been calculated using outputs on TBL and RT07 from methods presented in Chapter 5 and Chapter 6. The DPC is measured in milliseconds.

Data	System	Seg%	Spk%	DER%	DPC	BF%	K%	SEGF%
TBL	§5.2.1	38.5	100.0	21.7	3.0	43.1	86.4	1.5
	§5.3.1	99.5	100.0	12.5	0.6	71.0	81.0	54.3
	§5.3.2	76.6	100.0	8.1	0.6	73.6	89.5	52.4
	§5.3.3	85.4	100.0	9.2	0.7	69.1	89.3	42.2
	§5.4.1	94.5	100.0	10.7	0.5	74.0	84.4	57.3
	§6.2	124.2	92.5	22.8	0.6	68.3	77.9	36.6
RT07	§5.3.1	88.2	100.0	25.1	0.9	60.9	70.9	0.5
	§5.3.3	103.6	100.0	21.1	0.6	60.3	81.2	32.3
	§5.4.1	93.1	100.0	22.8	0.7	64.6	74.5	31.6
	§6.2	98.9	100.0	37.2	0.6	62.9	67.8	24.6

it is clear that the SEGF results are unusable for comparing to the other metrics. The results are less than 2% which seems to show the segmentation is poor for all datasets.

It is necessary to use outputs which have a good segmentation. Table 4.2 displays the results using the different metrics for the best performing methods in Chapter 5 and Chapter 6. In terms of DER, large variations are seen when compared to the SEGF. Two systems producing the same DER of 22.8% give SEGF results of 36.6% and 31.6%. This seems to show the DER is roughly similar to the SEGF. However, for a system with a similar DER of 21.7%, only 0.9% less, the SEGF score is 1.5%. This shows there is much poorer segmentation from this system but the DER masks this information. The DPC and BNDF are boundary evaluation methods. Across the datasets, a low DPC of 0.6 ms is detected from several methods. However, the equivalent SEGF vary from 24.6% to 54.3%. This shows the DPC also hides segmentation performance. For TBL, system §5.3.2 achieves 73.6% and system §5.4.1 achieves 74.0% BNDF whereas the SEGF scores for these are 54.3% and 57.3% respectively. This is a difference of 4.9% which again shows this measure is hiding the quality of the segmentation. Similar issues are also seen in terms of purity. System §5.2.1 performs worst on TBL according to the SEGF at 1.5%. However, K is high at 86.4%. The lowest number of segments is detected at 38.5%. This has caused the low SEGF as many hypothesis segments are not matched to reference segments given there are so few detected. This poor segmentation is picked up in the SEGF but completely ignored in the purity measure K. Lastly, it is clear to see that the SEGF scores do not give as high results as expected. System §5.3.2 from TBL has a DER of 8.1% and similarly good performance with the purity measure K and the DPC. The BNDF gives an indication there is a segmentation issue with the 73.6% result, however, the SEGF score is even lower, at 52.4%. Duration-wise

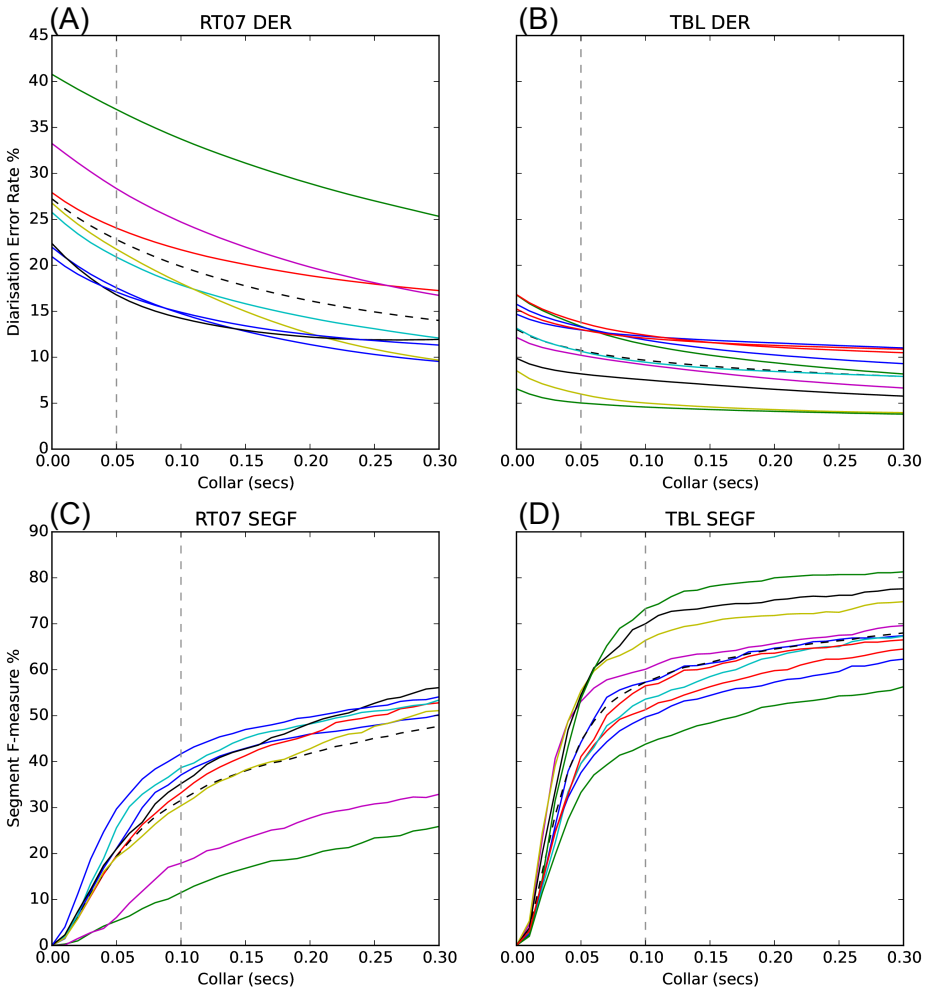

Fig. 4.8 The TBL and RT07 outputs from Method 5 in Chapter 5 are evaluated given the DER and the SEGF. As a collar is applicable to both methods, it has be varied from 0 to 0.3s. The black dotted line represents the average score for the dataset and the grey vertical dotted line shows where the collar falls in the defined scoring setup. The solid lines are the results for each recording within the specified dataset.

this method is considered good whereas there is much more room for improvement in the segmentation as the SEGF displays.

It has been noted in Figure 4.2 that as the collar is increased for the DER, the amount of effective scoring time is reduced. This impacts the results by skewing the true performance as more of the difficult segments of time are removed. A collar is also applied in the SEGF which increases leniency on the boundary matching, without removing any time from being scored. The collar is varied to investigate its impact across on the results. System §5.4.1 is considered for both TBL and RT07 datasets and the best output is scored using the DER and the SEGF. The collar is varied from 0 to 0.3 seconds and the results across every recording

Table 4.3 Results where recordings from RT07 and TBL have achieved the same SEGF performance. They are displayed alongside their DER performance for comparison.

File	Time(s)	PRC%	RCL%	SEGF%	MS%	FA%	SE%	DER%
RT07a	1221.16	52.9	47.4	50.0	4.8	9.6	3.6	18.0
TBLa	2144.56	52.8	47.4	50.0	6.2	1.4	6.7	14.2
RT07b	957.87	56.5	44.8	50.0	4.9	3.2	5.5	13.5
RT07c	626.77	55.4	45.7	50.0	11.2	0.5	0.3	12.0
RT07d	2319.45	47.0	53.4	50.0	8.8	1.8	1.2	11.9
TBLb	1072.65	61.2	42.3	50.0	2.8	0.7	7.8	11.3
TBLc	1055.79	61.1	42.3	50.0	2.8	0.7	7.8	11.2
RT07e	1263.09	56.8	44.6	50.0	6.0	2.3	1.8	10.1
TBLd	1902.26	56.5	44.9	50.0	5.0	0.6	1.3	6.9
TBLe	1180.45	63.9	41.1	50.0	3.3	0.6	2.4	6.3
TBLf	1137.97	65.2	40.6	50.0	3.2	1.0	0.7	4.9

within the datasets are plotted in Figure 4.8. Knowing the issue of the scored time with the DER, a smaller collar gives a more accurate representation of the errors. Plots (A) and (B) show the DER results for both datasets. The difference between the average DER at collar 0.05 seconds and at 0.25 seconds is $\sim$5% for RT07 and $\sim$1% for TBL. This is at a cost of scoring on less data. Plots (C) and (D) display the SEGF scores. Most of the improvement is seen within a collar of 0.1 seconds for both plots, though it is clearer in plot (D) than plot (C). A stability in the rank ordering of the recordings can be seen for the TBL SEGF results. After a collar of 0.05 seconds, the results increase slowly but do not drastically change. However, the DER shows a reordering even at a collar of 0.25 seconds. The rank ordering is important as the outcome of comparisons should not be affected by smoothing parameters. The collar has a large effect on the DER in terms of time lost and ranking order, the SEGF avoids the time lost and the reordering is minimal compared to the DER.

The next analysis looks at how the DER varies when the SEGF scored is fixed. Individual recordings from both RT07 and TBL were considered across different methods with different collars. Eleven were seen to have the same SEGF score at 50.0%. Results can be seen in Table 4.3 and have been ordered with the highest DER first. The components of both metrics are included for a deeper investigation into their effects on the performance. It is clear that a vast variation in DER scores exists when comparing the fixed SEGF to the DERs. The DERs range from 18.0% down to 4.9%. However, as they all achieve the same SEGF score, these results show that the DER is masking the true segmentation. The SEGF is a combination of PRC and RCL measures. There is a trend of higher PRC and lower RCL matching the lower DERs. There are exceptions to this however when looking at recordings RT07b, RT07e and TBLd. They have similar PRC and RCL with 13.5%, 10.1% and 6.9% DERs respectively.

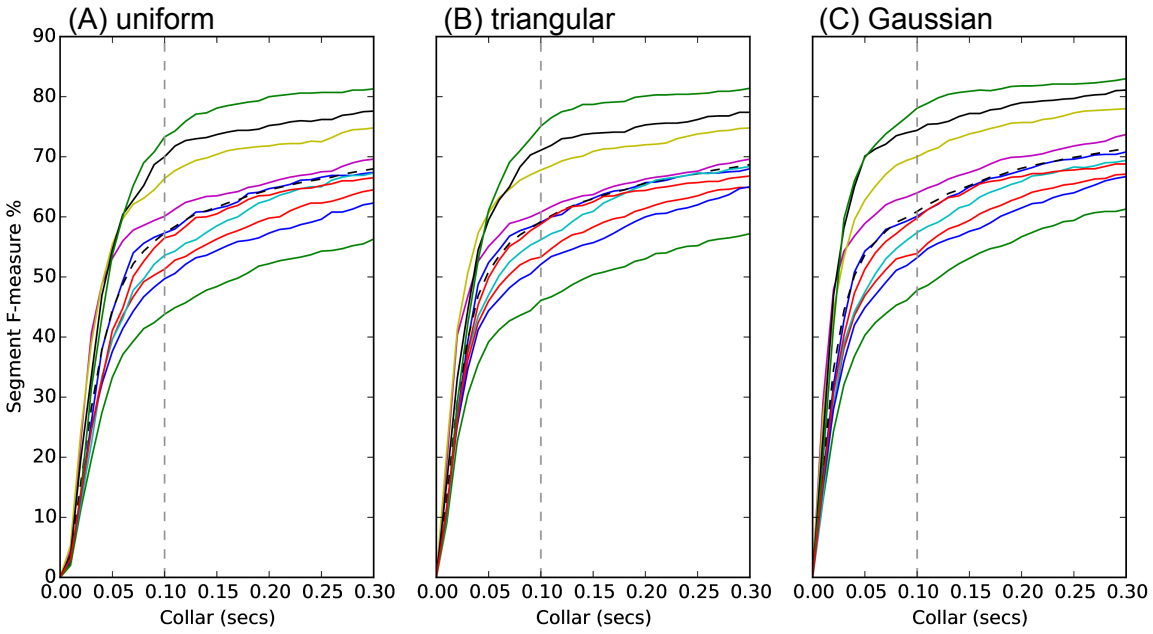

Fig. 4.9 The distribution around the reference boundaries for the SEGF is varied where (A) is uniform, (B) triangular and (C) Gaussian. Results are from the TBL data using Method 5 in Chapter 5 and Figure 4.8(D) is the same as (A).

There is no clear correlation between the MS, FA, SE and the PRC and RCL. Incidentally, these DERs are fairly good, as opposed to the 50.0% SEGF score. This reinforces that it is harder to achieve a high SEGF measure and segments are more difficult to match than time.

Lastly, the proposed metric allows for distributions to be imposed around the reference boundary when a collar is in place. This allows for a decision to be made on both start and end boundaries and how they fall within the collar. The previous results have used a uniform distribution. It is investigated how two distributions introduce leniency within the scoring: Triangular Segment F-measure (t-SEGF) and Gaussian Segment F-measure (g-SEGF). The results can be seen in Figure 4.9 for TBL data from system §5.4, where the collar is varied from 0 to 0.3 seconds. The three plots in the figure show the increase in scores across the recordings. For all the collars, the triangular and Gaussian distributions give small improvements. The Gaussian distribution improvements are higher. This has helped to add leniency in the segment matching process. Another advantage over using an alternative distribution over a uniform distribution is the change in performance when the collar is very small. Scores with small collars increase quicker. This allows for more useful scores when the collar is low. If many similar low scores exist, a difference between the segmentation quality is hard to see. Applying one of the alternative distributions helps to see this distinction at small collars. The SEGF scores in Table 4.2 are recalculated using a

triangular and a Gaussian distribution, both with a collar of 0.1 seconds. Across the methods and datasets, the triangular distribution increases the scores by an average of 2.1% absolute, and the Gaussian distribution increases the scores by an average of 4.6% absolute.

4.3.4 Discussion

The results show how the SEGF gives an clearer indication of the segmentation performance. The other metrics, DER, DPC, BNDF and K, provide acceptable performance results despite the poor segmentation that the SEGF indicates. It is harder to achieve good results with the proposed metric as segments are much harder to match than general regions of speech which the DER considers. In Table 4.2, the system with the lowest SEGF score also has the worst DPC and BNDF scores. This is because these measures are related in the fact they both consider boundaries. However, the purity score and the DER are not the lowest, the former being the third best purity score in the table.

A collar is proposed as in the DER which works in a similar way. However, the DER removes this time from evaluation whereas the SEGF keeps the time. The former may lose up to a third of the amount of scored time when using a collar of 0.25 seconds as seen in Figure 4.2. This masks the true performance by removing difficult portions of time from evaluation. The SEGF prevents this from happening, leading to a more accurate result. Considering the rank order of the recordings, changing the collar affects the order in the DER more than the SEGF measure. It is ideal for the rank order not to change given parameter changes so comparisons can be made.

Given a fixed SEGF result, the corresponding DERs greatly vary. This shows the SEGF and DER are not correlated. In Table 4.3, a slight trend is seen in which a high PRC entails a low DER but several exceptions to this rule are seen. The DER components, MS, FA and SE, show no correlations with the PRC and RCL of the SEGF. These results again show how the true segmentation performance is not picked up within the DER scoring.

As it is seen to be difficult to achieve similarly high scores as the other metrics, leniency within the boundary matching can be introduced. As well as the collar, different distributions around the reference boundaries, bounded by a collar, have been investigated. The t-SEGF and g-SEGF increase the scores and the Gaussian distribution gives larger improvements in results. These alternative metrics also show a larger distinction in the low collars allowing a difference to be seen between systems producing a poor segmentation.

There are several goals for a good metric to meet. The SEGF is shown to be sensitive. When the DERs gave the same results in Table 4.3, variations in the SEGF show further insight in the segmentation quality which the DER is not sensitive to. The rank ordering is also more stable fo the SEGF than the DER as seen in Figure 4.8. In terms of reproducibility,

there is no random element within the metric so evaluating a system output would always give the same result. The metric is quick to compute, a similar time to computing the DER. Lastly, the SEGF results give different information than the other described metrics. This allows for the user to see how usable the system is in terms of the segmentation quality.

4.4 Summary

The chapter has investigated the current method of evaluating speaker diarisation systems in terms of several metrics: DER, boundary measures and purity measures. When discussing the disadvantages to each, it was noted that none evaluated the segmentation quality. Detecting the correct segments of speech is fundamental to the task. It must be necessary to analyse how a system is performing in terms of its segmentation output, something which is lacking. This leads to the need for a new metric which fills the gap in the field by taking segments into account when scoring.

The proposed metric is based on the F-measure score. Instead of looking at boundaries as in the boundary F-measure, the segments themselves are evaluated. The F-measure requires detecting the number of hypothesis segments which have been matched correctly to reference segments. The unmatched reference segments are viewed as deletions and unmatched hypothesis segments are seen as insertions which provides PRC and RCL rates leading to an F-measure. A collar is introduced which compensates for reference boundary errors without removing time from the evaluation, unlike in the DER. Both start and end boundaries of segments are evaluated instead of just considering boundaries singularly like the boundary measures. A speaker mapping algorithm is proposed which considers a cost for each combination of pairs and selects the mapping giving the lowest cost.

Chapter 5

Speaker Diarisation with Auxiliary Information

Contents

Speaker diarisation is referred to as an unsupervised task in the literature (Miró et al., 2012; Tranter and Reynolds, 2006). This means no a priori information is known about the

Table 5.1 Auxiliary information can be organised by type, prior knowledge or supplementary data, and the kind of information it contains: acoustic, speaker, or timing.

Auxiliary Information	Prior Knowledge	Supplementary Data
Acoustic	room setup; recording setup; format/genre; domain	pretrained models for speech, non-speech models, noise, etc; IHMs
Speaker	number of expected speakers; identity; speech patterns; gender	pretrained speaker models
Timing	speaker boundaries; speech patterns; predicted amount of nonspeech	transcripts

audio. The input can vary, however the desired output is always the same, speaker labelled segments of time within the recording. Considering the input, an unsupervised system may use an SDM or MDMs. The RT evaluations, described in Section 2.8 evaluate systems for each type of channels separately. Despite the trend for an unsupervised method, systems exist which make the most of additional data and information as input. This leads to either a semi-supervised or supervised system, as discussed in Section 2.2. A fully supervised method uses information from the output to enhance the system. Moraru et al. (2004a) investigated ending the clustering once the reference number of speakers was detected. Semi-supervised systems use pretrained models, inferred knowledge or imperfect additional data to improve performance. The public domain toolkit SHoUT, as described in Section 3.4.3, uses models for speech and nonspeech trained on data from the BN domain to improve the SAD stage (Huijbregts, 2008).

Additional data and information is referred to as auxiliary information, as the information supplies extra help and support to a system. Auxiliary information can be grouped into two types: prior knowledge and supplementary data. Prior knowledge is seen as metadata which is additional information about the data. Supplementary data refers to physical data which is used by a diarisation system, such as additional channels or pretrained models. Information can be inferred from the single channel, the supplementary data or provided by the user. Furthermore, auxiliary information can be organised into three conceptual groups: timing information, acoustic information and speaker information. Table 5.1 sorts examples of the different auxiliary information into the these defined groups.

Acoustic information refers to information about the signal. In terms of prior knowledge, this can be about the room and recording setup. For example, the type of microphones, the number of microphones and the room layout. It is known that different quality of microphones or recording in rooms or outside can greatly disrupt ASR systems (Barker et al., 2015). It is also known that audio recorded from different domains have different traits, so tailoring

a system towards specific data types can improve performance, such as the public domain toolkits described in Section 3.4 (Huijbregts, 2008; Meignier and Merlin, 2010; Vijayasenan and Valente, 2012). For example, BN data is typically recorded in a higher quality setting, e.g. a studio, than meeting data, e.g. an office. In broadcast media, detecting the genre (Doulaty et al., 2016) infers the types of acoustic events that might exist. For example, a comedy show might contain laughter whereas this would be uncommon in a documentary. IHMs are commonly used as supplementary data in the case of beamforming (Anguera et al., 2007), as described in Section 2.3.2. This combines channels to produce a single channel with the aim of reducing the negative impacts from low quality channels. Participants of the RT07 and RT09 evaluations applied the beamforming technique (Bozonnet et al., 2010a; Friedland et al., 2012; Nguyen et al., 2009; Pardo et al., 2012; Wooters and Huijbregts, 2007). Pretrained models are also used as supplementary data. For example, speech and nonspeech models trained on specific domains (Huijbregts et al., 2012) and music models have been shown to improve performance in the SAD stage (Sinha et al., 2005).

Speaker information refers to information about speakers in the recording or speakers in general. Prior knowledge of the number of expected speakers is the most common form of auxiliary information for speaker diarisation. However, Moraru et al. (2004a) investigated this and it was not shown to improve performance. Speech patterns are the distinctive traits of how a speaker talks. For example, some people speak in long phrases and others add more pauses. This information helps the segmentation stage in how long or short the segments are expected to be. Also, the gender of the speakers can be detected to help improve SAD (Gauvain et al., 1998). Supplementary data comes in the form of pretrained speaker models. When trained on sufficient data the models can improve the clustering stage (Moraru et al., 2004a). Further work on speaker models exists in the field of speaker linking, which is an extension to diarisation (Brümmer and de Villiers, 2010; van Leeuwen, 2010). Many recordings are considered as opposed to one at a time, requiring speakers to be linked across recordings. Models trained to distinguish speakers, as opposed to specific speakers, are used for feature extraction (Yella and Stolcke, 2015; Yella et al., 2014).

Lastly, timing information relates to audio objects which have a given start and end time. For prior knowledge, these items could be speaker boundaries, or change points. These are points in the audio in which a different speaker begins to talk. Speech patterns are inferred from speaker boundary times as well as how a speaker behaves when talking before or after another speaker, which helps to predict the speaker labels. If the timing information of acoustic events is known then models are trainable on the data which leads to better detection of the events. Nonspeech timings can help to estimate the amount of nonspeech occurring in a recording. Much of this prior knowledge is inferred from the supplementary

data which can be in the form of transcripts. Transcripts are not typically involved in speaker diarisation as words are not the focus of the task. However, aligning transcripts produced speech and nonspeech segmentation with speaker labels (diarisation output) as speaker labels were present in the transcripts (Milner, 2012). Furthermore, speech patterns and speaker identities have been used to infer speaker labels as well as the current and previous or next speakers (Lamel et al., 2004).

The chapter is organised as follows. The relevant work in the literature followed by the motivation for semi-supervised methods is discussed in Section 5.1. Further details on timing information including a transcript alignment method for speaker diarisation is detailed in Section 5.2. This is followed by three methods in Section 5.3 based on acoustic information. A final method combining both timing and acoustic information is discussed in Section 5.4. The experimental setup and results are reported in Section 5.5 with a final chapter summary in Section 5.6. This chapter focusses on Objective 2, as discussed in Section 1.4, investigating semi-supervised speaker diarisation methods. It is supported by a publication (Milner and Hain, 2017).

5.1 Related Research

For acoustic information, the most common form of supplementary data in the literature is IHM channels. Speaker channels have been investigated in terms of combining them to improve quality by removing noise. Beamforming, as described in Section 2.3.2, creates a single channel from an unknown number of microphones using weighted delay-and-sum techniques. Anguera et al. (2007) proposed algorithms to reduce the impact of low quality channels. The algorithms considered automatic selection of the reference channel, the computation of the N-best channel delays, post processing techniques to select the delay values, and a dynamic channel-weight estimation. When tested against the RT06 evaluation test set, the methods achieved a 25% relative improvement compared to using the SDM channel. It is a popular technique and was widely adopted for systems in RT07 and RT09 evaluations. Alternatively, Milner (2012) performed diarisation directly on the IHM channels. An energy detection method on the channels was applied to studio recorded data (TBL, see Section 3.3.1) to detect speech. This was assumed to work well but the results were poor with DERs above 100% due to crosstalk, even with a cross-meeting normalised energy feature (Dines et al., 2006) applied. Aronowitz (2011) encoded prior knowledge of the acoustic information, such as speakers, channels, background noise and gender, in the feature domain. Standard features, such as MFCCs, were modified to combine the information sources from both the standard and a priori information. The features were applied to two state-of-the-art diarisation systems, one

BIC-based and the other supervector-based (Aronowitz, 2010). Performance improved for both systems when the a priori information came from a known speaker, however only small gains were seen when the information came from a speaker not present in the recording.

For speaker information, supplementary data exists in the form of pretrained speaker models (Reynolds, 2002). Models for speakers are commonly applied in the speaker identification and speaker verification fields but not for speaker diarisation. Moraru et al. (2004a) adapted their pre-existing diarisation system (Moraru et al., 2004b) to include speaker models and investigated the performance when one or all of the participants were modelled. The system used a BIC-based segmenter followed by AHC using a HMM-GMM model with a BIC distance metric and stopping criterion. Speaker models were created from mean-only Maximum A Posteriori (MAP) adapted UBMs. When the system was provided with all models the decision was made segment-by-segment as to whether the segment belonged to a certain speaker or not. When training data was available for a single speaker, the speaker model was used to either pre-segment or post-segment the recording. Pre-segmentation detected when the speaker talks and the remaining speech was segmented using BIC. Post-segmenting resegmented the final system output, the speaker labelled segments, to refine the speaker labelling. Experiments were carried out on RT03 data and broadcast news ESTER databases (Galliano et al., 2006). The results showed that when there was sufficient data to train speaker models for all speakers, then these worked well. However, when one speaker model was presented only a small gain was seen in both methods. Moraru et al. (2004a) did not investigate a range of speaker models, they only used all or one. The same work also investigated using the number of expected speakers as prior knowledge. The system used BIC as a decision metric and stopping criterion. The stopping criterion was removed and the clustering continued until the specified number of speakers was met. Interestingly, the experiments that used prior knowledge gave mixed results. Improvements were seen for the one dataset but not for the other. However, neither result performed better than manually choosing the number of speakers which gave the lowest error rate. The correct speaker number did not give the lowest error. Given the mixed results, further experiments are required across different datasets.

Transcripts are not commonly used in a speaker diarisation as words are not important. However, the timing information contained has been shown to be useful. Lamel et al. (2004) presented a diarisation system using speech transcripts from BN data. Linguistic patterns in the text were used to identify speakers in a rule-based fashion. Three classes were proposed: who is speaking (current speaker), who will speak (next speaker) and who just spoke (previous speaker). A total of 52 patterns spread over these three classes were detected in the transcripts. Decision rules were defined to detect the speakers based on the names

mentioned in the transcripts. Unseen data from the NIST evaluations in the late nineties were used for evaluation. The system achieved 8.9% speaker identification error rate, however, it was noted that only 10% of segments contained speaker information which was applicable for this method. This meant most segments cannot be given a speaker label in this way and therefore would need to be combined with other methods to complete the speaker labelling stage. Conversely, Milner (2012) applied transcript alignment which aligned the speaker labelled words to the IHMs, resulting in speaker labelled segments with timings. A small version of the TBL dataset was used for evaluation which is known to have incomplete transcripts (Section 3.3.1). Despite this, the transcript alignment method performed well and outperformed the energy based IHM method.

The research cited has investigated methods involving supplementary data: speaker channels, speaker models and transcripts. IHM speaker channels are often combined using beamforming to produce a single channel of better quality (Anguera et al., 2007). Speaker models have been investigated where it was shown that having models for every speaker improved results, however models performed better when trained on more data (Moraru et al., 2004a). Speech patterns from transcripts where investigated by Lamel et al. (2004) to predict speaker labels, whereas transcripts were aligned by Milner (2012) with good performance despite imperfect data. This outperformed an energy-based SAD method acting as diarisation using the imperfect IHM channels as input.

5.1.1 Motivation

Unsupervised methods for speaker diarisation, are preferred as this results in a system which is robust. This means that the system is applicable to many datasets and performs acceptably without the need to tune many parameters. However, this robustness is at the sacrifice of performance as research that involves auxiliary information has shown improvements over methods which do not (Anguera et al., 2007; Moraru et al., 2004a). This shows that a supervised or semi-supervised approach to diarisation is better when the necessary auxiliary information is available. Transcripts and IHMs are common types of auxiliary information across datasets. Research into methods involving these two types of supplementary data are investigated to enhance a speaker diarisation system. Speaker information in terms of pretrained speaker separation models are investigated separately in Chapter 6.

A transcript alignment method was investigated which performed well despite datasets containing imperfect transcripts (Milner, 2012). It is often assumed with speaker channels that performing diarisation on these would work well and the problem is solved. However, it was shown in Section 3.5.2.5 that low quality IHM channels can produce highly erroneous results. Data laden with crosstalk leads to high false alarm rates as depicted in Figure 5.1.

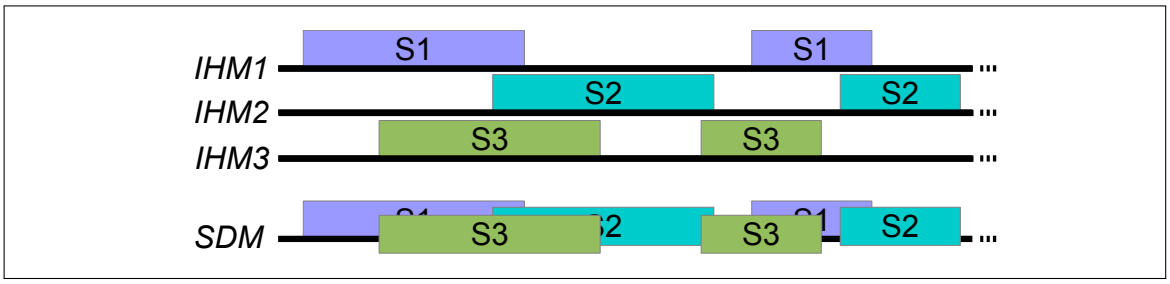

Fig. 5.1 When diarisation is performed on IHM channels containing heavy crosstalk, the corresponding SDM performance will contain overlapping speech. In evaluation this becomes FA which, if high, can lead to over 100% DER.

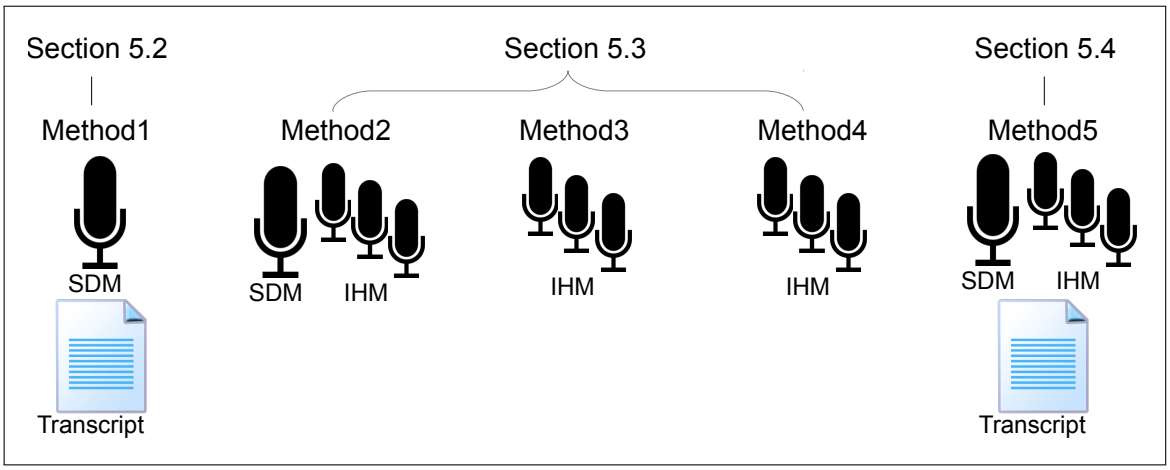

Fig. 5.2 For the five proposed methods, the types of supplementary data required for each is depicted. The SDM data is either combined with IHMs or transcripts or both. Methods 3 and 4 rely on the IHM channels only.

High amounts of speech has been detected across the IHMs. When assuming there is a single speaker per channel, the corresponding SDM result contains a large amount of overlap and therefore FA. Given the way the DER is calculated, if more speech is detected than the total speech time in the reference, the DERs can quickly rise above 100%.

Five semi-supervised methods are proposed which use a combination of imperfect IHM channels and transcripts as supplementary data. Figure 5.2 displays the data types necessary for each method. Firstly, a transcript alignment method is presented in Section 5.2.1. Three methods involving IHMs are then described. The first in Section 5.3.1 includes SDM data as well as IHMs in a two-step method: SAD combined with IHM frame scores. Next, a method which trains DNNs on concatenated IHM features is detailed in Section 5.3.2. An extension to this method described in Section 5.3.3 is portable to any dataset containing IHM

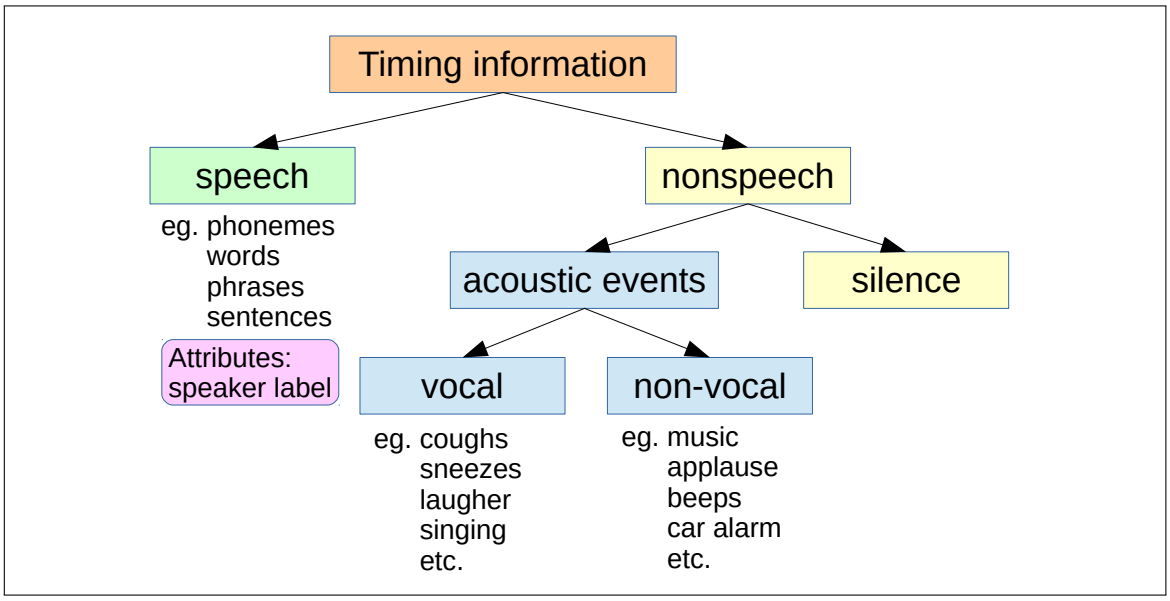

Fig. 5.3 Timing information can be sorted into two groups, speech and nonspeech. For the case of diarisation, the speech detected requires a speaker label. Nonspeech can be further divided into acoustic events, vocal or non-vocal noises, and silence.

channels. Crosstalk features are applied to these methods to help reduce its negative impact on performance and training with or without overlapping speech is investigated. Additionally, the related research does not investigate a method which combines both transcript and speaker channels. Thus, a method is examined in Section 5.4 which combines SAD segments with the transcript alignment to improve the performance when it is known that transcripts are imperfect.

5.2 Timing Information

Timing information gives a start and end (or duration) in time to an audio object. As seen in Figure 5.3, these timed objects can be grouped into two sets: speech and nonspeech. Speech is the relevant information sought after in speech technology tasks such as diarisation and ASR, and nonspeech is everything else which is ignored. Speech can be split into different units: phonemes, words, phrases or sentences. Phrases, or segments, with timing information is part of the goal of speaker diarisation. These speech units can have additional information attributed, other than a start and end time. If a speaker label is attributed to the given words, speaker boundaries can be inferred, as when the speaker label changes across speech a change point is detected.

FILE5 1 Speaker45	9.048	12.334	<Female,C1>	Well wait let us see if it
FILE5 4 Speaker33	12.890	15.619	<Male,C3>	Okay did the second flash go
FILE5 1 Speaker45	16.024	16.731	<Female,C1>	There
FILE5 4 Speaker33	16.479	17.692	<Male,C3>	(%COUGH) thanks
FILE5 3 Speaker25	17.742	18.450	<Male,C2>	Good

Fig. 5.4 Example of a rich transcript containing precise timings for segments of words. Speaker labels are also included as well as additional information such as vocal noises (COUGH), speaker genders and the speakers' IHM channel.

Nonspeech refers to anything that is not classed as speech. This itself can be split into two groups: acoustic events and silence. Silence is relative easy to detect, given energy-based SAD methods described in Section 2.4. However, SAD is usually not as simple as detecting silence from speech. Most recordings contain background noises such as an air-conditioner or low quality microphones which can affect the noise level. Further to this are other noises such as sneezes, music and car alarms. These noises are referred to as acoustic events and are either vocal or non-vocal noises. Vocal noises consist of sound coming from speakers which does not constitute speech. This could be coughs, sneezes, heavy breathing, laughter, singing and more. Non-vocal noises span a wide range of possibilities, such as music, machinery, applause, beeps, alarms, etc. In particular music combined with speech can be detected separately from speech without music (Sinha et al., 2005).

Transcripts provide access to timing information. They typically contain words or sentences with start and end times to which speaker labels are occasionally attributed. An example of a rich transcript can be seen in Figure 5.4. As well as the words spoken, this transcript includes speaker labels, channel labels, the recording filename and segment times. Additional metadata of the speaker gender and acoustic events, i.e. '%COUGH' can be seen. Transcripts, like most real data, exist in varying quality (Fox and Hain, 2013; Stan et al., 2016). Figure 5.5 displays a transcript with minimal information. The words exist and have been segmented with a given speaker label. However, a single time stamp is displayed as opposed to timings for every word, sentence or spoken phrase. Secondly, two speakers are named 'Female' and 'Male' as their real names are not known. It is unknown if a future unnamed speaker would also be given the same gender name, regardless of being the same speaker or not.

Another point to make is shown in both Figure 5.3 and Figure 5.4. The '%COUGH' and 'IT?S' shows that it is necessary to perform some amount of text processing before the transcript is usable in a system, diarisation or other. Unspoken words or specific characters

SirRobertWinston IT?S A VERY NARROW AREA ISN?T IT WHERE SOMEBODY IS SORT OF CLOSE TO THE EDGE LIKE THAT BUT STILL MAKING SENSE?
Female ROBERT WINSTON?S MUSICAL ANALYSIS ON BBC RADIO 4 (12:00:00) BEGINNING TOMORROW AFTERNOON AT HALF PAST ONE.
Male AND NOW TO WESTMINSTER WHERE WILDCAT STRIKES AND THE SEVERE WEATHER ARE CONCERNING MPS. TODAY IN PARLIAMENT.
MichaelMartin ORDER! ORDER

Fig. 5.5 This less informative transcript gives very rough timings, words and speaker labels, but no further information is seen.

must be processed, and in some cases removed, so the transcripts are compatible with a system. Given the large datasets available, this is typically carried out automatically, as manually going through line by line is time consuming and inefficient. This can cause small errors or mistakes to appear in the processed transcripts, negatively affecting a system.

Detecting spoken words is not necessary for diarisation, nevertheless the task of transcript alignment detects regions of speech. Additionally, with transcripts that contain speaker labels, each segment is given a label, thus resulting in speaker labelled segments of speech. This is the information required for a diarisation hypothesis.

5.2.1 Method 1: Transcript Alignment

Method 1 requires a transcript, a dictionary containing the possible words, and appropriate acoustic models. The transcript is aligned to the audio which results in the words of the transcript to be given a start and end time. When there is speech in the audio which does not occur in the transcript, a nonspeech or silence label is given. This method assumes that the transcripts contain speaker labels which leads to the timed words to be attributed to a speaker. This results in the desired output of a diarisation system, speaker labelled segments with a start and end time. The method is an integrated approach. An extension to this is to resegment the alignment output. The alignment may contain silences and pauses between and inside words. A threshold on the acceptable duration of a pause can be implemented where any silence longer than the set duration can impose new segment boundaries. This could improve the segmentation by reducing the false alarm rate.

Transcripts which are mostly correct and contain detailed timing information already perform well in this method. However, erroneous transcripts (as previously described) lead to problematic results. For example, if parts of the transcript are missing, this leads to large amounts of nonspeech being detected as it is not possible to reclaim this through alignment.

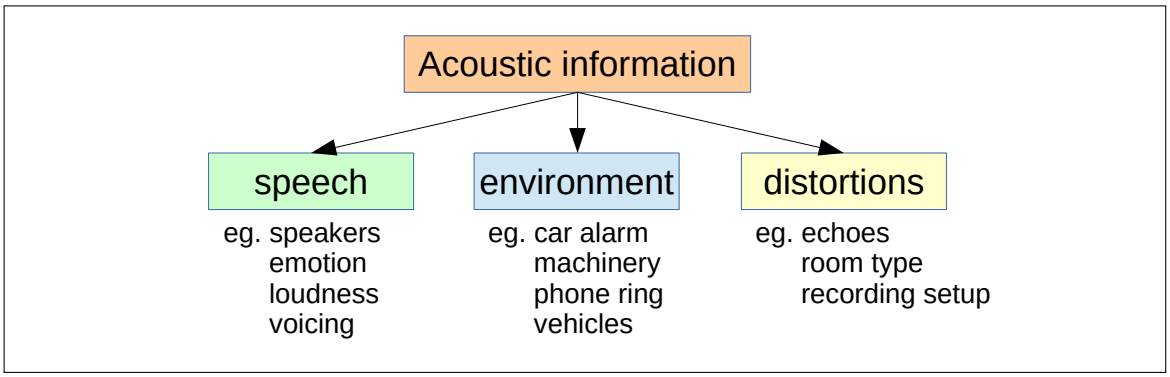

Fig. 5.6 Acoustic information can be organised into three types: speech, the vital part for speech technology; environment, noises around the speakers; and distortions, signal disruptions.

The method can only align the given text. A lack of timing information could also lead to the alignment becoming muddled and losing its place in the transcript, thus assigning incorrect times to the words.

Furthermore, given a transcript which contains speaker labels, each word is given a time from the alignment and inherit a speaker label from the transcript. Speaker changes and thus speaker labelled segments are inferred. This results in speaker-homogeneous speech segments, necessary for speaker diarisation.

5.3 Acoustic Information

Acoustic information refers to the sounds and noises in the audio. Figure 5.6 shows how the acoustic signal can be split into three types of noises: speech sound, environmental noise and distortions. Speech is the relevant and important part of the audio. The speech signal varies depending on the person speaking, their emotion, how loud they are talking, their use of voicing, etc. The preprocessing steps resulting in features extracted from the audio aim to focus on the speech signal as opposed to the other non-relevant signals (see Section 2.3).

Environmental noises contribute to the audio signal in different ways. There could be continuous background noises such as machinery in a factory and air-conditioning. Other noises may be more temporary such as car alarms, vehicles and phones ringing. Noisy speech is harder to detect accurately and organised challenges exist for ASR in noisy conditions (Barker et al., 2015; Harper, 2015; Kinoshita et al., 2016).

The signal can be distorted for various reasons. The room type varies from a studio designed for recording clean audio, to meeting rooms or to the streets in which many

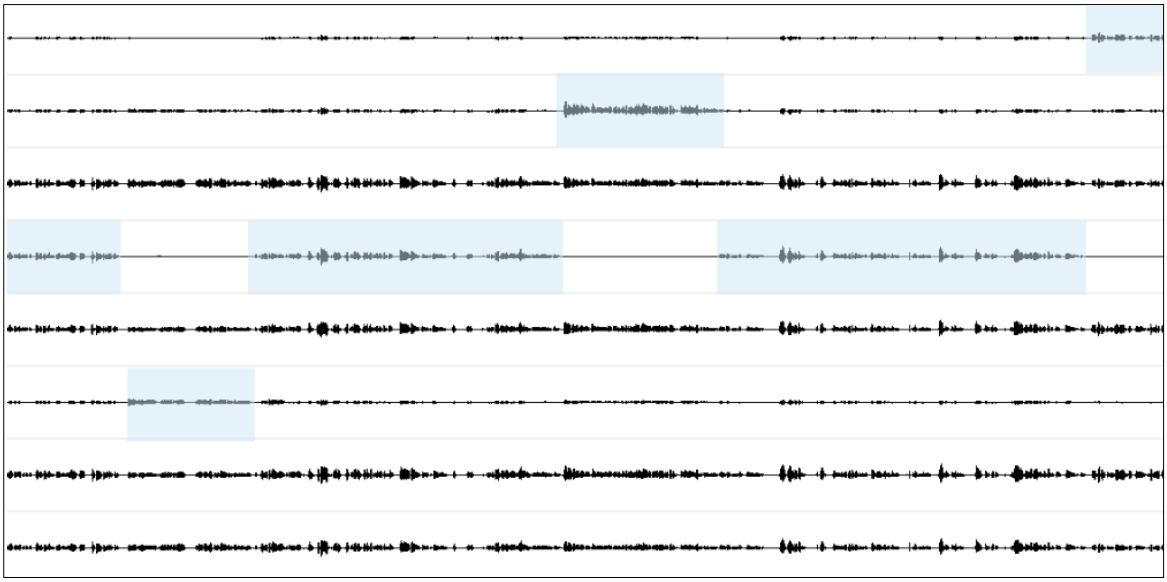

Fig. 5.7 Eight channels from a broadcast media programme are displayed. The highlighted regions represent speech from speaker-specific IHM channels, showing only four of these channels belong to speakers.

different noises can occur. The recording setup has an impact on the ASR performance across systems (Barker et al., 2015) and the higher the number of microphones the better the performance, as in the 4th CHiME Speech Separation and Recognition Challenge[1].

Acoustic information comes from the microphone recordings which can be divided into three types: SDM, MDMs and IHM channels. Datasets across domains may contain one of these types, a combination of two or all three. Detailed examples of different setups is seen in the Sheffield Wargame corpus (Fox et al., 2013, 2016). There may be one SDM channel in the centre of the room or several positioned around a room. Using this channel type is known to perform worse than the MDM condition for diarisation (Ajot and Fiscus, 2009). A microphone array, usually positioned in the centre of the room, can contain several microphones facing in different directions. These microphones are combined using beamforming (Anguera et al., 2007) (described in Section 2.3.2) to produce a single channel which aims to enhance the speech signal along with noise reduction (Adami et al., 2002). IHM channels are microphones on the lapel, head or table specifically for capturing a single speaker's speech. Theoretically, this should lead to straight-forward speaker labelling if it is known that a single speaker appears on a certain IHM. However, this was shown to be difficult in Section 3.5.2.5.

[1]4th CHiME Challenge : http://spandh.dcs.shef.ac.uk/chime_challenge/

Similar to the transcript data, channels can also be of varying quality. Performance can vary relating to the microphones used for recording, the recording room and environment, background noises and more. For speaker data, the number of IHM channels may not be equivalent to the number of speakers. Figure 5.7 shows an example of the channels in the TBL dataset (described in Section 3.3.1 on page 49). Eight channels exist, however, only four channels contain speech from specific speakers. As it is known that there are only four speakers, the correct channel for each speaker can be manually determined. Additionally, there are situations in datasets where only some of the participants have IHMs and other speakers do not (Fox et al., 2016).

Crosstalk appears on IHM channels (Dines et al., 2006; Wrigley et al., 2005). Crosstalk is described in Section 3.1.2 and refers to speech from one or more speakers detected on a different speaker's IHM channel. It can greatly disrupt systems based on IHM channels. Figure 3.1 on page 49 depicts the TBL dataset again. As well as listening to the channels, it is seen in the waveforms how three channels are contaminated with the highlighted speech. This effect is noticeably reduced in the third channel as this channel belongs to the host, who sits opposite the three guests.

The three methods presented incorporate either both SDM and IHM channels or just IHMs into diarisation systems. The first method is an approach combining of SAD on SDM channels with speaker labelling using DNN posterior probabilities from alignment on IHM channels. The second performs IHM channel detection with DNNs when each recording contains the same, fixed number of channels in the dataset. Finally, the third is an extension to the second in which each recording in a dataset contains a mixed number of channels. All three methods assume there exists a single IHM channel for each speaker. Therefore, a channel label is sufficient as a speaker label in all these methods.

5.3.1 Method 2: Combining SAD with IHM Frame Scores

Method 2 is a step-by-step method depicted in Figure 5.8. It consists of several stages: SAD, IHM alignment and frame decision using posterior probabilities. In the first stage, DNNs are trained to detect speech and nonspeech. The second stage takes the speech segments from the SDM channel and aligns them to the IHM channels. A posterior probability for each frame is given in the alignment, meaning each frame will have a score for every IHM channel. The channel with the highest score is decided as the channel, or speaker, for that frame.

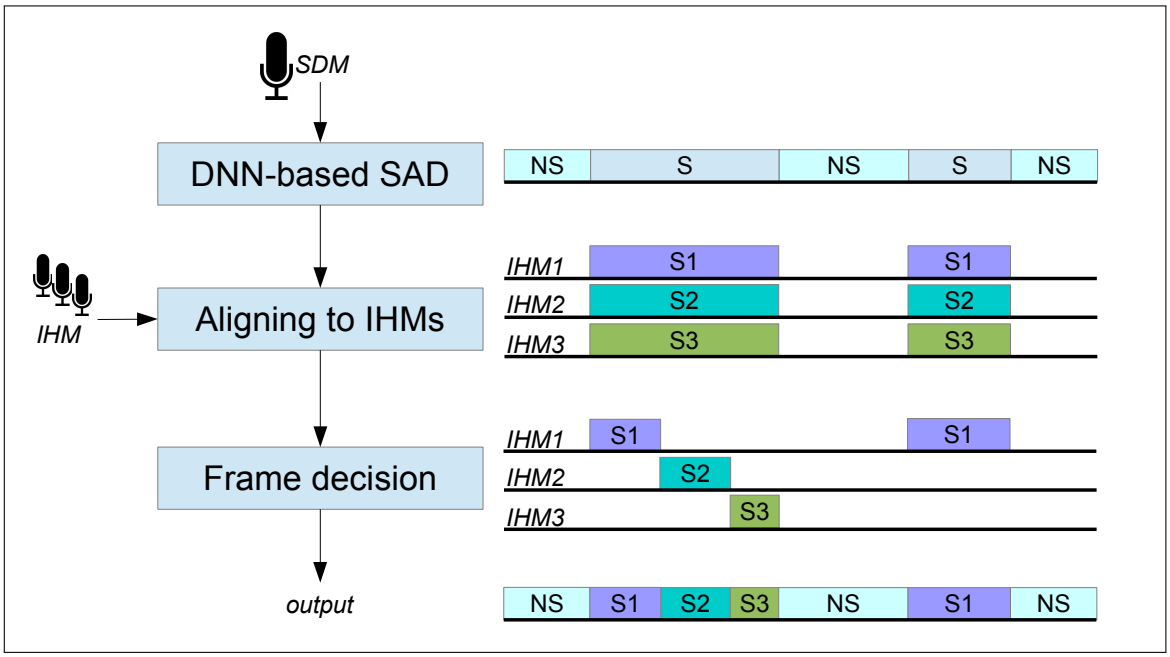

Fig. 5.8 Method 2 performs SAD on the SDM channel to produce a speech and nonspeech segments. The speech segments are then aligned to the IHM channels. Using the resulting posterior probabilities, a frame independent decision based on the posterior probabilities decides the channel.

DNN-based SAD

DNNs can be trained to classify speech and nonspeech given apt training data. This training data can be in the form of SDM, IHM or a combination of the two channel types. When decoding, a minimum duration constraint is applied to vary the quality of the output segments dependent on the task. For example, decoding using HMMs allows the introduction of a minimum state duration which forces longer segments to be detected. This prevents detecting too many segments with a single, or very short, duration. Also, a prior probability for nonspeech can be imposed. This is especially useful for SAD on SDM channels where in a TV programme it could be expected that the majority of the duration is speech. Lastly, a grammar scale factor is another tunable parameter which acts as an encoded probability.

These DNNs are able to detect speech on both SDM and IHM channels. For SAD detection on IHM channels, detecting large amounts of crosstalk would affect the performance. Crosstalk affects the performance by detecting the wrong speaker's speech on an IHM channel. Dines et al. (2006) applied crosstalk detection features investigated by Wrigley et al. (2005) which are described in Section 3.1.2. Dines et al. trained a SAD DNN with additional features to suppress the effect of crosstalk. A "cross-meeting normalised energy

feature" was introduced to reduce the crosstalk detected across the channels. The feature normalised energies across all N channels using:

$$E_i^{norm}(n) = \frac{E_i(n)}{\sum_{k=1}^{N} E_k(n)} \tag{5.1}$$

where $E_i(n)$ is the current channel i energy at frame n. Dividing the result by the sum of all N channels' energy produces a value between zero and one, making the feature independent from the original recording level. Dines et al. further included three other features determined to be useful for crosstalk detection by Wrigley et al.. Firstly, the fourth order moment of a signal divided by the square of its second order moment is known as kurtosis. It has been shown that the kurtosis of isolated speech utterances is typically more than the kurtosis of overlapping speech (LeBlanc and Leon, 1998). Mean cross-correlation and maximum normalised cross-correlation were computed (Wan, 2003). For each channel i, the maximum of the cross-channel correlation $C_{ij}(t)$ at time t between channel i and each other channel j was extracted:

$$C_{ij}(t) = \max_{\tau} \sum_{k=0}^{P-1} x_i(t-k)x_j(t-k-\tau)w(k) \tag{5.2}$$

where τ is the correlation lag, x_i is the signal from channel i, x_j is the signal from channel j, P is the window size and w is a Hamming window. The features described here are applied to the DNN-based SAD models necessary for Method 2.

IHM Alignment

The SAD stage detects speech and nonspeech segments on the SDM channels. The speech segments are assumed to be correct, but at this stage it is unknown which channel each segment belongs to. It is possible, of course, that there is more than one speaker within a speech segment as the SAD does not distinguish speech from different speakers. To determine which channel(s) belong to each speech segment, the segments are aligned to every IHM channel. Similar to the transcript alignment in Method 1, the speech segments are aligned to the IHM audio, as words are not known. The alignment gives a posterior probability for every frame in each speech segment. This can be seen as a score for each frame and represents how confident the system is at aligning that frame as speech. Every speech segment is aligned on all the IHM channels, providing each frame with a posterior probability for every IHM channel. Speaker labelling is performed in which the IHM channel providing the highest posterior probability is selected as the hypothesis channel, and thus speaker, for a frame. This allows a decision to occur on every frame, resulting in a diarisation output as shown in

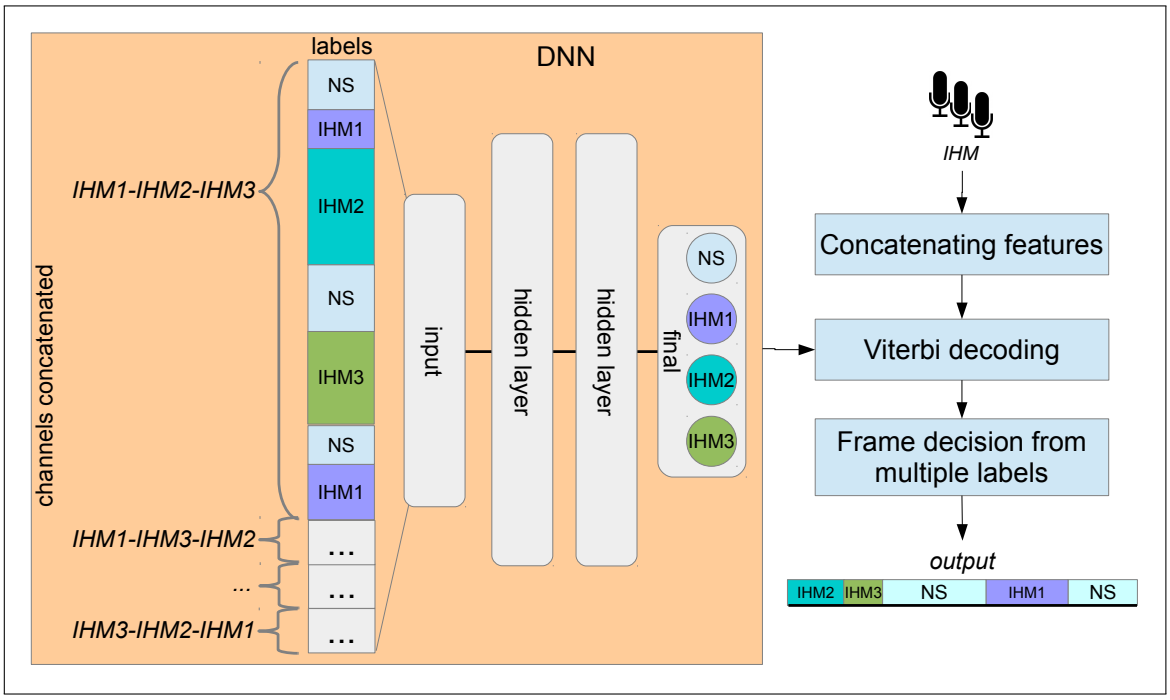

Fig. 5.9 Given a fixed number of channels in recordings, features extracted from IHMs are concatenated and labelled by channel and nonspeech. All the possible permutations of the IHMs are concatenated for training.

Figure 5.8. Alternatively, if the speech segments are known to be speaker-pure then a single speaker is needed for each segment. In this case, the posterior probabilities for the frames in a segment are averaged across the IHM channels to determine the channel, or speaker, which gives the highest score.

5.3.2 Method 3: Fixed Number of IHM Channels

DNNs can be used directly for channel detection. Determining which channel each segment of speech belongs to will infer the speaker label. This is an integrated method as it combines SAD, speaker segmentation and speaker clustering into one stage. The IHM channels required for this method do not need to be recorded sample-synchronous. DNNs are trained on concatenated features of all the speaker channels. It requires every recording to contain the same number of speakers. Every permutation of the concatenated features is used for training, as this may help prevent channels being biased in certain positions. Figure 5.9 depicts the ordering of the concatenated features with their equivalent label file for training. The example assumes there are three IHM channels for every recording. The channels are referred to as *IHM1*, *IHM2*, and *IHM3*. The features of these three channels are concatenated

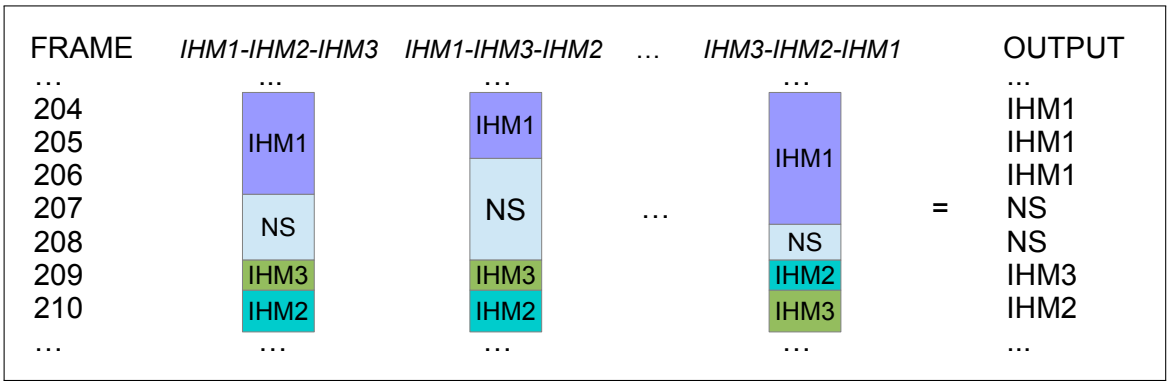

Fig. 5.10 A frame decision technique looks across the multiple channel labelled segments output. Displayed is the counting method which chooses the most occurring label (a channel or nonspeech) for each frame.

together in all the possible permutation, such as *IHM1-IHM3-IHM2* and *IHM3-IHM1-IHM2*. The method consists of three steps and the features are concatenated in the first step. The second step performs Viterbi decoding which produces channel labelled segments for the concatenated features. This is the required diarisation output, however, the third step further refines the output as the second step is performed on many permutations of concatenations. The outcome of the second stage is multiple outputs of channel and nonspeech labelled segments. This provides multiple labels for every frame as shown in Figure 5.10. To make a decision on the correct label, several techniques are investigated: counting occurrences or posterior probabilities, with or without a nonspeech bias. The counting method requires counting the occurrences for a single frame and selecting the channel, or nonspeech, that has been labelled the most. Instead, posterior probabilities are summed for the different labels given to a frame and the label with the highest score is chosen for that frame. Alternatively, the occurrences are counted or posterior probabilities are summed as before with a bias for or against nonspeech applied as a multiplier to increase or reduce the likelihood of selecting nonspeech. A bias for or against specific channels could also be applied, for example if a host in a TV programme is known to talk more than the guests.

This method requires training on speaker-pure segments as it is known that overlap can contaminate a cluster, causing incorrect decisions to be made (Huijbregts et al., 2012; Knox et al., 2012). Two possibilities are chosen for the training data in this method:

- **without overlap:** overlapping time is removed from the speech segments
- **with overlap:** overlapping time is kept in the speech segments

Furthermore, DNNs trained with crosstalk features has lead to improvements in performance when using IHM channels for SAD (Dines et al., 2006) and is investigated for this method.

5.3.3 Method 4: Mixed Number of IHM Channels

Method 3 is not portable to datasets which do not contain the same number of speakers in each recording. A different approach is required where pairs of features are concatenated instead. The method is similar to Method 3, as seen in Figure 5.9, however channel pairs are concatenated instead. Where there is speech on a different channel, this is labelled as nonspeech. As well as being applicable to all datasets, this alternative approach also reduces the amount of data (concatenated features) needed for training. For a single recording in Method 3, the number of possible permutations for training is $x!$, where x is the number of channels. Whereas for this method, the number of possible feature pairs for training becomes $x(x-1)$. For example, if there are 4 channels, then the amount of permutations needed for each method is 24 and 12 respectively. This reduces the amount of computation necessary. As in Method 3, the same options for decoding are applied: counting occurrences or posterior probabilities, with or without a nonspeech bias. The effect of overlapping speech and crosstalk in terms of training the DNN and the decoding stage is also investigated.

5.4 Combining Timing and Acoustic Information

Combining timing information with acoustic information uses transcripts alongside IHM speaker channels. Method 1 aligns transcripts to SDM channels and Methods 2, 3 and 4 incorporate IHM channels, but so far there is not a method which combines these two data types. Transcripts typically contain speech units (phrases, words, phonemes, etc) with timing information. Speaker labels can be attributed to the speech units. Whereas for IHM speaker channels, a speaker identity or label may not be known other than an IHM channel label. Methods 2, 3 and 4 are applicable in situations where only a channel identity is sufficient. However, to produce a method which combines transcripts with IHM channels, links between the transcript speaker labels and the IHM channels must be known. This allows for words to be attributed to a speaker, from the transcript, and to a channel, from the IHM channels.

A method is proposed which aims to use the best performing parts from the previous methods. This includes: SAD with DNNs (Method 2, Section 5.3.1), transcript alignment (Method 1, Section 5.2.1), and IHM alignment (Method 2, Section 5.3.1) to calculate posterior probabilities across channels. The last step is compared with selecting the channel with the highest energy for each frame. This method assumes transcripts have speaker labels and there is one IHM channel for each speaker. The last vital requirement is knowledge of which transcript speaker label corresponds to which IHM channel.

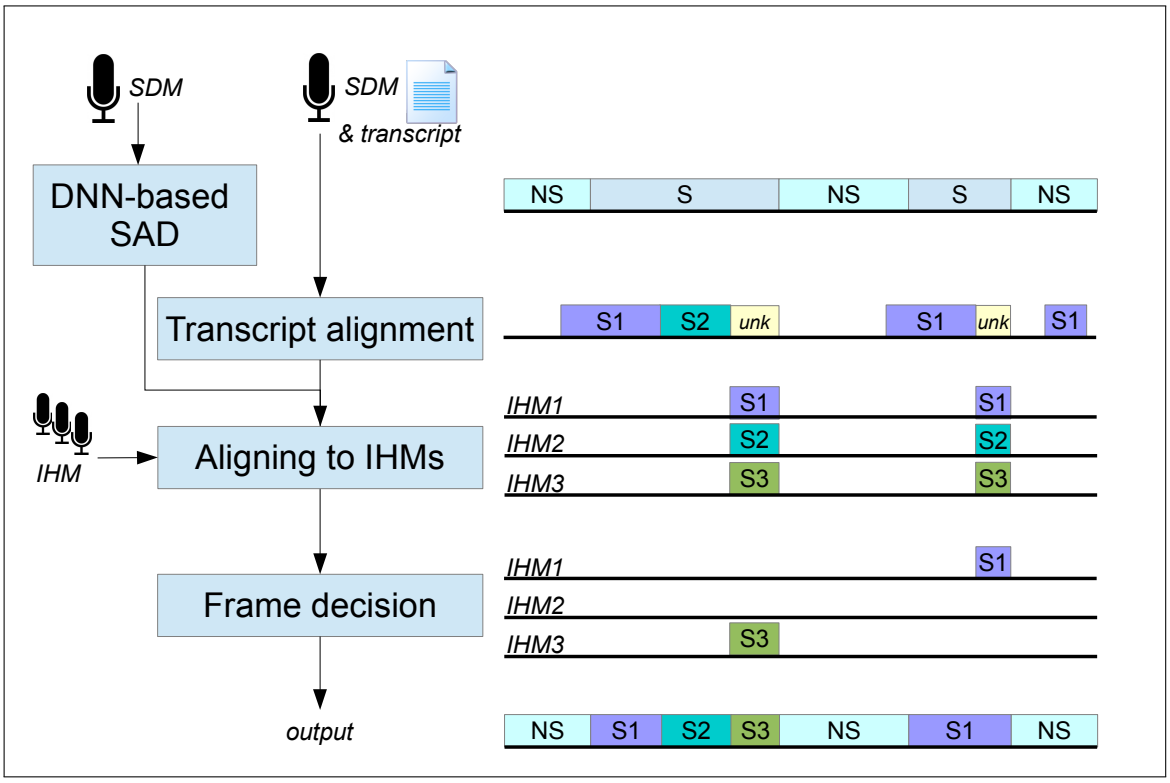

Fig. 5.11 Method 5 consists of three stages. A DNN-based SAD model detects speech and the transcript alignment provides speaker labels for the detected speech. The resulting segments are unlabelled speech segments, 'unknown'. These unlabelled segments are aligned to the IHM channels and a frame decision detects the channel given the posterior probabilities, as in Method 2.

5.4.1 Method 5: SAD, Transcript Alignment and IHM Frame Scores

This method combines both the transcripts and the IHM channels to perform diarisation. It consists of several stages based on previous methods: SAD using DNNs on the SDM channel, speaker labelling based on transcript alignment on the SDM channel, and finally, a second speaker labelling stage. Two techniques are possible in this second speaker labelling stage: energy detection across the IHM channels and posterior probabilities from IHM alignment.

Figure 5.11 depicts the method implementation. First, SAD is performed on the SDM channel as in the first stage in Method 2, detailed in Section 5.3.1. It is assumed that the SDM SAD performs well, as these errors in miss and false alarm speech cannot be recovered from this stage. Secondly, transcript alignment is performed and the SAD speech segments are compared to the output from the alignment. As imperfect transcripts contain errors, the SAD is assumed to be more accurate in terms of segment precision. This means if the transcript

alignment contains a speech segment not detected by the SAD, it remains as nonspeech. However, as SAD does not detect speaker labels, it is necessary to use the ones contained in the alignment output. Lastly, a second labelling stage is necessary. There can be portions of speech in the SAD output which the alignment did not detect. These remain as speech but it is necessary to detect a speaker label. This is when the IHM channels are incorporated.

Two techniques are possible for the final speaker labelling step. The first technique for speaker labelling is the same as the second step in Method 2 (Section 5.3.1). The IHM channel with the highest posterior probability is selected for each frame. Again, this is performed frame-by-frame or by keeping the segments and assuming one speaker exists for each unlabelled speech segment. The second is a form of energy detection, and removes the need to align the speech segments to the IHM channels. Standard features are extracted from the audio, e.g. MFCCs as described in Section 2.3.3. These features can include a dimension containing the energy for every frame. It is assumed that the channel with the highest energy for a given frame contains the current speaker. This allows for a frame-by-frame decision. Similar to the second stage in Method 2, the segment of unlabelled speech can remain a whole segment with one speaker label by averaging the energy on the IHM channels and then selecting the channel with the highest average energy.

5.5 Experiments

Five distinct methods have been presented which use a combination of transcript and IHM channels as auxiliary information to a diarisation system. Experiments are performed to test various aspects of each method. The datasets used are described in Section 5.5.1 with the experimental setup detailed in Section 5.5.2. Firstly, the baseline results from a public domain toolkit are discussed then each of the five presented methods are investigated and evaluated in Section 5.5.3. Lastly, an overview of the results are seen in Figure 5.17 and the outcomes are discussed in Section 5.5.4.

5.5.1 Data

To evaluate the five proposed methods, the datasets applied must contain transcripts and both IHM and SDM channels. Of the datasets described in Table 3.7, two test sets are applicable: TBL and RT07. The TBL dataset is broadcast media in a meeting format, and the RT07 dataset is meeting data. Both have SDM and corresponding IHM channels. Transcripts are of questionable quality exist for TBL, but not for RT07. Instead, the output of an ASR system trained on three meeting corpora (Documeet, Ecorner and TED talks) is used in place of a

transcript for Method 5 (Hasan et al., 2015). For both TBL and RT07, the number of speakers is equivalent to the number of IHM channels. Training data is necessary for the methods which involve DNNs. For TBL there is an equivalent training set from the same broadcast media programme, involving both SDM and IHM channels. For RT07, the AMI meeting corpus is the best match to RT07 meeting data and the IHM channels are used.

For RT07, this chapter uses the SHEF scoring method as the references and stricter collar are more accurate (previously discussed in Section 4.1). This means the results on RT07 are not comparable to other published work. However, the best performance is rescored in the discussion using the NIST method to produce a result comparable to other research.

5.5.2 Setup

For Method 1 and Method 5, transcript alignment is performed to produce segments with speaker labels where possible. Method 2 and Method 5 also rely on aligning speech segments to IHM channels. Viterbi is applied as the alignment algorithm and the programme HVite from HTK[1] (Young et al., 2006) is used. The process uses acoustic models trained on more than 170 hours of meeting data features extracted as PLPs from the AMIDA RT09 system (Hain et al., 2010).

For the methods which require DNN-based SAD, various DNNs are trained on different combinations of data from SDM and IHM channels. The DNNs are trained using TNet[2] (Veselý et al., 2010). The training algorithm applies stochastic gradient descent with cross-entropy error propagation. Log Mel-filterbank features with 23 dimensions are used for training the DNNs as they appear to yield better performance than MFCCs with DNNs (Hermansky and Sharma, 1998). A context window of 16 frames is applied and so 31 adjacent frames are decorrelated and compressed with Discrete Cosine Transform (DCT) into a dimension of 368 (31x23 $\rightarrow$16x23). For each dimension global mean and variance normalisation is computed before feeding into the DNN as input (Liu et al., 2014). Filterbanks are also concatenated with crosstalk features of 7 dimension, leading to a dimension of 480 once processed. Two layers of 1000 hidden units are followed by the final output layer with 2 target classes: speech and nonspeech.

The DNNs for the concatenation methods, Methods 3 and 4, are trained in the same fashion. There is a difference in the number of neurons in the input and output layers. For Method 3 when the number of speaker channels is fixed for every recording, only the TBL test set is applicable as it contains 4 speakers in every episode. With filterbank features there are 1472 input neurons and this increases to 1920 with crosstalk features. There are $N+1$

[1] HTK: http://htk.eng.cam.ac.uk/

[2] TNet: http://speech.fit.vutbr.cz/software/neural-network-trainer-tnet

Table 5.2 Baseline results using the SHoUT toolkit for both RT07 and TBL datasets. This is performed on the SDM channel.

Data	Seg%	Spk%	MS%	FA%	SE%	DER%
TBL	81.7	172.5	4.0	13.1	8.7	25.8
RT07	75.6	117.1	7.4	27.7	14.4	49.5

target classes representing the *N* channels plus nonspeech. Method 4 concatenates pairs of features. For two channels there are 736 input neurons, increasing to 960 with crosstalk features. The three final target classes represent the pair of channels plus nonspeech.

Method 5 considers the energy across the IHM channels. This is calculated by producing PLPs (as detailed in Section 2.3.3) where the 13th dimension represents the energy for each frame. The PLPs are produced using the HCopy programme from HTK.

5.5.3 Results

Results initially present the baseline experiments and then the experiments for the five presented methods. The first method involves timing information in the form of transcripts. The second method requires both SDM and IHM channels whereas Methods 3 and 4 only need IHMs. Method 5 involves all three types of auxiliary data. Experiments are performed to investigate the benefits or negative impacts each type of auxiliary data has over methods which do not use auxiliary data.

5.5.3.1 Experiments: SHoUT

Several toolkits exist for diarisation and these have been described in Section 3.4. The results in Table 3.9 and Table 3.10 show that the toolkit SHoUT achieves the lowest DERs across the datasets from different domains. SHoUT is designed for meeting data and based on the ICSI-RT07 system as described by Huijbregts and Wooters (2007). It consists of two parts: semi-supervised SAD using pretrained models from BN data and unsupervised clustering with resegmentation. The objective is to show performance on a semi-supervised method which has not used in-domain data. The results on IHM channels have previously been shown to be unsuccessful in Table 3.15 due to high mounts of crosstalk. Results for both datasets in terms of SDM channels are presented together in Table 5.2. The toolkit has under-segmented and under-clustered for both datasets. This means fewer than the expected number of segments and speakers have been detected. The expected number is the amount in the reference. The largest error is detected as FA and less error is labelled as MS. The RT07 DER is high at almost 50% implying the 25.8% on TBL could also be higher than

Table 5.3 The SDM performance for Method 1. Transcript alignment is applied to the TBL meeting dataset and the output is resegmented.

Channel	Alignment	Seg%	MS%	FA%	SE%	DER%
SDM	Aligned	17.1	8.8	11.9	3.7	24.5
	Resegmented	38.5	10.6	7.5	3.5	21.7

expected. These results show plenty of room for improvement, which the methods that include supplementary data aim to achieve.

5.5.3.2 Experiments: Method 1

Transcript alignment is performed in Method 1 and, as mentioned in Section 5.5.1, this can only be performed on TBL data. This method aims to see whether improvements are seen in the diarisation performance when transcripts are included in a system. The transcript is aligned to both the SDM and IHM channels to see how each channel type benefits from the transcripts. The transcripts provided for the recordings are not complete and have various issues, unlike the rich example of Figure 5.4. The TBL transcripts contain words each with a speaker label attributed. However, time stamps only appear every 5 minutes within the text. Additionally, the audio provided is the raw recording which includes the participants informally chatting at the beginning and the end, when the content for the programme is not formally happening. However, this does not appear in the transcript.

Transcript alignment on SDM channels

Table 5.3 displays the SDM result when the transcript is aligned and after it has been resegmented. The number of speakers is not displayed as this method gives the correct number due to the speaker labels in the transcript. Both the aligned and resegmented outputs improve over the toolkit baseline. This shows how involving transcripts as auxiliary data has helped the diarisation performance, despite the problems with the data. The SE has more than halved due to the information in the transcript. The MS rate is higher as portions of the transcript are missing. The resegmented output gives a higher MS and lower FA as portions of speech within the aligned segments has been relabelled as nonspeech. It introduces excess nonspeech. This shows that the transcript is not completely reliable.

Transcript alignment on IHM channels

The 5 minute segments were also aligned to the IHM channels and resegmented, with the assumption that those segments which do align to speaker channel *IHM1*, for example, belong

Table 5.4 The IHM performance for Method 1. Transcript alignment is applied to the TBL meeting dataset and the output is resegmented.

Channel	Alignment	Seg%	MS%	FA%	SE%	DER%
IHM	Aligned	31.4	65.5	141.4	0.0	206.9
	Resegmented	72.2	66.3	129.1	0.0	195.4

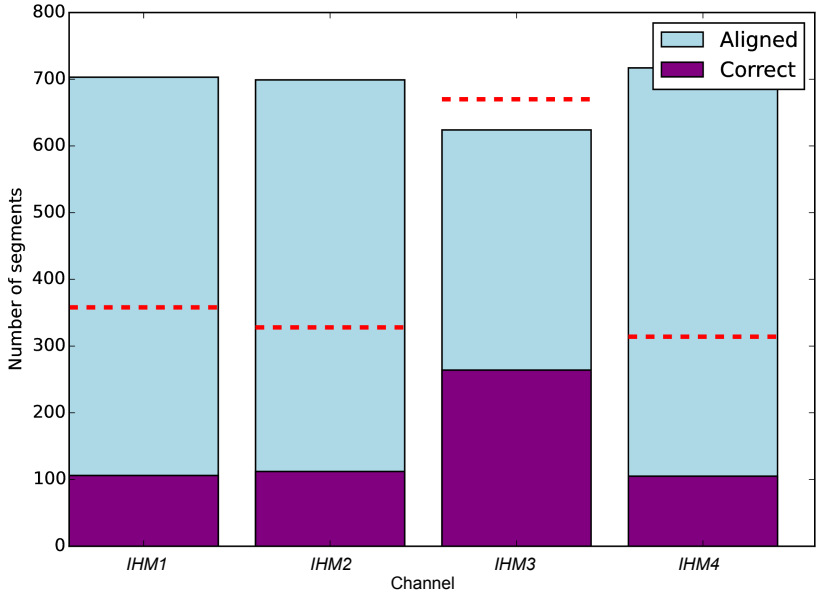

Fig. 5.12 Analysis of Method 1 on the IHM channels for TBL in which the expected (the amount in the reference) number of segments is the dotted line for each channel. The number of segments which have been aligned is shown including the amount which are correct.

to the speaker on channel *IHM1*. The results show this to be unsuccessful as displayed in Table 5.4. Similar to the baseline IHM results in Table 3.15, the DERs are greater than 100%. However, this time a larger proportion is detected as MS, instead of the majority being detected as FA. The segments which were aligned are investigated and an analysis is seen in Figure 5.12. The third channel belongs to the host. For the guest speakers, around twice as many segments as expected have been aligned to their respective channels. Roughly only 15% of the segments were correctly assigned. The hosts channel gives a different result as fewer than expected segments were aligned and 22% were correctly aligned. This is due to the crosstalk negatively affecting the guests more than the host, as the host sits opposite the guests. These results again show performing diarisation on the IHM channels is not an easy task.

Table 5.5 Training details of the DNN-based SAD models in which two datasets are used: IHM and SDM channels from TBLTRAIN and IHMs from AMI.

DNN	Training data		
	TBLTRAIN-SDM	TBLTRAIN-IHM	AMI-IHM
SAD1			x
SAD2		x	x
SAD3	x		x
SAD4	x	x	x
SAD5		x	
SAD6	x		
SAD7	x	x	

5.5.3.3 Experiments: Method 2

Method 2 is the first of the three presented methods incorporating speaker channel supplementary data into a diarisation system. The first stage trains DNNs for SAD. Different training datasets and combinations of these are investigated, along with combining filterbanks with crosstalk features. This is combined with a speaker labelling stage. The speech segments are aligned to each of the IHM channels which results in a posterior probability for every frame across the channels. For each frame, the channel with the highest posterior probability is selected.

DNN models for SAD on SDM channels

This section investigates how best to train a SAD DNN in terms of training data and features. Table 5.5 describes the training data applied to the different DNNs. Three datasets were considered and every possible combination of these is trained. It is known that training within domain improves performance so the TBLTRAIN represents the TBL test set and the AMI meeting data represents the RT07 test set. These models are applied to both TBL and RT07 test data. It must be noted that the MS and FA scores are scored against the total speech time, as opposed to the speech and nonspeech time.

Table 5.6 shows the results for the DNN-based SAD models for both TBL and RT07 on SDM channels. This task aims to detect speech and nonspeech only, and does not provide speaker labels for the speech segments. This means there is no speaker error and the DER is simply the sum of the MS and FA error rates. To find the optimum result for SAD, the number of states and the prior probability for speech have been varied. The number of states refers to adding a minimum constraint duration to the HMM for speech and nonspeech. For example, a minimum constraint of 10 frames forces a segment to be at least 10 frames in duration.

Table 5.6 Results of SAD for Method 2 when using DNN models trained on different combinations of training data are shown in Table 5.5. The expected number of speaker-pure segments is 8749 for TBL and 11144 for RT07.

DNN	TBL				RT07			
	Seg%	MS%	FA%	DER%	Seg%	MS%	FA%	DER%
SAD1	76.6	1.9	2.0	3.9	91.6	27.9	1.2	29.1
SAD2	74.4	16.9	1.4	18.3	93.2	28.5	1.1	29.6
SAD3	79.0	0.6	2.3	2.9	96.0	27.3	1.1	28.4
SAD4	65.0	5.8	2.2	7.9	95.9	27.8	1.2	29.0
SAD5	91.3	25.0	1.0	26.0	77.6	36.7	8.7	45.4
SAD6	74.0	0.5	2.4	2.9	63.7	3.7	6.7	10.4
SAD7	71.0	7.1	2.0	9.1	75.7	18.9	6.3	25.1

Changing prior probability for speech acts as information about how much speech to expect within the recording. For TBL, the performance varies from 26.0% to 2.9%, however most DNNs achieve DERs of less than 10%. *SAD2* and *SAD5* have higher DERs as these models were trained on IHM data only. This shows that performance improves when training and testing on the same channel type. Both *SAD3* and *SAD6* are trained with TBLTRAIN-SDM, same domain and channel type, and achieve 2.9% DER with the latter model detecting fewer segments. For RT07, the results are generally worse. *SAD1-4* are trained on AMI-IHM data and these DNNs give low FA rates. However, large amounts of MS is seen except for *SAD6* which is trained on TBLTRAIN-SDM data. This model gives the lower DER of 10.4% which shows that training on SDM data without IHM data is necessary for better performance on RT07 data. It also shows that training on SDM data from a different domain is better than training on IHM from the same domain in this situation. It is clear from the results that there is a large difference between the performance on TBL and RT07. However, both datasets achieve their lowest DER with *SAD6*, trained on TBLTRAIN-SDM.

DNN models for SAD on IHM channels

The models trained are applied to the IHM channels as well, bearing in mind both Tables 3.15 and 5.4 have shown the difficulties in performing diarisation. The results are seen in Table 5.7. For TBL data, the models *SAD1*, *SAD3* and *SAD6* have results over 100% DER and *SAD3* and *SAD6* detect an extremely large number of segments. These three are not trained on TBLTRAIN-IHM which shows that DNNs trained with in-domain data and the same channel type is key to achieving better performance on these crosstalk-heavy speaker channels. *SAD5*, trained on TBLTRAIN-IHM only, achieves the lowest DER of 20.0%. The RT07 results show that training on AMI-IHM, the same data domain, achieves the best performance. The

Table 5.7 SAD results of Method 2 for IHM for both TBL and RT07 in which every combination of the training datasets is applied.

DNN	TBL				RT07			
	Seg%	MS%	FA%	DER%	Seg%	MS%	FA%	DER%
SAD1	111.6	1.6	195.4	197.0	84.8	10.9	2.8	13.8
SAD2	60.3	6.3	16.7	23.1	87.7	11.2	2.6	13.8
SAD3	801.1	1.1	237.5	238.5	87.3	10.5	3.0	13.4
SAD4	63.1	4.3	25.1	29.4	87.7	10.6	3.2	13.8
SAD5	60.4	8.3	11.7	20.0	61.8	20.6	64.2	84.8
SAD6	912.6	1.0	236.3	237.3	100.2	3.1	82.9	86.0
SAD7	65.7	4.7	21.7	26.4	69.2	12.8	36.9	49.7

models trained without AMI-IHM achieve high DERs, however, the DERs are not above 100%. The lowest DER of 13.4% is achieved with the model *SAD3* which is not trained on TBLTRAIN-IHM. The best SAD model for RT07 outperforms the best SAD model for TBL. This is an opposite outcome to that seen in the SDM experiments. The crosstalk has a worse effect on the TBL data which leads to the next experiment where additional crosstalk features are investigated.

Training SAD DNNs with crosstalk features

Considering the previous DNNs trained on IHM data only, filterbanks and crosstalk features are concatenated and comparative DNNs are trained. It is investigated whether training on crosstalk features improves the SAD performance on the IHM channels. Results are seen in Table 5.8 and comparable to the results in Table 5.7. It is immediately clear that crosstalk features have improved the performance on the IHM channels. For TBL, the model *SAD1* gave a DER of nearly 200% which is reduced to below 100%, to 86.4%, by *SAD1+CT*. The previously best DER of 20.0% has been reduced to 11.9% with the inclusion of crosstalk features. Similar improvements are seen for the RT07 data, with the largest performance gain from 84.8% to 22.7% with model *SAD5+CT*. Adding crosstalk features to *SAD3* helps improve the previously best result of 13.4% down to 8.4%. Training DNN models on crosstalk features, alongside filterbanks, is beneficial across domains for IHM channels.

IHM alignment for frame scores

The second stage to the method proposed in Section 5.3.1 is a channel labelling stage. The previous output speech segments from SAD on the SDM channel are aligned on each IHM

Table 5.8 SAD results of Method 2 when training on crosstalk features (+CT) for IHM for both TBL and RT07. Using crosstalk features is only applicable to models trained on IHM data only.

DNN	TBL				RT07			
	Seg%	MS%	FA%	DER%	Seg%	MS%	FA%	DER%
SAD1+CT	73.2	4.0	82.3	86.4	87.1	5.1	4.0	9.1
SAD2+CT	74.4	3.3	13.2	16.5	88.4	5.0	3.5	8.4
SAD5+CT	68.4	4.8	7.1	11.9	88.7	12.0	10.7	22.7

channel given one of the trained SAD models. A posterior probability for every frame is calculated. Each frame is labelled as the speaker channel with the highest posterior probability. This produces a frame independent decision, no context in terms of surrounding frames is considered. The experiment shows whether the speaker labelling method is successful and shows which models give the best performance.

Speech segments on the SDM channels are output from every SAD DNN model which was not trained on crosstalk features. All the SAD models can be used to align the segments to the IHM channels. All possible combinations of the two are considered and the DER results are displayed in Figure 5.13. The same scale is applied for both datasets to show the differences between the alignment models and the segments across the two domains. The figure shows how the DNNs previously trained for SAD perform when aligning the segments from the previous stage. The TBL results show many combinations have similar performance. Speech segments from *SAD2* and *SAD5* and the alignment models *SAD3* and *SAD6* are outliers with worse performance. The former are not trained on TBLTRAIN-SDM and the latter are not trained on TBLTRAIN-IHM. This shows the speech segments are higher quality when trained on in-domain SDM data and alignment models trained on in-domain IHM perform better when aligning segments to IHM channels. Combining these alignment models with erroneous segments gives the worst performance. The RT07 results show clearly that speech segments from model *SAD6* lead to better performance and speech segments from *SAD5* leads to worse performance. The remaining models give similar results. *SAD6* is trained on TBLTRAIN-SDM data and achieved the best SDM SAD results as seen in Table 5.6. Model *SAD5* is trained on TBLTRAIN-IHM which is a mismatch in data and channel type. *SAD5* and *SAD6* are the two alignment models which give slightly worse performance as neither are trained on AMI-IHM, the same domain as RT07 data.

The best performance on both datasets is seen in Table 5.9. For TBL, the model *SAD2* aligns the speech segments from model *SAD3*. The highest error is SE at 6.1% and the final DER is 12.5%. This is a 52% relative reduction from the baseline. For RT07, the segments from *SAD6* are aligned with the *SAD2* model. *SAD2* achieves the best performance

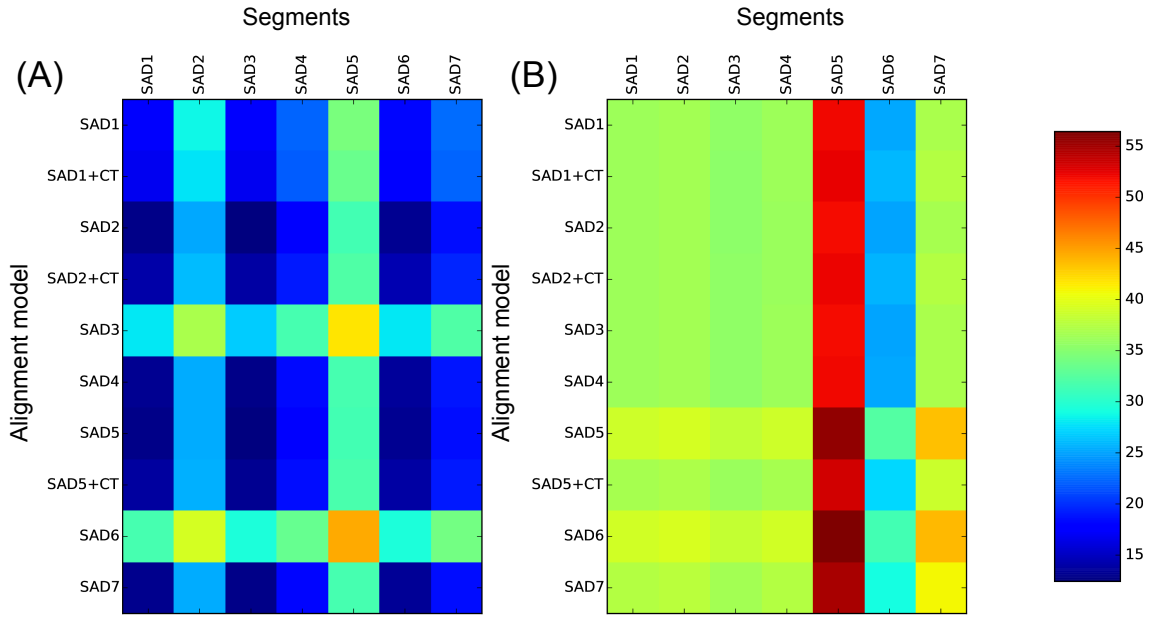

Fig. 5.13 Diarisation performance for Method 2 for (A) TBL and (B) RT07. 'Segments' refers to the speech segments from the SAD stage and 'Alignment model' refers to which SAD model was used in the the IHM alignment step. The scale is the same to show the different ranges of the TBL and RT07 DER performance.

Table 5.9 Best performance for Method 2 for TBL and RT07 datasets. Both use the alignment model from SAD2 but different segmentation outputs give better results: SAD3 for TBL and SAD6 for RT07.

Data	Segmentation	Alignment	Seg%	MS%	FA%	SE%	DER%
TBL	SAD3	SAD2	99.5	4.4	2.0	6.1	12.5
RT07	SAD6	SAD2	88.2	10.6	6.5	8.0	25.1

across both datasets. The DER is 25.1% and the highest error is MS. This is a 49% relative reduction in error when compared to the baseline performance. Both datasets have seen a roughly 50% relative improvement over the baseline. It shows that despite the heavy crosstalk, incorporating speaker channels into a diarisation system outperforms Method 1 which involves transcript data.

5.5.3.4 Experiments: Method 3

Method 3, as well as Method 4, investigates a diarisation method which incorporates IHM channels. This involves training DNNs on features extracted form the IHMs which are concatenated. As opposed to the previous Method 2, the method is an integrated approach. It requires a fixed number of IHM channels per recording, which does not apply to the RT07 data.

Table 5.10 Results for Method 3 when training on overlap for TBL data. Two different training setups are compared, training DNNs with (+OV) and without overlap.

DNN	Seg%	MS%	FA%	SE%	DER%
CAT4	83.1	4.3	2.5	1.5	8.3
CAT4+OV	76.9	4.3	2.6	1.3	8.2

Table 5.11 Results for Method 3 when training on crosstalk features (+CT) in addition to filterbanks for TBL data.

DNN	Seg%	MS%	FA%	SE%	DER%
CAT4+CT	33.9	4.6	3.7	1.4	9.7
CAT4+OV+CT	81.6	4.3	2.4	1.7	8.4

Therefore, only the TBL dataset is evaluated. The objective is to show whether diarisation performed on the IHM channels without SDM as an input achieves good performance. DNNs are trained with or without overlapping speech data and with or without additional crosstalk to investigate how to prevent the negative effects on the performance. Furthermore, experiments on different frame labelling techniques investigate the best method for producing speaker labelled segments. Initially, experiments use the counting technique described in Section 5.3.2 to label the frames with a channel.

Training DNNs with and without overlap data

The first experiment investigates including overlapping speech in the training data and results are displayed in Table 5.10. The DNN model *CAT4* is trained without overlapping speech and *CAT4+OV* is trained with overlap. The results are similar, with the DNN including overlap achieving DER of 8.2%, lower by only 0.1%. Labelling overlap as speech implies detecting more overlap regions as speech. Both DNNs detect a similar amount of speech time at 315.3 minutes for model *CAT4* and 314.3 minutes for *CAT4+OV*, but the latter detects 537 fewer segments.

Training DNNs with crosstalk features

The results in Table 5.8 show how training SAD DNNs on crosstalk features improves performance. This experiment trains DNNs on the same training data as before, however, they are trained on crosstalk features as well as filterbanks. This aims to show whether performance is also improved for these DNNs. Both DNNs perform worse than their counterparts seen in Table 5.10. The model *CAT4+CT* detects fewer than half of the number of segments that *CAT4* detected. The DER has risen by 1.4% for *CAT4* and by 0.2% for

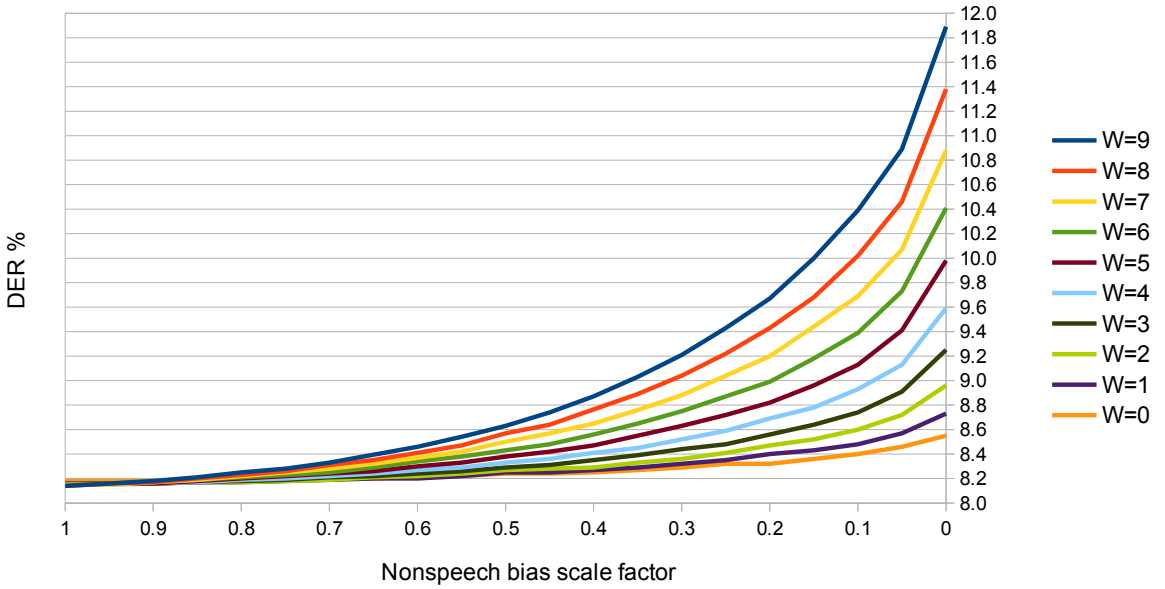

Fig. 5.14 Results for Method 3 on TBL when applying different context windows of length w. The nonspeech bias is introduced to prevent too much nonspeech from being detected.

CAT4+OV. This shows that additional crosstalk features do not help this method unlike previously seen for SAD. This could be because these DNNs are trained for a different task than for SAD.

Applying different frame decision techniques

The best DER of 8.2% is seen in Table 5.10 for DNN *CAT4+OV*. The previous experiments counted how many occurrences of each channel for each frame. The labelling decision is made on the highest occurring speaker channel. As opposed to this, posterior probabilities calculated from the decoded output are calculated for each channel. The channel with the highest score can be selected. Furthermore, a bias against nonspeech can be introduced to help reduce the amount of missed speech which contributed the most error to the results. Scaling the nonspeech counts, or posterior probabilities, by a percentage would cause speaker channels to be selected instead of nonspeech. Lastly, a window of w number of frames around the target frame is added to consider the context around the frame. Figure 5.14 displays the results for the counting technique biased against nonspeech at various scales. Windows with contexts from 0 to 9 frames either side the current frame are investigated, where a window of 0 frames represents no window. A nonspeech bias of 100% means the nonspeech counts are the same as the channel counts, while 0% is when nonspeech counts equal 0, which is equivalent to no nonspeech detected. The graph shows how decreasing the scale of the nonspeech bias increases errors. This means that treating nonspeech as being equivalent to a

Table 5.12 Best results of Method 3 for TBL comparing posterior probabilities with simple counting, varying the bias of nonspeech and changing the length of the context window.

Decision	NS bias s.f.	w	Seg%	MS%	FA%	SE%	DER%
posteriors	1	0	76.6	4.3	2.5	1.4	8.14
	1	2	76.6	4.4	2.4	1.3	8.11
	1	3	76.6	4.4	2.4	1.4	8.11
	1	4	76.6	4.4	2.4	1.4	8.11
	0.95	0	76.6	4.4	2.4	1.4	8.11
	0.95	3	76.6	4.4	2.4	1.4	8.11
	0.95	4	76.5	4.4	2.4	1.4	8.11
counts	1	0	75.6	4.3	2.6	1.3	8.18
	1	8	75.5	4.3	2.6	1.3	8.14
	1	9	75.5	4.3	2.6	1.3	8.14

speaker is beneficial. The graph also shows how minimal the impact varying the window size is when the nonspeech bias is high, and a negative impact when the nonspeech bias is low. A nonspeech bias of 20% results in a DER of 9.7% for a window size of 9 frames and 8.4% for 1 frame.

Table 5.12 summarises the parameters which provide the best performance for both the posterior and count techniques and which nonspeech bias and window size. The DER is displayed to a higher precision than usual to show the differences. The results show that posterior probabilities are more suitable for channel decision than the simple occurrence counts. However, it is very small difference, at 0.03%. Windows of size 2, 3 and 4 achieve better performance, a gain of 0.03%, than no context. For the counting technique, a gain of 0.04% with windows of size 8 and 9 compared to no context is seen. These performance gains are minimal. For this method, the TBL data does not benefit from a bias against nonspeech or applying a window.

5.5.3.5 Experiments: Method 4

Method 4 is described in Section 5.3.3 and is an extension to Method 3. Instead of concatenating all the IHM features together, pairs of features are concatenated instead. This removes the need for requiring every recording to have a fixed number of speaker channels. Therefore, both the TBL and RT07 datasets are applicable and DNNs are trained on TBLTRAIN-IHM data unless otherwise stated. Experiments, similar to the previous method, compare training the DNNs with or without overlapping speech and with or without additional crosstalk features. An extra experiment considers different training data. For

Table 5.13 Results for Method 4 when training on overlap for both TBL and RT07. Two different training setups are compared, training DNNs with (+OV) and without overlap. The metrics MS, FA, SE and DER are percentages.

DNN	TBL					RT07				
	Seg%	MS	FA	SE	DER	Seg%	MS	FA	SE	DER
CAT2	94.4	17.0	1.4	1.0	19.4	75.6	56.5	1.2	0.4	58.2
CAT2+OV	94.8	20.3	1.1	0.9	22.4	71.6	60.9	0.8	0.4	62.1

Table 5.14 Results for Method 4 when training on crosstalk features (+CT) additionally to filterbanks for both TBL and RT07. The metrics MS, FA, SE and DER are percentages.

DNN	TBL					RT07				
	Seg%	MS	FA	SE	DER	Seg%	MS	FA	SE	DER
CAT2+CT	90.7	7.7	0.9	1.2	10.9	53.8	59.7	1.3	0.2	61.2
CAT2+OV+CT	120.6	34.8	0.7	1.1	36.5	37.4	79.6	0.4	0.1	80.1

labelling the frames, the counting technique described in Section 5.3.2 is applied, with further experiments investigating posterior probabilities, a nonspeech bias and a context window.

Training DNNs with and without overlap data

The first experiment investigates whether training on overlapping speech improves performance, and results are seen in Table 5.13. Method 3 saw an improvement when training on overlapping speech, however, this improvement is not seen in Method 4. For both TBL and RT07 data, the majority of the error is MS. The best result for TBL is 19.4% DER which is better than the baseline but worse than the previous methods using IHM speaker channels, Method 2 and Method 3. For RT07, the results are poor and do not improve on the baseline. There is a clear issue with detecting too much nonspeech.

Training DNNs with crosstalk features

Training the DNNs with filterbanks and crosstalk features is investigated to see whether these additional features improve performance and specifically whether the amount of missed speech can be reduced. Table 5.14 displays the results. For TBL data, adding crosstalk features to *CAT2* and *CAT2+OV* gives different results. The model *CAT2+CT* detects fewer segments and less missed speech. The DER greatly improves from 19.4% to 10.9%. This is much closer to the best performance of 8.2% in Method 3. Method 3 should outperform Method 4 as it is given more data to train on at the sacrifice of portability to other datasets. However, *CAT2+OV+CT* detects more segments, larger MS and the DER increases from

Table 5.15 Results for Method 4 when training on AMI-IHM data for both TBL and RT07. Two DNNs are trained, one on filterbank features and the second trained with both filterbanks and crosstalk features (+CT). The metrics MS, FA, SE and DER are percentages.

DNN	TBL					RT07				
	Seg%	MS	FA	SE	DER	Seg%	MS	FA	SE	DER
CAT2A	118.3	16.6	1.0	4.9	22.5	78.9	58.9	0.5	0.1	59.5
CAT2A+CT	87.8	22.9	0.9	5.0	28.8	61.7	62.4	0.5	0.1	63.0

22.4% to 36.5%. For RT07, similar results are seen in both DNNs, additional crosstalk features leads to higher MS and higher overall DER. However generally fewer segments are detected.

Training on AMI data

The previous DNNs have been trained on TBLTRAIN-IHM data only. This is apt for TBL and it was shown in Table 5.7 that for SAD, training on TBLTRAIN-SDM resulted in better performance than training on the same domain, AMI-IHM data. For RT07, training on AMI-IHM data may be more beneficial as they belong to the same domain. A DNN trained on AMI-IHM data is investigated as well as a comparative DNN trained with additional crosstalk features. Results are seen in Table 5.15. The model *CAT2A*, trained on AMI-IHM as opposed to TBLTRAIN-IHM data, performs worse than *CAT2* for both TBL and RT07 data. For TBL, *CAT2* achieves 19.4% while *CAT2A* achieves 22.5%, a rise of 2.1% absolute. RT07 results increase from 58.2% to 59.5% with *CAT2A*. Again, with the exception of *CAT2+CT*, crosstalk features do not lead to performance improvements. Although AMI-IHM is the same domain as RT07, meeting data, it has not provided a performance gain.

Applying different frame decision techniques

The best performance for TBL was seen with DNN *CAT2+CT* whereas the lowest DER for RT07 was provided by *CAT2*. As the amount of missed speech contributes the most to the DERs, a larger effect is seen in RT07, further experiments into applying a bias against nonspeech and adding a context window to help the frame decision improve are performed. As in Method 3, this is investigated in terms of the occurrence counting and posterior probabilities for the frame decision technique. Graphs are displayed for the counting technique but not the posteriors as the results are similar. Figure 5.15 (A) displays the performance for TBL data. It is clear that applying a nonspeech bias improves the performance, which was not seen in Method 3. For a nonpseech bias between 35% and 10%, the best performance for each window size, w, is seen. The larger the window, the worse the

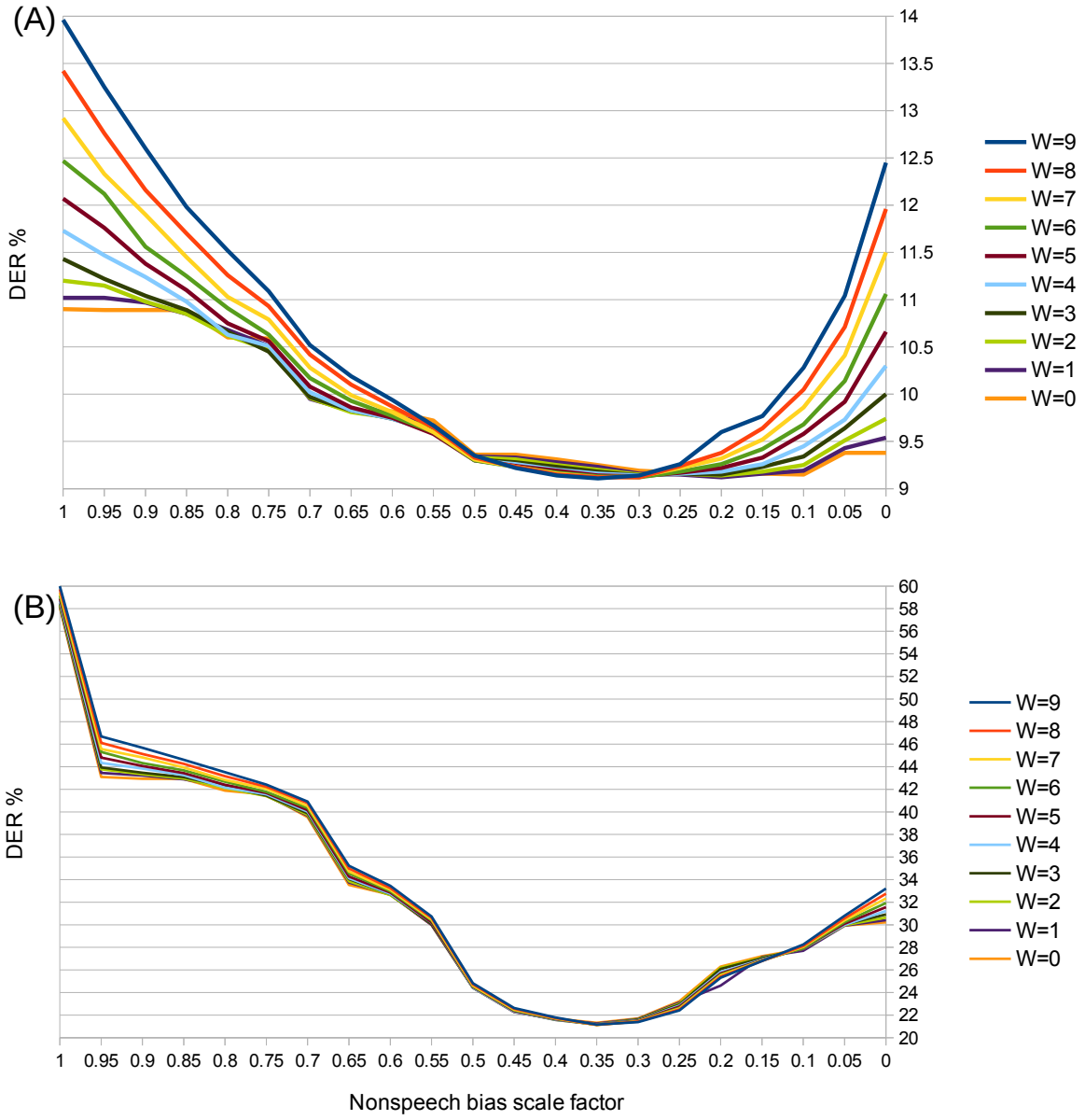

Fig. 5.15 Performance for Method 4 with different context windows, w. The nonspeech bias is varied for (A) TBL and (B) RT07.

performance with a high or very low nonspeech bias. This shows how the window sizes are not that distinguishable when the nonspeech bias that gives the best performance is applied.

Figure 5.15 (B) displays the results for the same previous experiments with RT07 data instead. A similar trend is seen in which the nonspeech bias greatly helps improve performance. The optimum nonspeech bias is 35% for all window sizes, which was previously best for the larger windows for TBL data. The gain in improvement is much higher. For

Table 5.16 Best results for Method 4 on both TBL and RT07. The posterior probabilities are compared with the counting technique. The nonspeech bias and the context window, w, are varied to produce the best performance.

Data	Decision	NS bias s.f.	*w*	Seg%	MS%	FA%	SE%	DER%
TBL	posteriors	0.25	0	92.0	4.8	2.6	2.0	9.42
	posteriors	0.25	1	91.5	4.8	2.6	2.0	9.41
	posteriors	0.25	2	91.2	4.8	2.6	2.0	9.41
	posteriors	0.25	3	91.0	4.8	2.6	2.0	9.41
	posteriors	0.25	4	90.8	4.8	2.7	2.0	9.41
	counts	0.35	0	85.6	5.3	2.4	1.6	9.25
	counts	0.35	9	82.8	5.2	2.5	1.5	9.11
RT07	posteriors	0.35	0	129.4	15.7	6.1	2.3	24.12
	posteriors	0.35	9	124.0	15.9	5.8	2.1	23.80
	counts	0.35	0	107.3	16.4	3.8	1.1	21.32
	counts	0.35	7	103.6	16.2	3.8	1.1	21.13

no window, the DER improves from 58.2% with no bias to 21.3% at a 35% bias. This is a huge drop of 36.9% absolute DER. The same trend in the window sizes is seen as before. Higher window sizes cause lower performance for high and low biases, whereas the DERs are similar across windows at the best nonspeech bias, varying from 21.3% with no bias to 21.5% with no nonspeech allowed.

Table 5.16 summarises the best performance achieved when varying the nonspeech bias and the context window size. The best performance for posterior probabilities as well as counts are included, however for this method occurrence counts consistently outperform posterior probabilities. The equivalent results with no context window applied are included for comparison. Adding a bias against nonspeech reduces the amount of MS detected which is caused by selecting too much nonspeech. For TBL, the reduction is from 7.7% to 5.2% and for RT07, the reduction is 40% absolute, from 56.2% to 16.2% MS. The best performing bias for both datasets is 35%. The posterior probabilities are an alternative to counting occurrences and were investigated as they hold more information than simple counts. This was expected to lead to an improved decision making step. However, they do not show any improvement. Various context windows were applied to again improve the frame decision step. The best performance for RT07 was achieved when applying a context of 7 frames either side of the current frame. This improved the result by 0.19% when compared to no context applied. The gain for TBL is 0.14%. However, these are small gains. The windows help to reduce the number of segments. Making a decision without context can lead to many short segments being labelled differently. The windows reduce this and act as a smoothing technique.

5.5.3.6 Experiments: Method 5

Method 5 is the final proposed method. It focusses on combining both timing and acoustic information which is in the form of transcripts and IHM speaker channels. TBL and RT07 datasets are used for evaluation of the method. However, as a transcript is not available for RT07, the output of an ASR system is used instead (more details in Section 5.5.1 and presented by Hasan et al. (2015)). The initial SAD segments are taken from the models providing the lowest DER in Method 2. This was determined to be *SAD3* for TBL and model *SAD6* for RT07, as seen in Table 5.6. Speaker labels and further speaker boundaries are inferred from the speaker labelled segments output in the transcript alignment method. The remaining 'unknown' segments, those that the transcript does not provide a speaker for, are labelled by channel either by considering the energy or posterior probabilities for each IHM channel. Experiments investigate whether inferring speaker labels and boundaries from the transcript alignment helps performance. For the second speaker labelling step, using the energy and the posterior probabilities are compared. This is compared to leaving the 'unknown' labels in, as if they belonged to a separate speaker, to show the gains in performance when labels are determined. Further to this, the energy or posteriors labelling can either label the whole 'unknown' segment with one channel, or detect a channel for every frame. The channel labelling can happen directly on the SAD, as well as the combined SAD and transcript segmentation, so this too is investigated.

Labelling using energy across the IHM features

Firstly, the channel labelling technique using energy is investigated and Figure 5.16 displays the results. The three techniques to speaker labelling using the energy are shown: unknown, frame and segment. The figure shows that giving the 'unknown' labelled segments a channel, be it by frame or by segment, results in a large gain in performance as this is a reduction in one whole speaker. Using SAD+transcript segments for TBL with an 'unknown' speaker gives good performance, opposed to the others, because the transcript segmentation is well matched to the SAD output. Labelling by segment shows a gain in performance, except for RT07 using SAD+transcript in which the frame technique improves by 0.1%. Using the information in the transcripts does help improve performance, by 4.8% for RT07 and by 8.1% absolute for TBL data. RT07 achieves the best result of 25.3% labelling by frame and TBL achieves 11.3% DER labelling by segment.

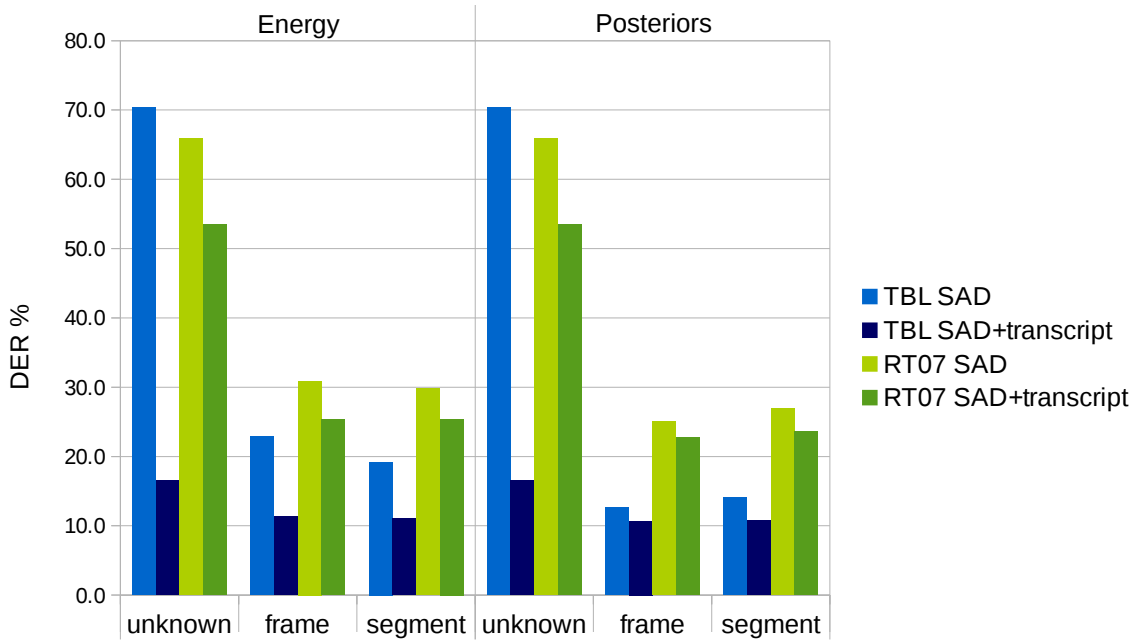

Fig. 5.16 Method 5 performance comparing energy and posterior probabilities for speaker labelling. Three segmentations are evaluated: leaving the unlabelled segments as is, 'unk', making a frame-by-frame decision, and making a single decision for each segment.

Labelling using posterior probabilities from aligned IHMs

Figure 5.16 also shows the same experiments when the posterior probabilities across the IHM channels are considered instead of the energy. The experiments using SAD without transcripts, are equivalent to the results in Method 2. The 'unknown' results are the same as previously seen. Speaker labelling these segments using posterior probabilities improves the performance over using the energy as the DERs are all lower. Again, the TBL result when combining SAD with transcripts is already a low score when an extra 'unknown' speaker is included. Unlike the energy technique, using posterior probabilities to decide by frame achieves better performance than deciding by segment. The gains range between 1.8% and 0.1% absolute, with larger gains achieved when using the SAD without the transcript.

It is clear to see that combining SAD with the transcripts improves diarisation performance. This shows that auxiliary information of speaker labels and boundaries in the form of transcripts, despite them not being complete and lacking timing information, still outperforms not using them. Making a further channel decision based on posterior probabilities consistently outperforms the energy technique, for both datasets. This shows that the energy across the IHM channels is not as reliable as the posterior probabilities. The energy is corrupted by crosstalk and louder speakers close to the microphones. Finally, for the posterior probabilities

Table 5.17 Best performance of Method 5 for TBL and RT07 which was achieved by combining SAD segmentation with transcript segmentation and speaker labels and speaker, or channel, labelling each frame given the maximum posterior probabilities for each frame.

Data	Speaker Labels	Seg%	MS%	FA%	SE%	DER%
TBL	SAD+transcript+posteriors (frame)	97.2	4.5	2.0	4.2	10.7
RT07	SAD+transcript+posteriors (frame)	93.1	10.7	6.4	5.7	22.8

it is clear that making the decision frame-by-frame gives better performance. The overall best performance for both datasets is seen in Table 5.17. TBL achieves 10.7% and RT07 reaches 22.8% DER.

5.5.3.7 Experiments: SEGF and other metrics

The best results from each method are re-evaluated using different metrics. The segment-based metric proposed in Chapter 4, SEGF, is applied as well as the DPC, BNDF and the purity measure K. The DER is the established error metric for speaker diarisation but the alternatives give deep insight into the errors for each method. Table 5.18 displays the results, where the SEGF has a collar of 0.1 seconds. The SHoUT results are included as a baseline.

For TBL performance, the DPC results are either the same or better than SHoUT except for Method 1. Method 1 uses transcript alignment and fewer segments are detected, 3371. This segmentation has increased the average time distance between the matched reference and hypothesis boundaries. This has also affected the BNDF, which has the poorest score of the five methods. Four of the five methods achieve a higher purity score K than the baseline. Method 2 does not and this decides on speaker labels by considering the posterior probabilities across IHM channels. Method 5 also uses this technique, as well as taking speaker labels from the transcripts. This shows how the posterior probabilities are not ideal but have been improved by the transcript labels in Method 5, which shows a small gain over SHoUT of 1.2%. Method 1 only relies on the transcript speaker labels and this has an improvement of 3.2%. For the SEGF, the SHoUT performance, and Method 1, are very poor. This is reflected to some extent in the BNDF scores. Method 5 achieves the highest score by segmenting using a combination of SAD, transcript alignment and posterior probabilities across channels. Again, the lack of correlation between the DER and SEGF metrics is seen which shows the need for segmentation evaluation, which the DER does not do.

The RT07 scores are also included in Table 5.18. In terms of the DPC, the methods beat SHoUT except for Method 2 which is worse by 0.1 ms. The results are fairly similar, unlike the large difference seen for TBL with Method 1. However, Method 1 is not applicable to RT07 due to lacking transcripts. All the methods achieve at least double the BNDF score

Table 5.18 The overall best performance for each method is presented on both TBL and RT07 test sets. The DER is compared to the boundary measure, DPC and BNDF, the purity measure K and the SEGF proposed in Chapter 4. The DPC is measured in milliseconds.

Data	System	Seg%	Spk%	DER%	DPC	BNDF%	K%	SEGF%
TBL	SHoUT	85.7	172.5	25.8	0.7	26.3	83.2	0.7
	Method 1	38.5	100.0	21.7	3.0	43.1	86.4	1.5
	Method 2	99.5	100.0	12.5	0.6	71.0	81.0	54.3
	Method 3	76.6	100.0	8.1	0.6	73.6	89.5	52.4
	Method 4	85.4	100.0	9.2	0.7	69.1	89.3	42.2
	Method 5	94.5	100.0	10.7	0.5	74.0	84.4	57.3
RT07	SHoUT	75.6	117.1	49.5	0.8	30.0	71.3	0.5
	Method 2	88.2	100.0	25.1	0.9	60.9	70.9	0.5
	Method 4	103.6	100.0	21.1	0.6	60.3	81.2	32.3
	Method 5	93.1	100.0	22.8	0.7	64.6	74.5	31.6

of SHoUT with Method 5 having the highest result. For K, Method 2 does not outperform SHoUT, and this was the same for TBL data. It performs worse by 0.4%. Method 4 achieves the best result at 9.9% better than SHoUT. The SHoUT performance in terms of SEGF is also poor, and Method 2 achieves the same poor segmentation performance. Method 4 and 5 greatly increase this to the low thirties, with Method 5 achieving the highest by only 0.7%. Between Method 2 and Method 5, the difference is 31.1% for the SEGF and 2.3% for the DER. This again shows how the SEGF can give a more detailed understanding of the segmentation performance.

Across both datasets, the BNDF is consistent showing that Method 5 achieves the best performance for both TBL and RT07. In terms of the purity measure, Method 4 is best for RT07 whereas it is second best to Method 3 for TBL, which outperforms by 0.4%. For the SEGF scoring, both datasets achieve the highest scores on Method 5. This reinforces the notion that the DER masks performance and combining results form different metrics gives detailed information which the DER misses.

5.5.4 Discussion

Experiments have investigated five different methods based on different forms of auxiliary data. The aim has been to ascertain the benefits of using transcripts and IHMs in a diarisation system to improve the performance over an unsupervised system. The supplementary data is shown to give improvements to different degrees, despite both containing imperfections. Each method shows the best performance for both datasets in the results. Figure 5.17 displays these in terms of the error contributions: MS, FA and SE.

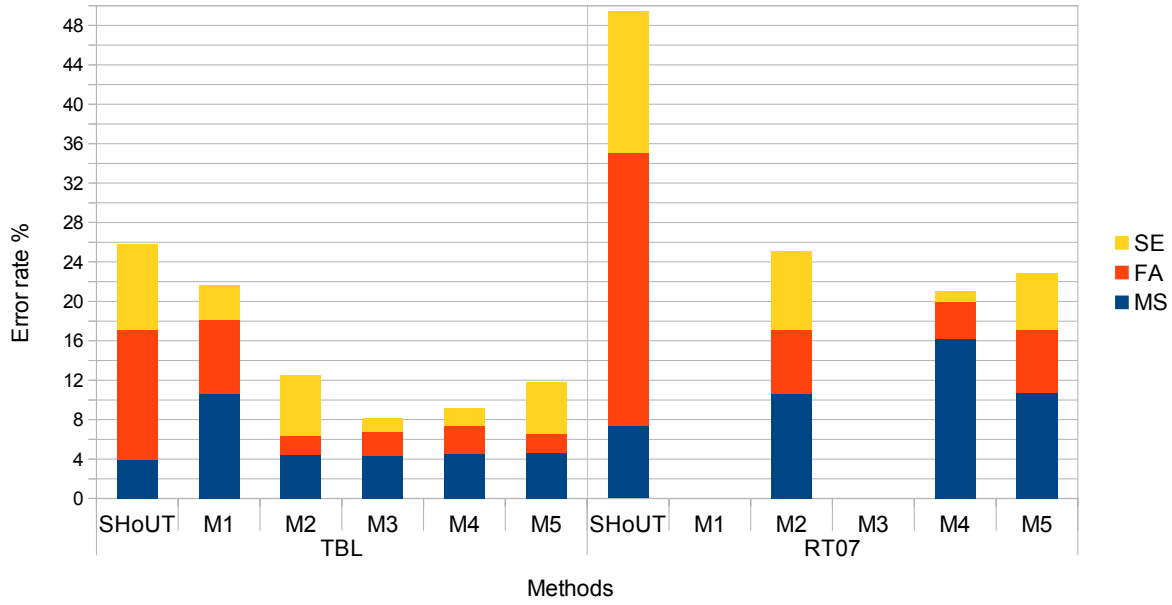

Fig. 5.17 Overall baseline and method results for TBL and RT07, in which the DER is displayed broken down into the sum of missed speech (MS), false alarm (FA) and speaker error (SE).

The transcript alignment in Method 1 was performed for TBL data only. This timing information data contained words with speaker labels with limited time information. The method was able to achieve a reasonable performance gain over the baseline. Every five minute segment was successfully aligned and a resegmentation process improved the segmentation. Speaker error was greatly reduced thanks to the speaker labelling information provided.

Three methods involved IHM channels and the first also involved SDM channels. This step-by-step method trains DNN-based SAD models, then a channel label for every frame is determined from the posterior probabilities calculated when aligning the SDM SAD segments to the IHM channels. The TBL result is 9.2% absolute, 43% relative, better than the transcript alignment technique. This implies that using crosstalk-heavy speaker channels result in better performance than using inaccurate transcripts. Methods 3 and 4 are different as they do not use SDM channels and are integrated approaches The latter is an extension to the former which is portable to different datasets. This comes at a sacrifice in performance as seen for TBL data. Training on more data and all the concatenated features gives more information to the DNN resulting in better performance. Training on IHM channels without SDM gives yet further improvements as the latter seems to hinder performance.

The last method involves transcripts as well as both SDM and IHM channels. Including the IHM channels as an input has reduced the MS and FA when compared to Method 1 for

the TBL data. For both TBL and RT07, it outperforms Method 2 and the SE is similar, but it does not outperform Method 3 or Method 4. This shows that the transcripts do help to improve performance. But, both the transcripts and the SDM channels hinder the maximum performance which is gained from using the IHM channels alone. A further method would be necessary to perform diarisation with transcripts and IHM channels to confirm how much of the performance gain is from each auxiliary information source.

The three IHM channel methods all considered crosstalk features as a way to reduce the negative impact. Method 2 trained comparative DNN-based SAD models with and without crosstalk features. They could be tested against IHM channels only and it was seen that the segmentation improved for every DNN trained on crosstalk features. However, the models did not improve performance in Method 3 which trained DNNs on concatenated IHM channel features. Method 4 trained DNNs similarly to Method 3, however a mixed performance was seen. The TBL data benefited from crosstalk features whereas RT07 did not. Methods 3 and 4 train models for diarisation whereas Method 2 trains DNNs for SAD. These are two different tasks which could be the reason for the different results.

Posterior probabilities are used in Methods 2, 3, 4 and 5, however in Method 2 they are not compared to an alternative until Section 5.5.3.6. Method 3 and 4 uses posterior probabilities versus channel occurrence counting and Method 5 uses posterior probabilities against energy detected on the IHM channels. In Method 3 it was shown that posterior probabilities outperform counts by a very small margin. However, this does not occur in Method 4, which shows better performance using the simple counting technique. Finally, experiments for Method 5 show gains in performance over using the energy instead. These IHMs contain crosstalk which can negatively impact the energy levels.

For both datasets, all the proposed methods improve on the baseline public domain toolkit. The best results are based on using IHM channels only. For RT07, Method 4 achieves the best performance in which experiments were performed to reduce the amount of nonspeech detected with great success, although MS is still the predominant error. For TBL, Method 4 is the second best performing against Method 3. This is because it is an extension to Method 3 in which the technique is applicable to all datasets but at a cost of performance. These methods have shown that performance is gained when using auxiliary data in a semi-supervised diarisation system. The auxiliary data applied, transcripts and IHM channels, contain imperfections such as missing sections, rough time stamps and crosstalk. Despite these issues, the proposed methods all achieve better performance over the baseline.

Lastly, as mentioned in Section 3.5.1, RT07 can be scored in two ways. The experiments in this chapter have been evaluated using the SHEF method. To be comparable to other publications and research, the NIST method is applied to the best results and displayed in

Table 5.19 Overall RT07 results comparing the two scoring setups: SHEF (manual reference and collar of 0.05s) and NIST (specific portions of the evaluation reference with a collar of 0.25s).

System	RT07 (SHEF)				RT07 (NIST)			
	MS%	FA%	SE%	DER%	MS%	FA%	SE%	DER%
SHoUT	7.4	27.7	14.4	49.5	3.9	9.2	12.3	25.3
Method 2	10.6	6.5	8.0	25.1	8.8	1.4	6.2	16.4
Method 4	16.2	3.8	1.1	21.1	14.2	0.9	1.2	16.3
Method 5	10.7	6.4	5.7	22.8	8.9	1.4	1.1	11.4

Table 5.19. The more lenient collar and not scoring the entire recordings in the NIST setup greatly affects the performance using SHoUT by almost a 50% reduction. The scores reduce in all cases for the three methods, except for the SE rate in Method 4, which has a small rise from 1.1% to 1.2%. This is due to the reduction in MS and FA. It is interesting to see how Method 2 and Method 4 give similar scores for the NIST setup whereas a difference in performance, of 4% absolute DER is seen with the SHEF setup. This reinforces the argument that the NIST scoring hides the true performance. Furthermore, the SHEF scoring shows Method 4 to have the best performance, however, this is Method 5 for the NIST setup. This is an effect of the collar in the DER. The collar around the reference boundaries removes those portions of time from being evaluated upon. SHEF uses a collar of 0.05s which results in 14412.37 seconds of scored speaker time in the reference, whereas NIST uses a collar of 0.25s which gives 5734.73 seconds of scored speaker time. This is half the amount of the SHEF scoring which causes a very different scoring result with fewer possibilities to make mistakes.

5.6 Summary

Many see speaker diarisation as an unsupervised task, however research into semi-supervised and supervised methods exist. Moraru et al. (2004a) applied speaker information as auxiliary data and performance gains were seen using one or more speaker models. Aronowitz (2011) derives features from acoustic information and improvements were seen across evaluations.

For this chapter, two different types of auxiliary information where considered: timing information and acoustic information. Transcripts were presented as a form of timing information. Transcripts range in quality as seen in Figure 5.4 and Figure 5.5. The quality of the transcripts provided for TBL data was more similar to the latter, in which speaker labelled words existed but timing information only appeared every five minutes. The acoustic

information was in the form of IHM channels, as an addition to SDMs. However, crosstalk occurs across both datasets.

Methods were proposed based on different combinations of transcripts, SDM channels and IHM channels. It was shown that all combinations improved over the baseline. Different degrees of improvements are seen. Transcripts give the least improvement. Combining both channel types performs better, however, the greatest gains are seen when using IHM channels by themselves. A combination of all three outperforms all but the IHM channel methods. Crosstalk features were introduced and greatly improved IHM performance for SAD segmentation in Method 2. However, mixed results were seen in Method 4 and no improvement seen in Method 3. Posterior probabilities thought to contain deeper information were compared to other techniques with mixed results. Improvements over energy as a channel labelling technique was seen whereas Method 4 worked best using simple counts instead.

This chapter has shown that a diarisation system can be enhanced with auxiliary information to give performance gains over unsupervised systems. The semi-supervised methods used imperfect data and still gave improvements. For TBL and RT07 datasets, it was shown that the acoustic information methods could only make the best use of the IHM channels when the nonspeech detection is controlled.

Chapter 6

DNN-based Speaker Clustering

Contents

DNNs have grown in popularity within the field of speech technology in the last decade. They have successfully been incorporated into ASR with great improvements (Hinton et al., 2012). However, research is still necessary to successfully apply DNNs to all stages of a speaker diarisation system. In terms of feature processing, DNN-based features have shown promise (Yella and Stolcke, 2015). Features were extracted from the bottleneck layer of a pretrained ssDNN which learnt how to classify, or separate, speakers. As seen in Chapter 5,

DNN-based SAD models are successful for speech/nonspeech detection (Dines et al., 2006). Speaker segmentation using a windowing method (see Section 2.5) was presented using AANN models to decide whether both windows belong to the same speaker (Jothilakshmi et al., 2009). This research also presented a clustering method in which each segment was represented by an AANN model and confidences for merging were calculated for every pair. Chapter 5 showed success in segmentation and clustering with DNNs trained on concatenated channel features. However, the methods presented required datasets with IHM channels.

In Chapter 5, a third type of auxiliary information was discussed in Table 5.1 on page 90. The supplementary data for speaker information is pretrained speaker models. Speaker models are applied to the different types of learning: unsupervised, semi-supervised and supervised (described in Section 2.2). The state-of-the-art speaker clustering method creates clusters given the speaker-pure segments. These clusters are used to resegment the data after every pair-wise merge of clusters, such as the ICSI-RT07 system described in Section 2.8. The clusters are GMMs which are not trained on external data, and the method is unsupervised. Using speaker models of the participants in a supervised fashion is not typical, however, Moraru et al. (2004a) has shown performance gains given models trained on enough data. These models were applied in the clustering stage of their system. In the middle ground, speaker models have been applied using ssDNNs as previously described in semi-supervised methods (Yella and Stolcke, 2015). These models are trained on unknown speakers and features are extracted.

The chapter is organised as follows. The related research and motivation for the work are discussed in Section 6.1. The proposed DNN-based clustering method consists of several stages and is presented in Section 6.2 with the task of training a ssDNN is detailed in Section 6.2.1. The experiments are shown in Section 6.3 and the chapter is summarised in Section 6.4. This chapter investigates Objective 3, as discussed in Section 1.4, and is supported by a publication (Milner and Hain, 2016a).

6.1 Related Research

Early work on training ssDNNs was discussed by Konig et al. (1998). A Multilayer Perceptron (MLP) was trained to classify speakers and applied in the speaker verification field. This network aimed to nonlinearly project acoustic features to a lower-dimensional set with the idea of maximising speaker separation. A Nonlinear Discriminant Analysis (NLDA) technique was applied in which a 5 layer MLP was trained where the output target classes were speakers. There were three hidden layers of which the second was a bottleneck layer. To extract the features from this bottleneck layer, the third hidden layer and final layer were

removed and the remaining three layers were used to project the speaker data. A speaker verification system was trained on these extracted features. Switchboard data was used for training whereas the 1997 NIST Speaker Recognition Evaluation corpus was used for testing. When the NLDA features system were combined with the cepstrum based system, around 15% relative improvement was seen, however, using further development data could see higher gains.

For ASR, Liu et al. (2014) investigated using an ssDNN for extracting features in the field of far field speech recognition. A DNN was trained with a bottleneck layer and the final outputs were monophone or triphone states. The bottleneck layer of size 13 gave the best performance and features were extracted from the bottleneck layer to be used in an HMM/GMM single pass retraining method for ASR. The research compares PLPs with PLPs combined with bottleneck features extracted from differently trained DNNs. The latter gave the better performance with an average 25% relative Word Error Rate (WER) reduction. Next, different inputs to the DNN were investigated. Log filterbank features were extracted from a beamformed MDM channel and from concatenated channels. The concatenation method performs similar or better when concatenating either 2 or 4 microphones. Finally, Liu et al. included additional speaker information for DNN adaptation. An ssDNN was trained where the target classes were speakers and bottleneck features were generated. These features are said to provide a projection similar to i-vectors. Performance improvements were seen and benefited from the DNN trained on multiple concatenated channel features.

For feature processing optimised for diarisation, NNs were trained to learn a feature transform (Yella et al., 2014). For pairs of speech segments, an Artificial Neural Network (ANN) was trained to decide whether they belong to the same or different speakers. A separate input and bottleneck layer were trained for each segment before the weights were tied in the second hidden layer which connects each half of the first hidden layer, the two bottlenecks, to the final layer. Features were extracted from the bottleneck layer activations of the model and used either with or without traditional MFCCs in a standard HMM-GMM framework based on the ICSI-RT07 system (Wooters and Huijbregts, 2007). It was argued that using hidden-layer activations as features made the task easier as the initial ANN layers transformed the input features into a space which was more conducive to speaker discrimination. AMI and ICSI data was used for training and testing whereas NIST RT data was used for testing only. The network consisted of the two input and two bottleneck layers, followed by a single hidden layer before the output layer. It was shown that these ANNs do not outperform MFCCs, but when combined, relative gains of 11-14% in speaker error are seen. Follow up research by Yella and Stolcke (2015) investigated further ANN architectures for generating features. The previous network was extended to a deeper network in which

an extra hidden layer was included. This layer was added before the bottleneck layer from which the features were extracted. Secondly, a speaker classification ANN was investigated based on the research by Konig et al. (1998). The target classes were speakers and the hidden layer activations from the second hidden layer, the bottleneck, were used as features. Lastly, an ANN autoencoder was trained which encodes the input in a representation which was then used to reconstruct the input, meaning the output targets were the inputs. Again, the features were derived from the second hidden layer. AMI and ICSI data was also used for the training and testing and NIST RT data was used for testing only. Initially the reference segments were used along with a final SAD output for NIST RT data only. All the test sets show improvements when combining MFCCs with the four types of features.

In terms of speaker segmentation and speaker clustering, AANN models have been applied which did not use ssDNNs (Jothilakshmi et al., 2009). The AANN consisted of several hidden layers and the desired output was the same as the input vector meaning the size of the input and output layers was the same. A five layer model was trained as it is able to cluster the input data in the nonlinear subspace, as opposed to linear. The speaker segmentation stage used a windowing method such as those described in Section 2.5. An AANN was trained on the left hand side of a window and features on the right hand side were used for testing the model. If the regions of the window belong to different speakers, the confidence score should be low and if the regions belong to the same speaker then the score should be high. A threshold was used to decide on where the speaker boundary falls. The clustering method trained an AANN for each segment and features from the other segments were used for testing. This gave a confidence score for pairing each segment with every other segment and results in terms of distances were computed. Clustering was performed on these scores. A collection of broadcast news data was used for evaluation named AUdata, as well as RT03S test set. AUdata achieves 11.7% and RT03S achieves 12.0% DER, 0.9% less than results for the CLIPS and LIA system (Meignier et al., 2006).

An ssDNN was first presented for speaker verification by Konig et al. (1998) in which features were extracted from the second hidden layer. The research by Liu et al. (2014) extended this by extracting features for ASR and Yella et al. (2014) and Yella and Stolcke (2015) extracted features for diarisation. Yella and Stolcke saw improvements when these features were combined with MFCCs in a diarisation system. An unsupervised speaker segmentation and clustering method was presented by Jothilakshmi et al. (2009) which used AANNs but not ssDNNs.

6.1.1 Motivation

DNNs have been successful in various stages of a speaker diarisation system: feature processing (Yella and Stolcke, 2015), SAD (Section 5.3.1), segmentation and clustering (Jothilakshmi et al., 2009, Sections 5.3.2, 5.3.3). Jothilakshmi et al.'s segmentation and clustering method was unsupervised, whereas the DNN-based methods in Chapter 5 required supplementary data in the form of IHM channels. It has been previously shown in this thesis that semi-supervised methods using timing and acoustic auxiliary information lead to improvements in performance. Speaker information was not investigated. In the field of diarisation, speaker models are the supplementary data for speaker information. Moraru et al. (2004a) investigated using speaker models for specific participants in the recordings. As the expected speakers were known, using this data to train models led to the supervised methods. A semi-supervised method using speaker models requires models pretrained on data not occurring in the evaluation. A DNN-based approach to segmentation and clustering incorporating speaker information is investigated. Yella and Stolcke (2015) and Liu et al. (2014) used an ssDNN to extract features from the bottleneck layer. It is assumed that the bottleneck layer has captured the information that is learnt on how to classify speakers. The presented method aims to take this information from an ssDNN to directly perform diarisation.

6.2 Semi-supervised Speaker Clustering with DNNs

The method presented behaves like an AHC, or bottom-up, technique. The key element of this semi-supervised method is the pretrained ssDNN containing a bottleneck layer which captures how to classify speakers. DNNs are discriminant classifiers and by nature do not cluster. However, the method presented reconstructs a DNN from the pretrained ssDNN. This new DNN is trained on speaker labelled segments and decoded on unlabelled speech segments which leads to improvements. If the segments contain speakers with little data, those speakers are less likely to be detected as the DNN is trained on a small amount of data. Furthermore, speakers that are not detected in the decoding stage can never be recovered, as the data is reassigned to different clusters. In this way the method acts as if it is clustering.

Two inputs are required for the clustering method to work: speech segments and an ssDNN. The input segments to any clustering method for diarisation usually aim to be speaker-pure segments. However, the method clusters and resegments the input. This allows for the segments to be speech-only (from SAD) or speaker-pure (from speaker segmentation). The second input required is an ssDNN which has been previously trained on external speaker labelled data. The method comprises of five steps and the iterative process is seen in

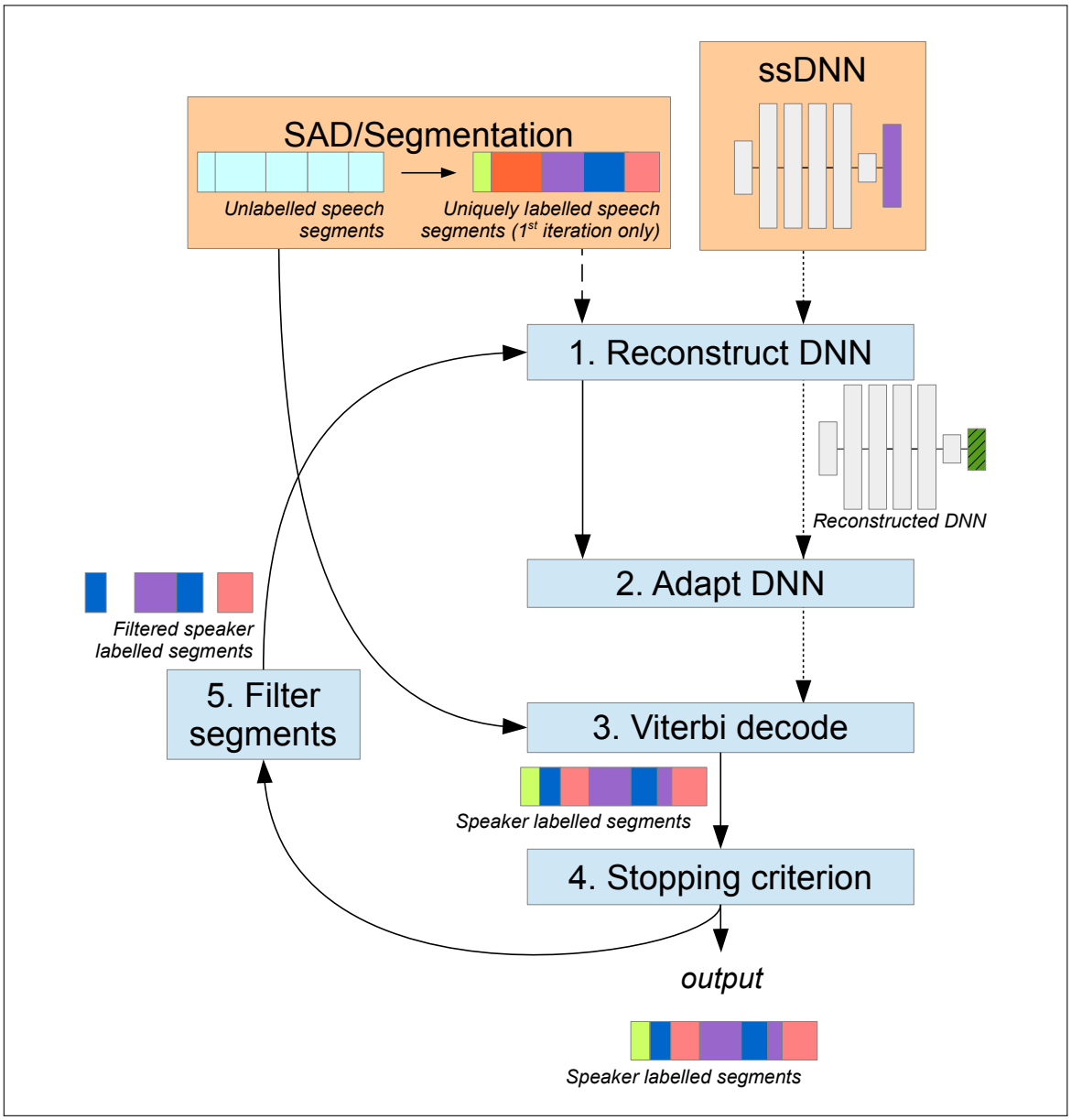

Fig. 6.1 The presented DNN-clustering algorithm is an iterative process requiring a SAD segmentation and a pretrained ssDNN. It consists of five stages: reconstructing new DNNs, adapting these DNNs, Viterbi decoding, checking a stopping criterion and, if the algorithm is allowed to continue, the segments are filtered.

Figure 6.1. Firstly, a DNN model is constructed based on the pretrained ssDNN. The final layer is removed and replaced with a randomly initialised final layer. The number of target classes represents the number of speakers in the input segments. For the first iteration, this is the uniquely labelled speech segments in which each segment is labelled as a separate

speaker. After reconstructing this DNN, a single iteration of adaptation, or training, is performed using either the uniquely labelled speech segments (first iteration only) or filtered speaker labelled segments. Once the model has been adapted to the given data, decoding is performed on the unlabelled speech segments. This results in speaker labelled segments. If resegmentation is permitted, the unlabelled speech segments may be labelled with more than one speaker, creating speaker boundaries and more segments. A stopping criterion is enforced which considers whether the method should cease or not. If the method should stop, the speaker labelled segments are the final output. If the method should continue, Step 5 is performed. This step filters the speaker labelled segments by removing the potentially unreliable segments. The process starts again from Step 1 given the filtered speaker labelled segments as the input, not the uniquely labelled speech segments.

6.2.1 Training an ssDNN

An ssDNN is a pretrained model which has learnt how to classify, or separate, speakers. As opposed to the DNNs for SAD presented in Section 5.3.1, the model is trained on speaker labelled data instead of speech and nonspeech data. This means the number of target classes in the final layer represents the number of speakers in the training data. There is also a fundamental change in the architecture. The most important layer is the bottleneck as the penultimate layer. This is a compression layer which reduces the dimensionality and aims to capture the information learnt by the model. The ssDNN is trained using the hybrid DNN-HMM approach described by Veselý et al. (2010). The discriminative DNN represents the probability density of the acoustic patterns associated to states with the HMM. The model is a feed-forward DNN which implements stochastic gradient descent algorithm with error backpropagation. Cross-entropy is applied as the objective. The weight update is performed per bunch, a block of N frames. Transforms alternate between being linear and nonlinear and hidden layers use the sigmoid nonlinearity whereas the output layer uses the softmax. The DNN maps a short sequence of frames, defined with a context window, into a probability distribution over HMM states. Viterbi decoding is performed to produce output segments.

Once trained, the model will be able to detect different speakers. The input layer is followed by several hidden layers before a bottleneck layer which compresses the structure. The final output layer has target classes representing the speakers in the training data. The previously described research which involves these models have used different numbers of hidden layers and different sizes of the hidden and bottleneck layers. Konig et al. (1998) used two hidden layers which contained 500 hidden units and encompassed the bottleneck layer of 34 neurons. Research based on this used two hidden layers, one with 512 hidden units and the other with 100 hidden units (Yella and Stolcke, 2015). The authors' bottleneck layer

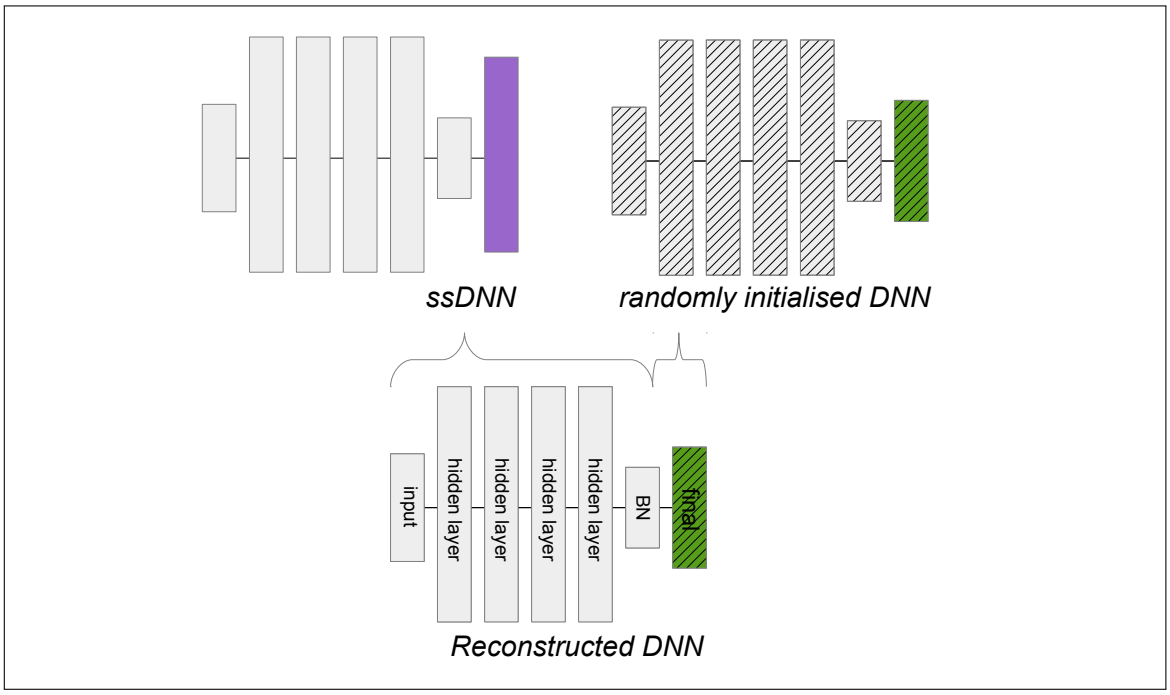

Fig. 6.2 In Step 1, DNNs are reconstructed, one for each recording, from the pretrained ssDNN. The reconstructed DNNs remove the final layer of the ssDNN and replace with a randomly initialised layer in which the target classes represent the clusters from the input (uniquely or filtered) speaker labelled segments. The bottleneck layer, BN, from the ssDNN is kept.

was 20 neurons to be a similar dimension to the MFCCs used as features. Furthermore, Liu et al. (2014) extracted bottleneck features from an ssDNN with three hidden layers where the first two contained 1745 hidden units and the third was the bottleneck with 13 neurons. The bottleneck was placed before the output layer as this position gave the best performance in the authors' initial experiments. This shows a large variation in the ssDNN topologies which have been investigated.

6.2.2 Reconstructing and Adapting a New DNN

Step 2 requires constructing a new DNN. This model is based on the pretrained ssDNN. The ssDNN model contains a bottleneck layer which captures the compressed information which has learnt how to separate speakers. The final layer of the ssDNN is removed and replaced by a final layer from a randomly initialised DNN. The (uniquely or filtered) speaker labelled segments, which are used for adaptation, provide the number of speakers representing the target classes in the new final layer. Building a DNN from the ssDNN is referred to as

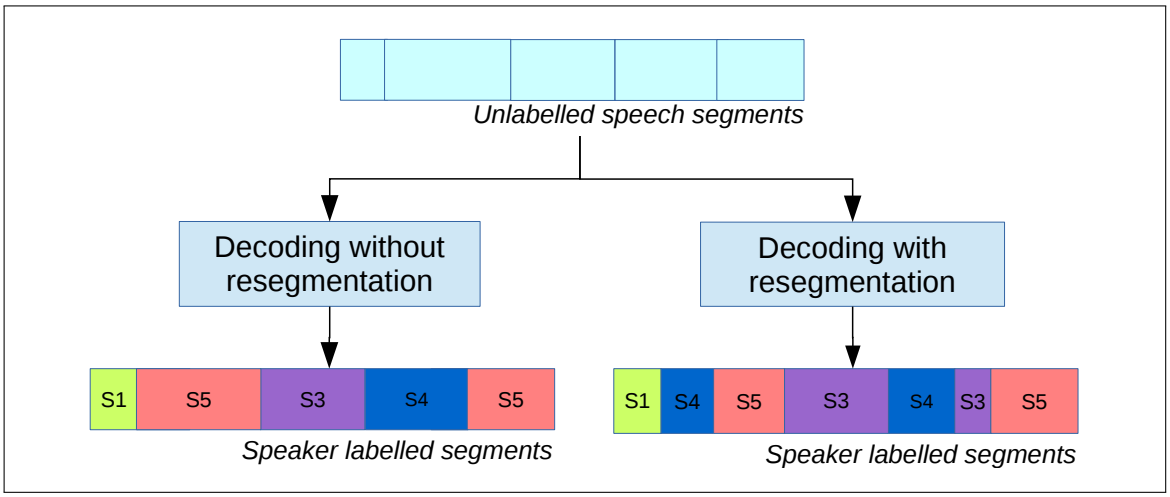

Fig. 6.3 In Step 3, the input to the decoding stage is the unlabelled speech segments. Viterbi decoding is performed either without or with resegmenting the data. The latter results in more segments where a speech segment has been labelled with more than one speaker.

reconstructing a DNN. Figure 6.2 shows this process and the reconstruction is performed in every iteration of the process.

The ssDNN model is ideally trained on a large amount of data consisting of a wide range of speaker labelled data. The hidden layers and specifically the bottleneck layer has captured the information required to distinguish between speakers. Step 2 adapts the newly constructed DNN to the provided segments. This is the uniquely labelled speech segments in the first iteration and the filtered segments in all further iterations. The bottleneck layer information from the ssDNN is used to learn the differences between the new speakers. A single iteration of adaptation, or fine tuning, is performed. The model is then used in the decoding stage.

6.2.3 Viterbi Decoding

Viterbi decoding is performed on the unlabelled speech segments every iteration. If a target class has detected no data in the decoding, the class is removed from the adaptation. This means classes are lost, but never gained. Viterbi decoding allows the speech segments to be labelled with one or many speakers, as seen in Figure 6.3. This splits, or resegments, the segments into further speaker boundaries. The resegmentation process allows for the input segments to be speaker-pure or speech only. However, this method uses segments which were originally defined as speech, this means any time defined as nonspeech cannot be recovered. It is possible to keep the same original segment boundaries and prevent resegmentation when

the log probabilities for each segment are considered. For each segment which contains more than one class, the class with the highest average log probability per frame is chosen for that segment.

As well as allowing resegmentation, other techniques are applied to improve the decoding output. To prevent many short segments from being detected, a minimum segment duration is enforced (Wooters et al., 2004). Viterbi decoding is performed using a HMM-GMM for the target classes. The states of the network are the speakers and a minimum state duration can be required (Young et al., 2006). Additionally, the grammar scale factor exists as a multiplier on the language model probabilities. The term grammar is referred to in the HTK book (Young et al., 2006) and represents the meta-structure or network of the HMMs. In this method for diarisation, there are HMMs for the different speakers as opposed to words or speech and nonspeech. The network contains the relationships of the HMMs and the grammar scale factor acts as a prior which raises the structure to a power. Changing this scale factor affects the output of the decoding.

6.2.4 Automatic Stopping Criterion

A fixed number of iterations can be applied, or a more sophisticated automatic stopping criterion is used. The DNN-clustering technique does not merge clusters pairwise like AHC. If data for a cluster does not exist in the speaker labelled segments used for fine tuning, then the model will not learn that speaker and the decoding stage cannot label a segment as that class. The class has been lost and its previously assigned segments are spread to one or more other clusters. For this iterative method, a simple technique for stopping is a fixed number of iterations. Instead, an automatic stopping criterion is applied based on confidence scores. A word-based confidence estimator is proposed by Zhang et al. (2014) for ASR and can be applied to diarisation. The confidence score represents the acoustic confidence of the DNNs on the hypothesis speaker label by accumulating the log posterior on each frame corresponding to that speaker:

$$C_{pos}(s_i, t_s, t_e) = \frac{1}{t_e - t_s - 1} \sum_{t_e}^{k=t_s} \log p_k \tag{6.1}$$

where s_i is the hypothesis speaker, the time spans from t_s to t_e, and p_k denotes the DNN posterior estimate value at frame k. For the stopping criterion, a confidence score is calculated for each speaker labelled segment. Considering the length of the segments, an overall average frame confidence score is calculated for each iteration and a threshold is placed on the change between iterations. If the change is less than that threshold, meaning the scores are stabilising

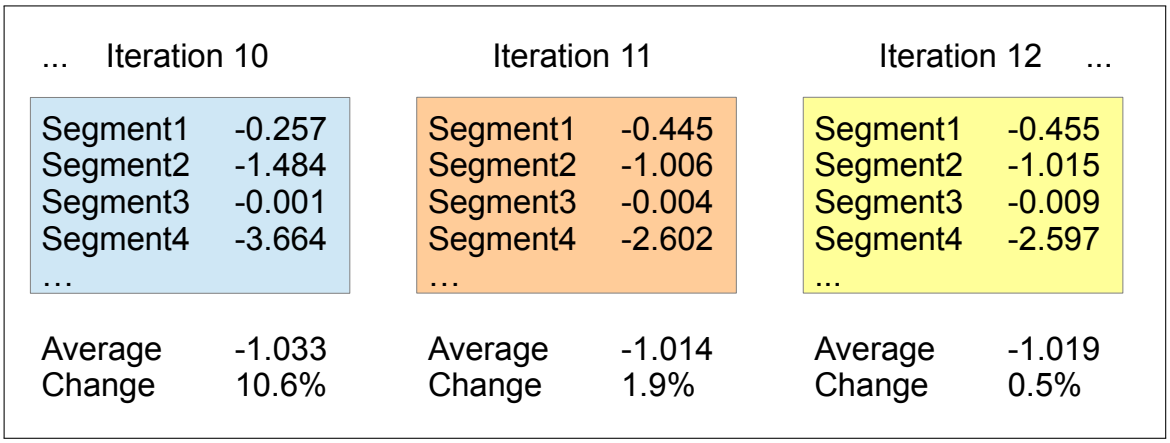

Fig. 6.4 In Step 4, the stopping criterion calculates the log posteriors for each segment as a confidence score. An average log posterior across the segments normalised by the number of frames is calculated for each iteration, and a threshold is applied to the change in this average score. Once the change goes below the threshold, the automatic stopping criterion ceases the iterations.

and converging, the process can stop and the current decoding output becomes the final output. This is shown in Figure 6.4.

6.2.5 Segmentation

Different input segments are required for different stages and Figure 6.5 displays the possibilities. For the input to Step 1 and 2 of Figure 6.1, the first iteration requires unlabelled speech segments output from a separate SAD or speaker segmentation system. For Step 1 and 2 in the first iteration, initial classes, or clusters, are required. The unlabelled speech segments are uniquely labelled where each segment represents a class. For Step 3, Viterbi decoding uses the unlabelled speech segments as input and the output is speaker labelled segments. Viterbi decoding is only applied within a segment. This means the initial speech boundaries are kept, but resegmentation permits additional speaker boundaries within these segments. If the stopping criterion is met, these speaker labelled segments are the final hypothesised output. If the stopping criterion is not met, the speaker labelled segments are filtered in Step 5. This removes segments which have been deemed unreliable. These filtered speaker labelled segments are the input to Step 1 and 2 again.

It was determined to be necessary in methods based on the ICSI-RT07 system that the speaker models should not be trained on all the data, but only the reliably-identified pure data (Sinclair and King, 2013). Han and Narayanan (2008) proposes a selective AHC in which segments longer than 3 seconds are initially clustered before the short segments,

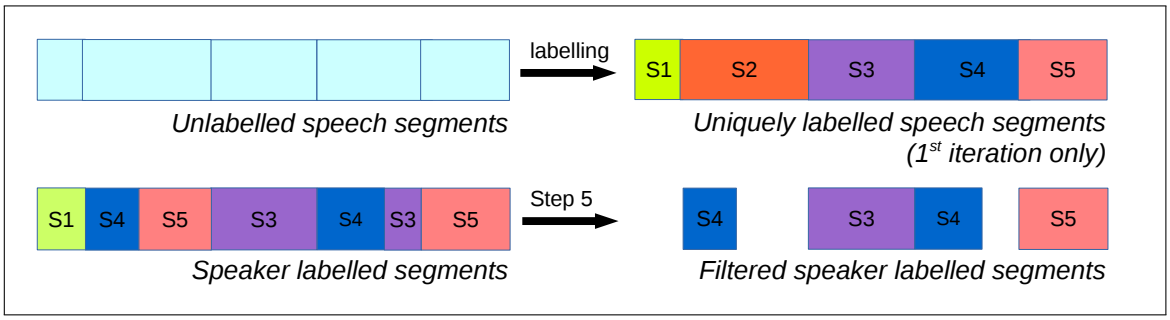

Fig. 6.5 At different stages, differently labelled segments are required. For the first iteration, every segment in the unlabelled speech segments is uniquely labelled as a separate speaker. The unlabelled speech segments are used by the Viterbi decoding stage which outputs speaker labelled segments. These are then refined to give filtered speaker labelled segments which are more reliable for the adaptation stage.

known to be more unreliable, are clustered into those previously detected clusters. The speech segments used for adaptation are vital to be as accurately labelled as possible, this means speaker or cluster labels. Errors would propagate through a system. Selecting accurate segments for adaptation would give the system a better starting point, and avoids adapting to inaccurate data.

As mentioned in Section 6.2.3, clustering and resegmentation are both performed in this method. Resegmentation creates speaker boundaries within the speech segments. However, if it is known that the speech segments are speaker-homogeneous, the segments which split can be deemed unreliable. Thus, those segments are filtered out of the input for the adaptation. Furthermore, short segments contain little information to base an entire speaker on each in the first iteration. Filtering on duration would lead to longer segments which contain more information assuming they are more useful when correctly labelled. Alternatively, segments are filtered by confidence using the same DNN-based confidence scores as calculated for the automatic stopping criterion. The segments with the least confidence are not used for the adaptation stage. Further to filtering by duration or by confidence, DNN performance can improve when presented with more data (Deng and Dong, 2015). Knowing initially that some segments could disrupt the adaptation, with each further iteration the amount of filtered segments could be reduced. This assumes each adaptation iteration is getting closer to stability and convergence.

6.3 Experiments

Experiments are performed on reference segments to determine the best setup for the method. This includes investigating the topology and training data of the ssDNN. Both RT07 and MGB data are considered for evaluating the method. Techniques to filter the segments in Step 5 and refine the adaptation in Step 2 are investigated. This is followed by experiments comparing a fixed number of iterations to the confidence score automatic stopping criterion. Finally, SAD segments are provided to the system for real evaluation of the proposed clustering method compared to using a public domain toolkit which uses Out-Of-Domain (OOD) supplementary data. The datasets are described in Section 6.3.1 followed by the experimental setup in Section 6.3.2.

6.3.1 Data

A robust clustering method is applicable to different domains. The method is designed for SDM data meaning more datasets are used for evaluation than in Chapter 5. The ssDNN must be trained before the method is performed. Several are pretrained on a combination of different meeting and broadcast media datasets. The meeting data considered is AMI IHM data (Section 3.2.1) and ICSI IHM data (Section 3.2.2). MGB data (Section 3.3.2) is used as broadcast media training data, however, this dataset is much smaller than the meeting datasets as little speaker labelled data exists.

Table 6.1 also details the reference segmentation used for tuning the setup of the method. Determining the optimum structure and setup of the clustering method is carried out using reference segments from RT07 and MGBMINI datasets, with overlap removed. This is referred to as MGBMINIREF and RT07REF, of which the SHEF reference is applied. This allows for comparisons and differences to be seen across the two domains and removing overlap prevents impurities negatively affecting the clustering. TBL data is not used in this stage as the MGB data contains many different programmes which is more useful as datasets of broadcast media contain a multitude of different programmes which vary in speakers, genre, format, etc. After the method setup has been defined, SAD segments are evaluated for RT07, TBL and the MGB test sets.

6.3.2 Setup

The topology of the ssDNN is investigated and described in the following experiments. The models were trained using filterbanks of 23 dimensions with a context window of 16 frames on both sides. Log Mel-filterbanks are used as opposed to MFCCs as they yield

Table 6.1 The reference segments are used for tuning the parameters in the method. In the meeting domain, the SHEF reference for RT07 is defined as RT07REF (SHEF) and referred to as simply RT07. In the meeting domain, a smaller dataset is comprised and referred to as MGBMINIREF.

Dataset	Domain	#Files	#Segments	#Speakers	Time (hrs)
RT07REF (SHEF)	meeting	8	10896	35	5.8
MGBMINIREF	media	7	4150	179	5.2

better performance with DNNs (Hermansky and Sharma, 1998). For the SAD segmentation experiments, the DNN-based SAD model *SAD6* trained on TBLTRAIN-SDM data (described in Section 5.3.1) is used as it achieved the best segmentation results for both TBL and RT07. For the MGBMINI, MGBDEV and MGBEVAL datasets from the broadcast media domain, a previously trained DNN-based SAD model used for the MGB challenge 2015 (Milner et al., 2015; Saz et al., 2015) is applied.

6.3.3 Results

Results are presented for the experiments performed to evaluate the presented technique. Firstly, the topology and training data of the ssDNN is investigated. Next, techniques to filter and select the potentially reliable segments for the adaptation are compared. Then, techniques to improve the decoding performance are investigated. The confidence score stopping criterion is investigated as a way to reduce computation when compared to using a fixed number of iterations. Lastly, SAD segments from previous DNN models are compared with speech segments from SHoUT. These speech segments are used as input to the DNN-based clustering technique and the clustering part of SHoUT for comparison.

6.3.3.1 Experiments: ssDNN implementation

Experiments begin by seeking the optimum implementation and setup of the ssDNN. Firstly, the different topologies are considered. This includes changing the hidden and bottleneck layer sizes as well as the number of hidden layers. Secondly, several ssDNNs are trained on different datasets and tested on both MGBMINIREF and RT07REF to see the domain differences, as well as allowing resegmentation to occur or not.

Investigating different ssDNN topologies

The investigation into the ssDNN topology was carried out specifically for the RT07 dataset. The ssDNN is trained on AMI-IHM data as initial experiments on an ssDNN on AMI-SDM

data performed slightly worse. The experiment varied the structure to determine which produced the highest performance. Three parts of the structure were varied:

- number of hidden layers: 2, 3 or 4
- size of hidden layers: 1000 or 1745 neurons
- size of bottleneck layer: 13, 26, 40, 100 or 150 neurons

Increasing the number of hidden layers creates a deeper networks and increasing the size of the hidden layer makes the network wider (Dong and Deng, 2015). Changing the DNN structure in this way is speculated to mean that the network is able to learn more complex data at different levels of abstraction, however, too deep could lead the network to overfit the training data. A wide network means an increase in parameters, however, a network with too many parameters will start to memorise the input, again overfit, and increase computation time. Lastly, the bottleneck layer captures the information learnt through the network of how to separate speakers. This compression layer reduces the dimension. The bottleneck layer is also the penultimate layer. Results are plotted in Figure 6.6 for each ssDNN in terms of the resulting DER, and the number of speakers. The decoded segments have been allowed to resegment, or split, allowing the number of segments to be considered too. As the reference segments are given, there is no MS or FA so the error falls under SE only.

The SE performance for the different ssDNN structures are depicted in plots (A) and (D). For the hidden layer, it is clear that a smaller bottleneck layer gives a lower SE. For most bottleneck sizes, 4 hidden layers gives a lower SE than either 2 or 3 hidden layers. Plots (B) and (E) display the number of segments detected at every iteration. Again most bottleneck layer sizes detect fewer segments with 4 hidden layers. However, the actual number of segments detected range from $\sim$32000 to $\sim$64000 after the first iteration, which is up to 6 times the reference number of segments, 10847. The experiment only investigates the effect different ssDNN structures have however this clearly shows the need for introducing techniques which will reduce the number of segments. Plots (C) and (F) show the number of speakers detected at every iteration. The number of speakers stops decreasing by 6 iterations, and it is clear that the smaller the bottleneck layer, the quicker the number of speakers converges. Again, the larger number of hidden layers, 4, generally detects a number of speakers closest to the reference. It is seen across all the plots that the ssDNNs with the bottleneck of size 13 generally give the lowest SEs, which corresponds to research by Liu et al. (2014) who use a bottleneck layer of size 13 for their ssDNN. Models with this bottleneck layer size also detect the number of segments and the number of speakers closest to the reference. This could be an effect of the data as meetings within RT07 only

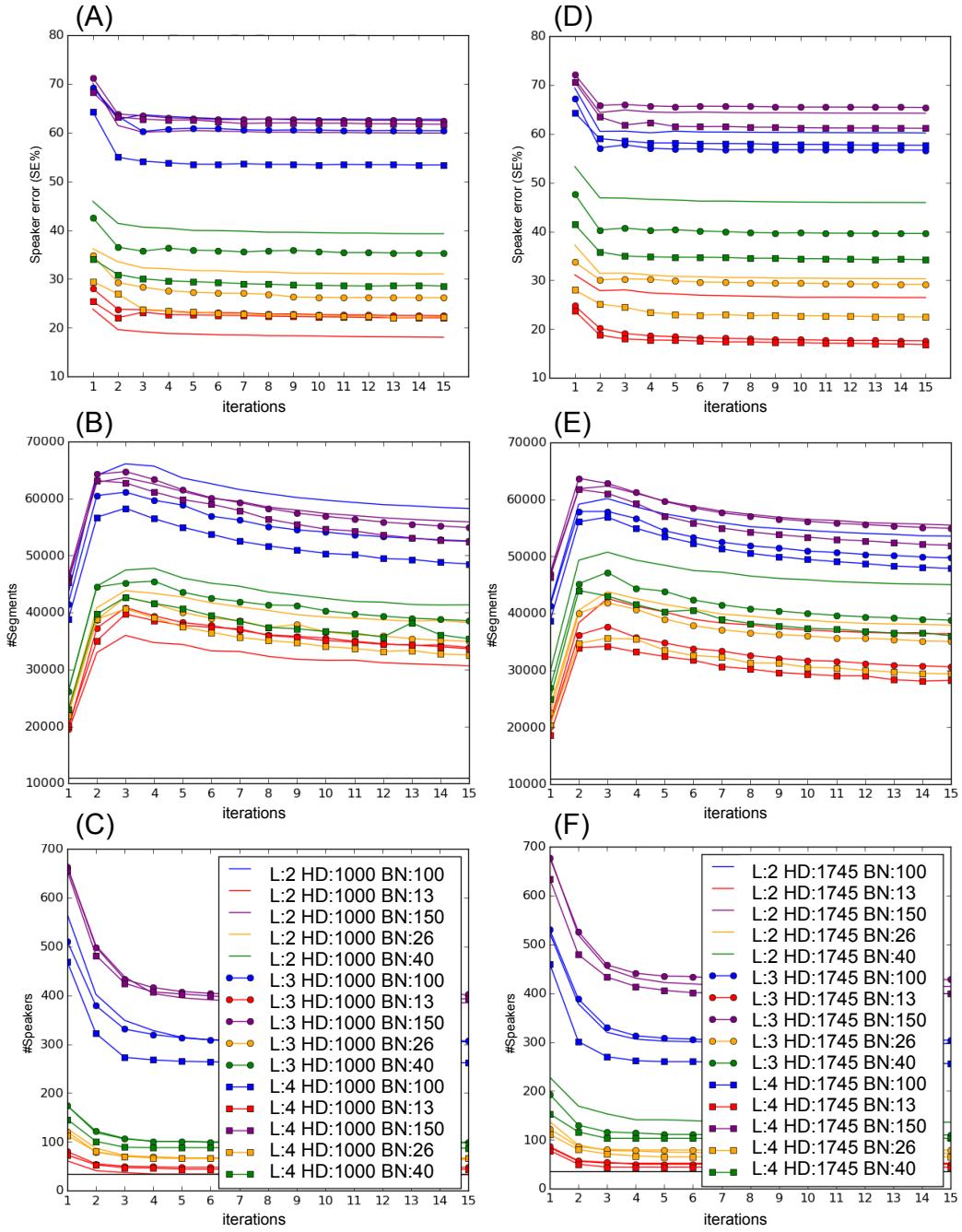

Fig. 6.6 Different ssDNN topologies are explored in terms of DER, segments and speakers detected. L refers to the number of hidden layers, HD refers to the hidden layer size and BN is the bottleneck layer size. Left hand plots, (A-C), refer to ssDNN structures having hidden layers with 1000 hidden units and right hand plots, (D-F), refer to hidden layers with 1745 hidden units. The colours refer to different bottleneck sizes. The different number of hidden layers is displayed as no icons for 1 layer, circles for 2 layers and squares for 3 layers.

Table 6.2 Results for ssDNNs trained on different combinations of meeting and broadcast media training data. The results are displayed as the number of speakers, #Spk, and Speaker Error (SE) from the DER for both MGBMINIREF and RT07REF.

Training Data	MGBMINIREF		RT07REF	
	Spk%	SE%	Spk%	SE%
AMI	17.9	66.4	80.0	44.6
MGB	11.2	69.0	54.3	57.2
ICSI	11.7	69.0	54.3	50.3
AMI+ICSI	19.0	65.2	68.6	49.9
AMI+MGBTRAIN	17.3	**62.3**	80.0	47.1
ICSI+MGBTRAIN	15.6	68.4	65.7	**42.8**
AMI+ICSI+MGBTRAIN	17.9	65.3	88.6	46.7

contain between 4 and 6 speakers. When hidden layers contain 1000 hidden units the best performance varies between 2 and 4 hidden layers. However, for hidden layers of size 1745 it is consistently the ssDNNs with 4 hidden layers which give the best performance across plots (D-F). The preferred topology for the ssDNNs is chosen to be 4 hidden layers of size 1745, with a bottleneck layer of size 13. This means a deeper and wider network with the smaller bottleneck gives better SE performance. The following experiments use this structure.

Training ssDNNs on data from different domains without resegmentation

Next, experiments consider ssDNNs trained on different datasets from different domains to see the effects on both MGBMINIREF and RT07REF test sets. These include IHM channels from two meeting datasets, AMI and ICSI, and SDM channels from broadcast media dataset MGBTRAIN, previously described in Section 6.3.1. IHM data is used for training when possible The first experiment does not permit resegmentation and a decision is made for the whole segment. This means a single speaker label is applied to each segment. Table 6.3 displays the results of the seven differently trained ssDNNs for both MGBMINIREF and RT07REF.

It is expected that good performance for RT07REF data is seen across the models trained on other meeting domain data and this is true. The ssDNN trained on AMI achieves 44.6% and the model trained on ICSI achieves 50.3%, as opposed to the broadcast media trained model which gives 57.2% speaker error. However, combining AMI and ICSI data gives better performance than ICSI only data by 0.4% and does not do as well as AMI data by itself. The best combination is ICSI+MGBTRAIN, which outperforms AMI by 1.8%. This is unexpected due to the mismatch in domain. A further combination of all three datasets reinforces that AMI data by itself gives better performance. For MGBMINIREF, it is clear

Table 6.3 Results for ssDNNs trained on different combinations of meeting and broadcast media training data, and in this case, resegmentation, or splits, is allowed. The results are displayed as the number of segments, #Segs, the number of speakers, #Spkrs, and Speaker Error (SE) from the DER for both MGBMINIREF and RT07REF.

Training Data	MGBMINIREF			RT07REF		
	Seg%	Spk%	SE%	Seg%	Spk%	SE%
AMI	576.5	24.6	61.1	251.2	108.6	**29.8**
ICSI	485.5	19.6	65.8	1415.4	74.3	31.5
MGB	517.3	17.3	64.7	2357.8	65.7	40.1
AMI+ICSI	596.5	30.2	59.2	228.6	102.9	32.8
AMI+MGBTRAIN	524.5	24.0	**56.4**	245.3	105.7	33.7
ICSI+MGBTRAIN	612.2	31.3	58.0	219.1	102.9	34.4
AMI+ICSI+MGBTRAIN	620.6	34.6	56.9	270.1	122.9	35.3

that AMI data gives the best performance. The three models with the highest error are not trained on AMI data. Training on AMI+MGBTRAIN has the best performance and this is due to the MGBTRAIN data being in the same domain. MGBTRAIN by itself does not perform well due to the lack of data. The MGBTRAIN speaker labelled data only consists of 6.5 hours and 650 speakers, which equates to 36 seconds on average for each speaker. This is very little data for training an ssDNN. The AMI data has 182 speakers in 50.9 hours giving 16.8 minutes on average for each speaker. Improvement is seen when training on AMI and MGBTRAIN as the latter is the same domain as the test set. The scores are generally high which implies many long segments are being incorrectly classified by the DNN clusterer. In terms of speakers, every setup over-clusters to below 35 speakers for RT07REF and below 179 for MGBMINIREF.

Training ssDNNs on data from different domains with resegmentation

Next, an experiment is performed where resegmentation is applied and the previously trained ssDNNs are evaluated. The results for both datasets are seen in Table 6.3. This time RT07REF behaves differently. The model trained on ICSI+MGBTRAIN data gave the best performance in the previous experiments but when resegmentation is performed, it gives one of the worst performances. The best model is trained on AMI only and achieves a speaker error of 29.8%, an improvement of 14.8%. It does not merge too many clusters as it detects 38 speakers, the reference is 36. However, allowing resegmentation has created more than twice as many segments. For MGBMINIREF, the previous ssDNN model trained on AMI+MGBTRAIN data again achieves the best performance with resegmentation. This achieves a 56.4% speaker error, an improvement of 5.9%. Again, more than 4 times as many expected segments have

Table 6.4 Results for experiments exploring ssDNNs trained on AMI and different amounts of the MGBTRAIN dataset for MGBMINIREF data.

Training Data	Seg%	Spk%	SE%
AMI	576.5	24.6	61.1
AMI+HALFMGBTRAIN	566.8	25.1	56.9
AMI+MGBTRAIN	524.5	24.0	56.4

been detected. However, more speakers are detected across the models but none of the models detect an amount close to the reference of 179. This may be because of the nature of broadcast media data in which many speakers will speak for only a short period of time, as seen in Figure 3.7 on page 65. The experiment shows that allowing resegmentation consistently improves the performance. This suggests that long segments had previously been incorrect and splitting, adding speaker boundaries, allows for parts of these segments to be labelled correctly. This is because the DER is a time-weighted metric (Section 4.1). However, every model over-segments the data as too many segments are detected.

Training ssDNNs with in-domain data

A further experiment is carried out for MGBMINIREF data in which an eighth ssDNN is trained. The model is trained on AMI data and half the amount of MGBTRAIN, randomly selected. This investigates how dependent the models are on in-domain data. Table 6.4 displays the results. The performance improves when models are trained with more amounts of in-domain data. Including half of the MGBTRAIN dataset, referred to as HALFMGBTRAIN, the SE decreases from 61.1% to 56.9% which is a gain of 3.2%. Using all the MGBTRAIN data decreases the SE a further 0.5%. As the MGBTRAIN data consists of only 6.5 hours, the larger AMI dataset, 50.9 hours, helps to improve performance. It is also noted that training on more in-domain data detects fewer segments. This again could be from training on more suitable data. The detected number of speakers does not give a clear indication of this.

6.3.3.2 Experiments: Filtering segments by time

The next experiments use the best performing ssDNNs from the previous experiment and resegmentation is allowed. MGBMINIREF experiments use the ssDNN trained on MGB and AMI data while the RT07REF experiments use the ssDNN trained on AMI data only. The input segments for the adaptation stage should ideally be as accurate as possible to prevent the reconstructed DNNs learning errors which will only result in further errors. The segments can be filtered by removing a percentage of time in two ways, considering the duration of the

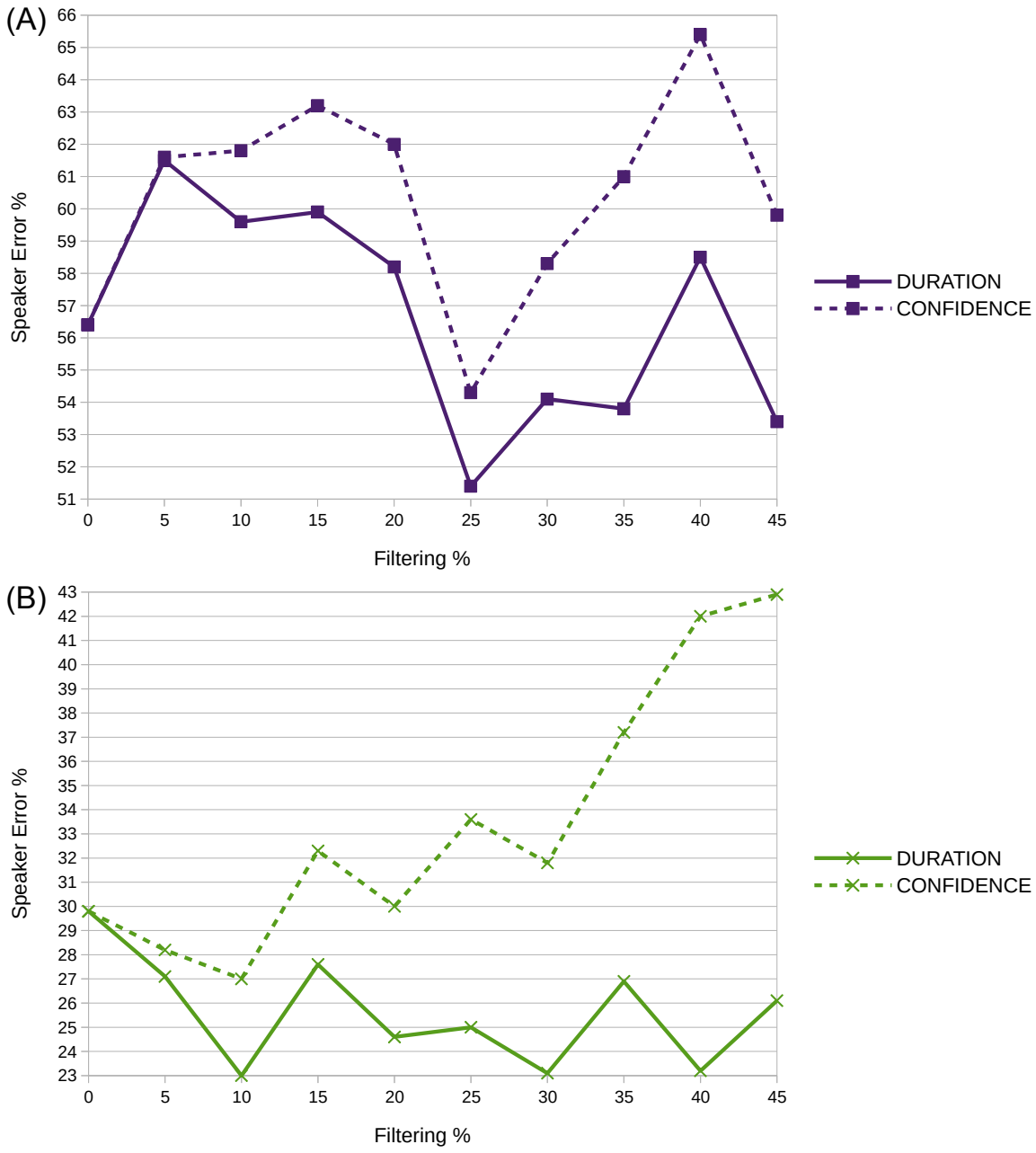

Fig. 6.7 Filtering the input segmentation by removing shortest segments first, 'duration', or low confidence segments first, 'confidence'. The percentage of time filtered is varied for (A) MGBMINIREF and (B) RT07REF.

segments or the confidence of the segments. This investigates improving the accuracy of the model.

Figure 6.7 (A) plots the SE applying either a duration or confidence score filtering for MGBMINIREF. It is clear that the duration filter consistently outperforms the confidence

score filter, by 0.1% at 10% filtering to 6.9% speaker error rate and 40% filtering. A trend of error reduction is seen until 25% filtering which gives the lowest error rate and the error increases after this with higher filtering. This shows that filtering out segments does improve performance by preventing the model being fine tuned to short, unreliable segments for both the short duration and the low-confident segments. However, as the performance gains decrease after a 25% filter, there is a balance between filtering the unreliable segments and filtering too much data. The duration filter of 25% achieves the best performance of 51.4% SE with 23173 segments and 74 speakers. This is the best performance in terms of SE, but it has detected 4 times as many expected segments.

The experiment on RT07REF data is plotted in Figure 6.7 (B). The same trend is seen where the duration filter consistently outperforms the confidence score filter. However, there is not a clear reduction in error as was seen with the MGBMINIREF data. Previously, a balance between the filtering of short, unreliable segments and the amount of data was seen. In terms of confidence filtering, 10% achieves the lowest speaker error of 27.0%. For duration filtering, both 30% and 40% achieves similar results of 23.1% and 23.2% respectively. This shows filtering out the unreliable short segments is more important and has a larger effect than the loss of data. The duration filter of 10% achieves the best performance of 23.0% speaker error with 21751 segments and 35 speakers, the expected number of speakers. This is the best performance in terms of SE and speakers detected, but it has also over-segmented as for MGBMINIREF data.

6.3.3.3 Experiments: Reducing over-segmentation

The experiment investigates whether applying a minimum state duration helps to reduce over-segmentation. The decoding stage resegments the data when it detects a different speaker inside a segment. This creates many speaker boundaries. Previously, when no minimum duration was set, the optimum ssDNN for MGBMINIREF trained on AMI+MGB with 25% of the time filtered by segment duration detected 23173 segments, which is 4 times the number of reference segments. For the ssDNN trained on AMI data with 10% of the time filtered by segment duration, the model detects nearly twice as many reference segments for RT07REF. As the input segments are known to be speaker-pure, it is assumed that the amount of hypothesis segments detected would be close to the expected number. The minimum state duration is varied from 10 states, 0.1 seconds, to 40 states which is 0.4 seconds. It is noted that any segment which is shorter than the fixed duration is decoded without a duration constraint.

Performance for MGBMINIREF is displayed in Figure 6.8 (A). The amount of detected segments is plotted along with the SE rate. The states were varied along with the different

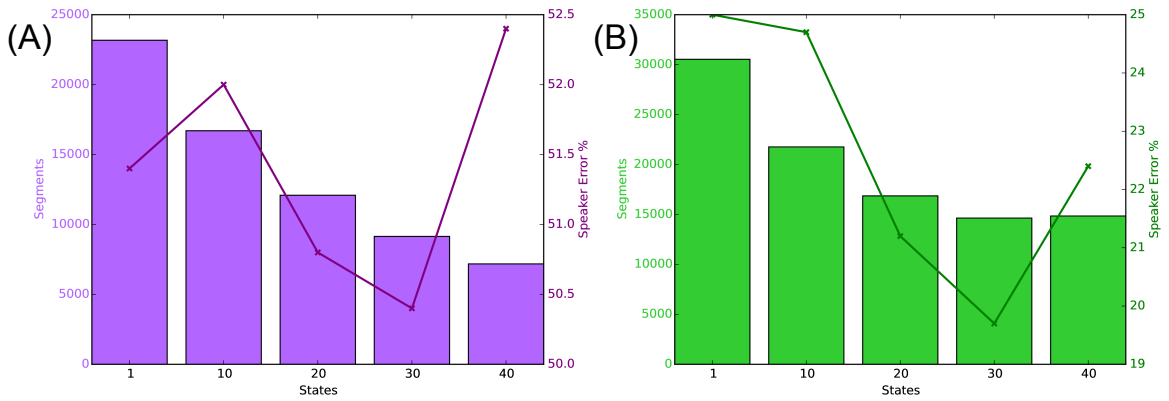

Fig. 6.8 The state duration is tuned for (A) MGBMINIREF and (B) RT07REF. The bars represent the number of detected segments and the points are the Speaker Errors (SE) given those segmentations.

Table 6.5 Results for state duration experiments ranging from 1 to 40 states, with RT07REF data. The results with the previous optimum duration filter, 10%, is displayed alongside the new optimum filter, 25%.

States	Duration Filter = 10%			Duration Filter = 25%		
	Seg%	Spk%	SE%	Seg%	Spk%	SE%
1	199.6	100.0	23.0	280.2	148.6	25.0
10	162.5	100.0	23.5	199.7	145.7	24.7
20	140.7	100.0	22.9	154.8	140.0	21.2
30	128.5	102.9	21.9	134.4	140.0	**19.7**
40	129.0	97.1	21.6	136.3	134.3	22.4

duration filterings. The fraction of time filtered which gives the previously best performance also gave the best performance with a fixed duration of 30 states. Enforcing a minimum state duration greatly reduces the amount of segments being detected. A larger duration filter reduces the over-segmenation. In terms of speaker error, initially at 10 states the error increases. It then consistently decreases at 20 and lower again at 30 states. An SE of 50.4% is achieved, with 9144 segments detected. The number of segments at 40 states is lower again, however the SE reaches its worst performance at 52.4%.

For RT07REF, the duration filter which gave the best performance when changing the number of states did not give the best performance in the previous experiment. Table 6.5 displays the results for both of these filters. For the 10% filter, the lowest speaker error is achieved at 40 states with 14058 detected segments. However, the 25% filter achieves an absolute error rate 1.9% lower with 14642 segments at 134.4 Spk% . The focus of this experiment is to reduce the over-segmentation, which both filters achieve. The 25% filter yields 5.4% more segments but with a significantly lower error rate, so this filter is

Table 6.6 The MGBMINIREF and RT07REF results for the refinement techniques are presented. The clustering can be refined in two ways: reducing the percentage of filtering (RDF) and removing the split segments from training (RMS).

Refinement	MGBMINIREF			RT07REF		
	Seg%	Spk%	SE%	Seg%	Spk%	SE%
-	220.3	38.5	**50.4**	134.4	140.0	19.7
RMS	284.2	54.7	56.3	131.7	85.7	16.8
RDF	220.7	39.1	51.3	136.8	134.3	21.9
RDF+RMS	292.8	55.3	53.1	123.5	82.9	**15.4**

determined as providing better performance with the increased number of states. Performance with the duration filter of 25% is plotted in Figure 6.8 (B) which shows the similar trends as in MGBMINIREF. Enforcing a minimum state duration again helps to reduce the over-segmentation. Like MGBMINIREF, the larger the state duration the fewer speaker boundaries are detected. However, this does not consistently decrease as seen before, as a 40 state duration achieves 202 more segments than with a 30 state duration enforced. The lowest SE of 19.7% is also seen at 30 states. These results show that for data from two different domains, applying a minimum state duration reduces the number of segments detected and improves the clustering SE rate.

6.3.3.4 Experiments: Further filtering techniques

Experiments are performed to refine the clustering method in terms of the input segmentation used for adaptation in Step 2. Two techniques are applied to improve the purity and reliability of the input segments in the adaptation stage. The first affects the input segments by removing those which have split in the resegmentation process. If the segments are known to be speaker-pure, then any that are split across two speakers must be incorrect. Secondly, the amount of time being filtered at every iteration is reduced by 5%. For example, with 25% filtering this becomes 23.75%, then 22.56%, and so on. Both techniques are applied separately and then combined.

The results for these input segmentation refinement techniques is seen in Table 6.6 for MGBMINIREF. Preventing the DNN from adapting to the split segments, RMS, increases the error rate by 5.9%, yields 16.2% more speakers and 63.9% extra segments. Looking deeper, without removing the split segments the second iteration uses 398.8% segments across the recordings when adapting. However, when removing split segments this is greatly reduced to 10.5%, across 7 recordings. More than one speaker is detected for many segments. This causes too little data to be available for adapting and reduces performance. Reducing the filtering, RDF, allows for more data to be used in the adaptation stage at each iteration. For

the second iteration, this technique adapts using 427.0% of expected segments as opposed to 338.8% across the recordings. However, the SE does not drop below the best performance without reducing the filtering. This shows that despite training on more data, these segments do not have the same quality as the unfiltered segments as an improvement in SE is not seen. Combining removing split segments and reducing the filtering improves over just removing the splits. This is because the reduced filtering technique allows more segments to be included, 12.1% at the second iteration. However this is still a small amount of data and the combination does not improve over using neither of these techniques.

Table 6.6 also shows the results for RT07REF when applying the two techniques for refining the input segmentation. A different result is seen in this data. Removing the split segments, RMS, helps to reduce the error rate from 19.7% to 16.8%. A similar number of segments is detected, however, 19 speakers, or 54.3%, are lost. Looking into the number of segments across recordings at the second iteration, without removing split segments the amount is 202.0% of expected segments and this is reduced to 39.1% when split segments are removed. This is not as low as in MGBMINIREF and contains enough reliable segments to give an improvement of 2.9% SE. The reduction of the filtering technique, RDF, performs worse than with a fixed amount of filtering. This means more and more unreliable segments are applied in the adaptation stage. Lastly, the combination of these two methods helps performance and gives the lowest error rate of 15.4%. Removing the split segments and reducing the filtering results in 44.1% of segments at the second iteration. This has removed the unreliable split segments and allowed extra more accurate segments to be adapted with. However, the number of speakers is fewer than the reference.

6.3.3.5 Experiments: Grammar scale factor

The experiment investigates changes to the implementation of the methods by varying the grammar scale factor. This again affects the decoding stage. The grammar scale factor adds a prior to the meta structure, or grammar, which represents a network of HMMs are their relationship. In this case, there are HMMs for each speaker. Table 6.7 shows the results for both datasets. The grammar scale factor was varied from 1 to 50 and the scale with the best speaker error is displayed. For MGBMINIREF, a grammar scale factor of 26 gives the lowest speaker error of 47.9%. This is drop of 2.5% absolute and brings the error to below 50% for the first time. It also benefits the segmentation as a larger reduction in segments is detected, closer to the expected number of 4150. In terms of speakers, fewer are detected. There are 179 expected speakers but many speak for a short amount of time making it difficult to detect them all. For RT07REF, the best performance comes from a grammar scale factor of 6. This reduces the SE by 2.3% absolute, nearly the same as for MGBMINREF. Fewer segments

Table 6.7 Results for both MGBMINREF and RT07REF datasets when the grammar scale factor is varied, GSF. The optimum value for each dataset in terms of SE is displayed and the default GSF is 1.

MGBMINIREF				RT07REF			
GSF	Seg%	Spk%	SE%	GSF	Seg%	Spk%	SE%
1	220.3	38.5	50.4	1	123.5	82.9	15.4
26	136.8	25.1	47.9	6	113.8	100.0	13.1

are detected, but not on the same scale as before. However, it again is closer to the expected number of 10896 segments. For this data there is an increase of speakers detected and in fact it yields the correct number of speakers, 35. Both datasets benefit from applying a scale factor to the cluster probabilities, however the scale is different due to the different data types.

6.3.3.6 Experiments: Stopping criterion

All previous experiments were performed to 20 iterations. An automatic stopping criterion is proposed in Section 6.2.4. Each recording stops at a different iteration and it is evaluated against iterating until a specified iteration is reached.

Fixed iterations

The SE for every iteration is plotted for each recording for the best configuration so far. For MGBMINIREF, this is displayed in Figure 6.9 (A). A large variation in performance is seen in the recordings. However, across the iterations minimal improvements are seen after the first couple of iterations. For example, MGBMINREF2 and MGBMINIREF6 increase in error after the first iteration and do not reach the lowest error seen in iteration 1. The rest do show improvement and the largest gains are seen in the first couple of iterations. The number of segments steadily reduces from 137.5% in iteration 2 to 136.8% at iteration 20. The amount of speakers detected decrease initially from 57.0% to 35.2% from iteration 4 onwards. This lack of improvement could be due to the ssDNN not being trained on enough in-domain data and the inherent difficulties and negative effects in the data itself. As each recording is from a different TV programme, the variation in style and content is wide.

Figure 6.9 (B) shows the performance across the recordings in RT07REF for each iteration up to 20. The performance across the recordings is more consistent than plot (A). This is because each recording has the same style of being a meeting. Variation is seen in terms of room setup, where every pair of meetings was recorded in the same room. The largest improvement is seen in iteration 2, in which the average DER result drops from

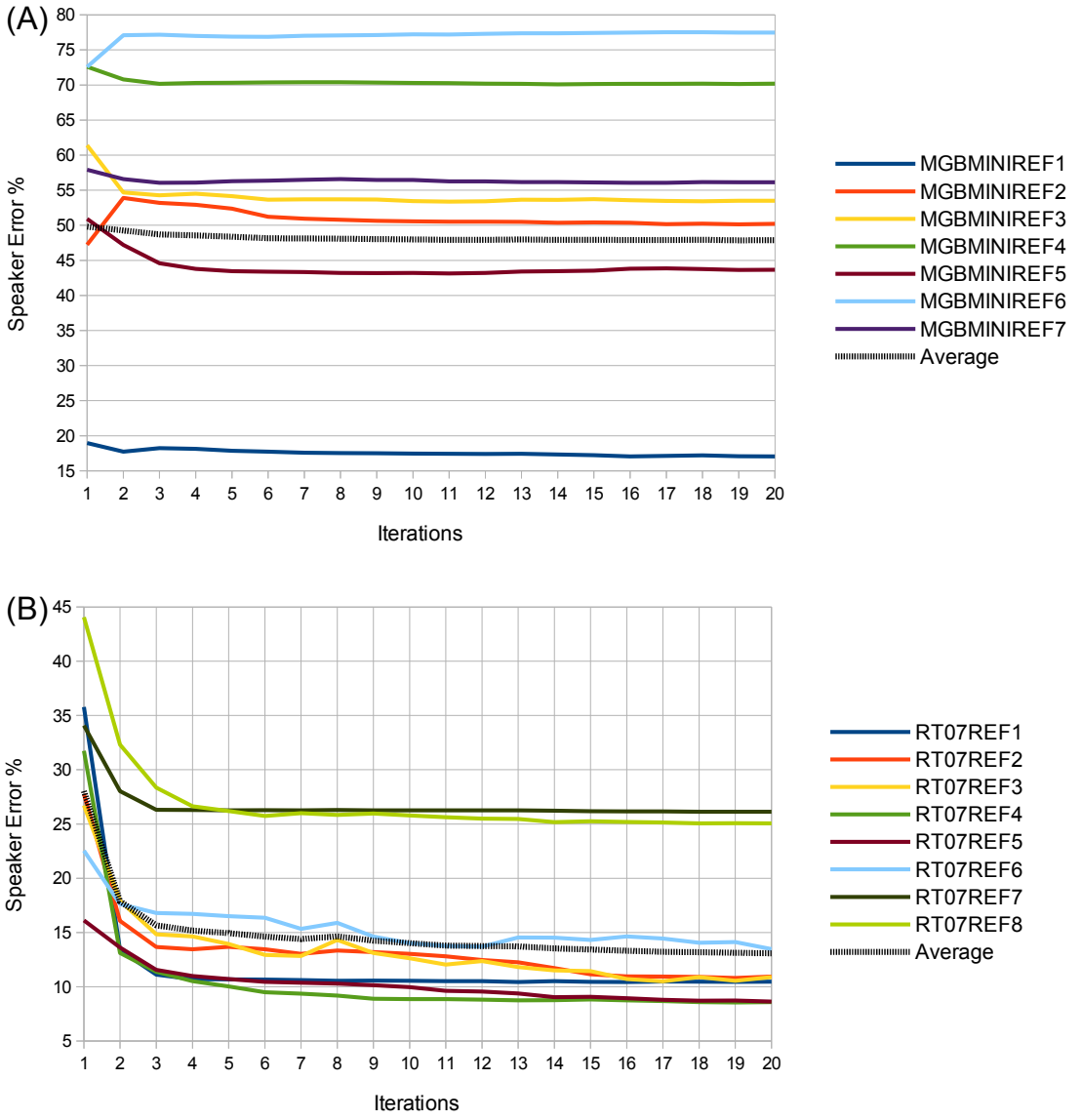

Fig. 6.9 The SE results for every iteration, up to 20, is plotted for all 7 recordings in (A) MGBMINIREF and (B) RT07REF. The black dotted line represents the average SE for the dataset.

28.0% with 80 speakers to 17.9% with 46 speakers. In terms of Spk%, this is a drop from 228.6% to 131.4%. At iteration 5, 10.0% of expected speakers are detected and no more clustering occurs in further iterations. After iteration 5, the speaker error decreases until an absolute error reduction of 1.1% is seen at iteration 20. The recordings RT07REF3 and RT07REF6 both show an increase of speaker error at iteration 8, showing the improvement in speaker error is not completely consistent across the iterations. This shows that the largest

Table 6.8 Results for the MGBMINIREF data comparing 5 fixed iterations to the stopping criterion with a 1% threshold. The number of iterations performed with the automatic stopping criterion is referred to as #Iters.

Setup	Fixed iterations (5)			Automatic stopping criterion			
	Seg%	Spk%	SE%	Seg%	Spk%	SE%	#Iters
Resegmentation	528.4	24.0	56.4	528.1	24.0	56.4	36
+DurationFilter=25%	671.8	43.0	52.9	658.6	43.0	52.8	41
+States=30	254.9	39.1	51.4	240.9	39.1	51.0	52
+GSF=26	141.7	35.2	48.4	139.1	35.2	48.1	47

Table 6.9 Results for the RT07REF data comparing 5 fixed iterations to the automatic stopping criterion with a 1% threshold.

Setup	Fixed iterations (5)			Automatic stopping criterion			
	Seg%	Spk%	SE%	Seg%	Spk%	SE%	#Iters
Resegmentation	252.8	108.6	29.8	252.5	108.6	29.8	40
+DurationFilter=25%	323.3	154.3	27.8	319.1	154.3	27.7	46
+States=30	143.9	140.0	21.8	140.4	140.0	21.1	56
+RDF+RMS	140.1	91.4	19.3	137.1	91.4	18.3	53
+GSF=6	120.1	100.0	14.9	119.5	100.0	14.8	50

improvements are seen in iteration 1 to 5, and the first iteration successfully detects the correct number of speakers.

Confidence score automatic stopping criterion

The stopping criterion is investigated. For each of the previous experiments, the results at iteration 5 are compared to using the automatic stopping criterion with a threshold of 1%.

Results for MGBMINIREF data can be seen in Table 6.8. As there are 7 recordings, the total number of iterations when stopping at the fifth iteration is 35. The progression through the previous experiments is seen. The number of speakers is consistent in both stopping methods whereas the number of segments varies slightly, fewer are detected with the automatic stopping criterion. In terms of speaker error, the results are the same or better with the automatic stopping criterion. A maximum improvement of 0.4% is seen. However, these results are achieved with more iterations, up to 49% more. The automatic stopping criterion achieves better performance with a cost of more iterations.

Table 6.9 displays results for RT07REF with a fixed number of iterations compared to using the stopping criterion with a 1% threshold. Similar results are seen as for MGBMINIREF. The speaker error is either the same or better without fixed iterations. The third experiment loses a speaker with the automatic stopping criterion but at the largest improvement of 1% SE

Table 6.10 Results are displayed for the two stopping criterion when the best possible setup is applied. The fixed iterations method is performed for 50 iterations per recording and from these the iteration with the lowest SE is manually chosen; this is what an ideal automatic stopping criterion aims for.

Stopping Criterion	MGBMINIREF				RT07REF			
	Seg%	Spk%	SE%	#Iters	Seg%	Spk%	SE%	#Iters
Fixed iterations: 5	141.7	35.2	48.4	35	120.1	100.0	14.9	40
Fixed iterations: 50	135.6	35.2	47.7	350	113.3	100.0	13.0	400
Automatic (1%)	139.1	35.2	48.1	47	119.5	100.0	14.8	50
Manual	124.5	39.1	46.7	178	113.4	100.0	12.9	213

Again, fewer segments are detected without fixed iterations, which is closer to the expected amount. This occurs in either the same number of iterations or more, up to 40% more. This does not help to reduce the computation or time however. Nevertheless, allowing each recording to stop at a different iteration does achieve a better performance.

Given perfect conditions

The final experiments seek the best possible performance for both stopping methods. For a fixed number of iterations, the best method implementation is 50 iterations per recording. For the automatic stopping criterion, the iteration (from 1 to 50) providing the lowest speaker error for each recording is manually selected. This result is the ideal result for the criterion. Table 6.10 displays the results for both datasets. The MGBMINIREF performance shows that ten times as many iterations has an improvement of less than 1% SE, a small reduction in segments and the same number of speakers. In terms of the automatic criterion, manually selecting the iterations with the lowest error achieves a better result than 50 fixed iterations per recording, by 1% SE. For RT07REF, a larger improvement is seen from 5 iterations to 50 iterations per file in which there is a reduction of 1.9% SE. However, a smaller gain is seen when comparing the manually selected iterations to the 50 iteration per file, where there is a reduction of 0.1% SE. This shows that as the automatic stopping criterion has the lowest possible result in both cases, using the automatic stopping criterion can outperform the fixed number of iterations.

6.3.3.7 Experiments: SAD segments

So far, the best implementation of the method for two different domains has been decided along with an acceptable automatic stopping criterion. These previous experiments have used the speaker-pure reference segmentation with overlapping time removed. Segments

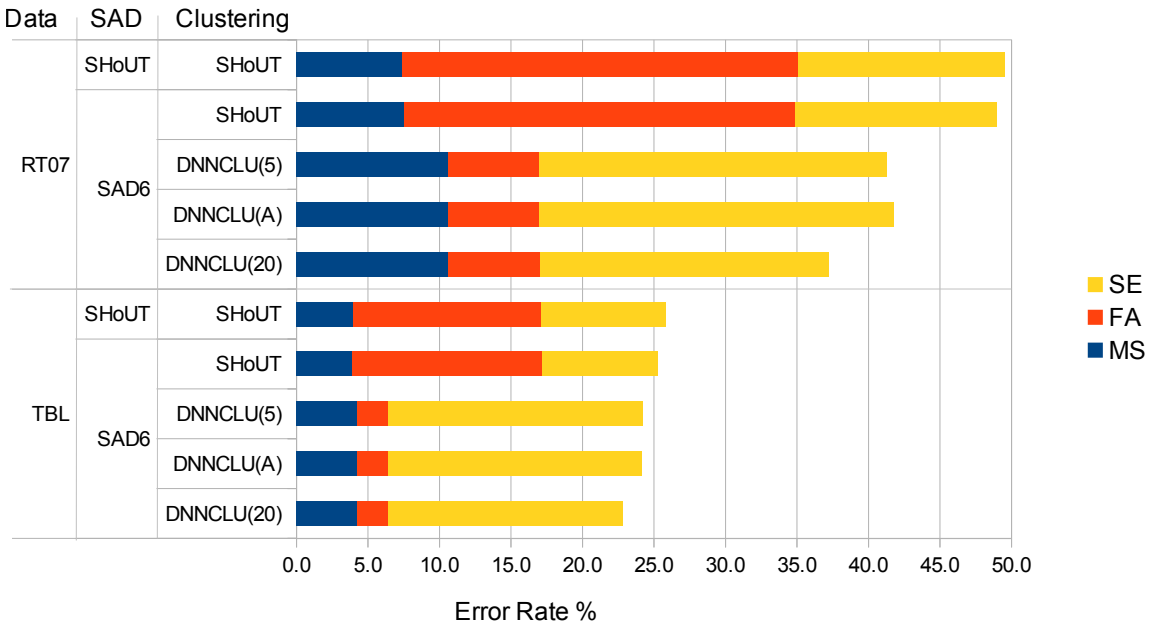

Fig. 6.10 Results when SAD segments are provided to the DNNCLU method, as opposed to using the reference segments. The tuned parameters for RT07 work best for TBL data. The number of iterations per recording is shown in brackets where (A) refers to the automatic stopping criterion.

from a SAD or speaker segmentation method are used to evaluate the clustering technique. Test sets are used from the meeting domain, RT07, and the broadcast media domain: TBL, MGBMINI, MGBDEV and MGBEVAL. The results are evaluated against the public domain toolkit SHoUT, as described in Section 3.4.3. It consists of two steps, SAD followed by clustering which allows resegmentation to occur, as in the proposed DNN clustering method. The SAD stage is semi-supervised as it uses speech and nonspeech models pretrained on BN data. The clustering and resegmentation is unsupervised. DNN-based models for SAD described in Section 6.3.2 are applied for all the data. To evaluate the clustering method, either SHoUT or the proposed DNN-clustering technique is applied. For the latter, both a fixed number of iterations (5 or 20 per recording) is considered as well as a stopping criterion with a threshold of 1%. The proposed method is denoted DNNCLU and the number of iterations used is noted in brackets.

Figure 6.10 displays the DER results for both RT07 and TBL data. For RT07 data, using the toolkit for SAD and clustering achieves a DER of 49.5%. Using SAD6 model instead, achieves 49.0% DER. This shows that the *SAD6* model yields a better segmentation. However, as SHoUT clusters and resegments including for nonspeech, the MS and FA rates are not the same. When using *SAD6* with DNNCLU, the MS increases however the FA rate

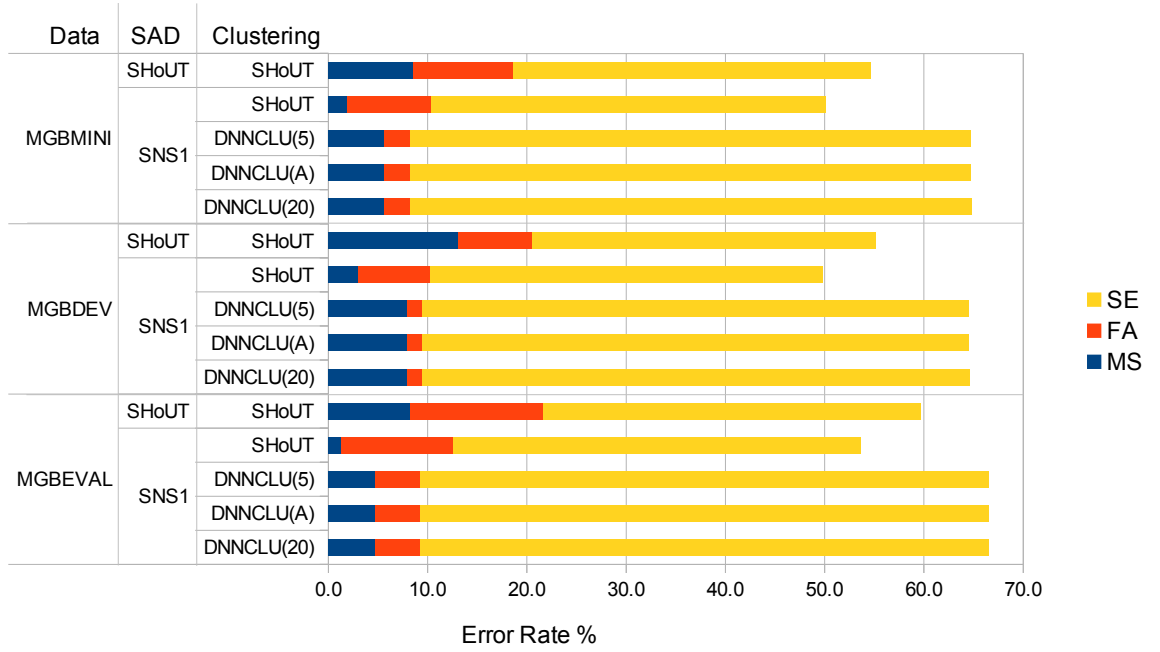

Fig. 6.11 Results when SAD segments are provided to the DNNCLU method, as opposed to using the reference segments. The parameters tuned for MGBMINIREF are applied for these segments. The number of iterations per recording is shown in brackets where (A) refers to the automatic stopping criterion.

greatly reduces yielding a better segmentation overall. The DNNCLU method improves over the toolkit results. However, the automatic stopping criterion performs worse than at 5 fixed iterations. Performing 20 iterations gives the lowest DER of 37.2% with a MS of 10.6% and 6.4% FA. For TBL data, the method implementation for the meeting domain gave better performance than for the broadcast media domain. This is due to the TBL programme being a discussion show which takes a meeting format, which has been recorded in a studio. Similar trends are seen as for RT07 data. *SAD6* outperforms the SHoUT SAD and DNNCLU outperforms the toolkits clustering stage. The clustering improvements are smaller than previously seen, however, the automatic stopping criterion does improve over 5 fixed iterations by a small 0.1%. Using 20 iterations per recording, a total of 200 iterations, achieves the lowest DER of 22.8% with a MS of 4.3% and 2.1% FA.

Figure 6.11 displays the DER results for MGBMINI, MGBDEV and MGBEVAL datasets. Across these MGB datasets, the trends are the same. The results using the *SNS1* SAD model and the SHoUT clustering and resegmentation show a larger improvement over using SHoUT in both stages. This in fact is the lowest result, as the DNNCLU method fails to beat the toolkit performance. This is largely due to the pretrained ssDNN. It was shown in Section 6.3.3.1 that the more in domain data available for training, the better the performance. Only 6.5

hours of speaker labelling MGB data was available. This is compared to around 50 hours for AMI meeting data, making the ssDNN model trained on AMI much more reliable. In this case, the automatic stopping criterion either performs worse or better by 0.1%. This is a minimal change with the DER being so high.

6.3.3.8 Experiments: SEGF and other metrics

The results from the method using a DNN-based SAD model and DNN-based clustering with the automatic stopping criterion are rescored. The SEGF is applied as well as other metrics: DPC, BNDF, and the purity measure K. Table 6.11 displays the results which are comparable to the baseline results of SHoUT in Table 4.1 which is included here for ease of viewing. In terms of DER performance, the DNNCLU method with the automatic stopping criterion outperforms SHoUT across both RT07 and TBL data. However, the method is not successful for the three MGB datasets. Each produces a performance around 65% DER which is nearly double the DER of SHoUT for MGBMINI. The DPC scores beat the relevant baselines for all datasets except MGBEVAL. The DPC score for this dataset is larger, possibly due to fewer segments being detected. The BNDF measure scores are better than SHoUT for every test set. For RT07 and TBL, the results are doubled which shows allowing resegmentation, as well as having good initial SAD segmentation, detects the optimum speaker changes. However, when looking at the purity measure K, performance for DNNCLU is worse than SHoUT across all the datasets. This shows the clustering method needs to improve on making a decision on the speaker labels. For the SEGF metric, improved performance over the baseline is seen for all datasets. However, the baseline numbers are poor to begin with and only minimal improvement is seen for the MGB datasets. Overall, these results for RT07 and TBL are not as good as the results for the Methods proposed in Chapter 5. As the purity scores are not as good as the baseline but the BNDF and SEGF are better, there is a larger need for improving the speaker labelling rather than the boundary and segment decisions.

6.3.4 Discussion

Section 6.3.3 has displayed the results of the steps taken to produce a well performing clustering method which takes a DNN approach. It was evaluated on two different domains: meeting and broadcast media data. For the meeting domain the RT07 test set was used for evaluation and from the broadcast media domain, datasets from the MGB challenge were used.

Experiments investigated the best performing setup and implementation of the proposed clustering and resegmentation method by using the reference segmentation, as this is assumed

Table 6.11 The final results using SAD segmentation and DNNCLU are presented and compared with results using SHoUT for both SAD and clustering. The DERs are evaluated alongside boundary measures (DPC and BNDF), an overall purity measure (K) and the Segment F-measure (SEGF) presented in Chapter 4. The metrics DER, BNDF, K and SEGF are percentages and the DPC is measured in milliseconds.

Data	System	Seg%	Spk%	DER	DPC	BNDF	K	SEGF
RT07	SHoUT	75.6	117.1	49.5	0.8	30.0	71.3	0.5
	DNNCLU(A)	114.6	114.3	41.9	0.7	61.3	64.3	23.8
TBL	SHoUT	85.7	172.5	25.8	0.7	26.3	83.2	0.7
	DNNCLU(A)	138.0	92.5	24.1	0.7	66.7	75.4	35.9
MGBMINI	SHoUT	129.0	21.8	36.0	3.6	47.5	54.3	0.7
	DNNCLU(A)	130.2	21.2	64.7	3.0	54.3	47.0	3.3
MGBDEV	SHoUT	141.0	70.1	53.9	2.7	48.9	54.7	1.5
	DNNCLU(A)	127.6	20.7	64.4	2.5	55.8	49.3	4.6
MGBEVAL	SHoUT	166.7	77.0	59.5	5.8	44.7	54.2	0.8
	DNNCLU(A)	161.0	22.3	66.5	7.7	54.4	42.9	4.5

to be speaker-pure. The key component to this DNNCLU method is the pretrained ssDNN, which makes this method semi-supervised. Different topologies were considered in terms of number of hidden layers, and size of the hidden layers and bottleneck layer. It was determined that performance improved with wider and more hidden layers, and a smaller bottleneck layer. Secondly, combinations of different datasets, from both the meeting and broadcast media domain, were compared. The ssDNN trained on AMI-IHM data performed best for RT07REF data when resegmentation was allowed, and the ssDNN trained on AMI-IHM and MGB-SDM performed best for MGBMINIREF with resegmentation. It was then shown that more in domain data improved results, however there is not enough available for MGB.

The next experiment considered more accurate and reliable segments for adaptation. It has been shown that speaker models trained on reliably-identified pure data lead to higher performance even if train on limited data (Sinclair and King, 2013). Segments were filtered by removing a percentage of the total time in either short segments or segments with low confidence. It was determined that there is a balance between removing unreliable segments and the amount of data being removed. Removing a fraction of the shortest segments benefited both datasets the most. However, it was noticed that the method greatly over-segmented the data. Over four times as many expected segments had been detected for MGBMINIREF. A minimum state duration was enforced and this helped to reduce the over segmentation. The optimum number of states was determined to be 30 with a duration filter of 25% for both datasets. Further filtering was performed to help the adaptation stage by removing those segments which had been resegmented, and reducing the amount of

time filtered every iteration. This latter technique helps to regain data. For RT07REF, a combination of the two techniques improved performance, however, this was not the case for MGBMINIREF. This is because the majority of segments were resegmented. Adapting on too little data does not give improvements, no matter how reliable the remaining segments are. The last techniques affected the decoding stage. Equal prior for the clusters was changed to priors based on the size of the data assigned to each cluster. This did not work as well as equal priors. Furthermore, the grammar scale factor which multiplies the cluster probabilities did help, with a different value noted for the two datasets.

With the best performing implementation determined for both, an automatic stopping criterion was investigated as opposed to performing the method on each recording for the same number of iterations. It showed that the largest improvements were seen in the first 5 iterations, and at 5 iterations the number of speakers detected settles and does not drop further. When comparing the automatic stopping criterion to using 5 fixed iterations, the former achieves the same or better performance at a cost of up to 49% more iterations. However, when manually selecting the best iterations for each recording, the error is lower than using 50 iterations per recording. This means that the best possible performance can be obtained through using an ideal automatic stopping criterion.

Finally, experiments are performed with SAD segmentation to see how the DNNCLU method behaves in real situations. For meeting data and the TBL dataset, it is shown that using the *SAD6* DNN along with the DNNCLU method outperforms the toolkit performances. However, this is not seen for the broadcast media data from the MGB challenge. As mentioned before, this is due to the lack of training data available for the ssDNN. It is expected that given an equivalent amount of speaker labelled MGB or broadcast media data, the performance would improve.

For the RT07 results to be comparable to other research, the SHEF scoring setup is compared to the NIST scoring setup. Results with the DNNCLU method using the automatic stopping criterion are displayed in Table 6.12. The NIST performance in fact produces a result which is not lower than the SHoUT result. The FA has improved by 65%, however, the MS and SE results have doubled by around 50%. This results in an overall DER score which gives nearly 40% worse performance than the baseline. The reason for this could be due to the fact that the method was optimised for the SHEF reference segments which are different from the NIST segments. Along with the poor results on the MGB datasets, this shows that diarisation using this method is challenging across different datasets.

A last remark about this proposed DNN-based clustering metric is about the processing time. The method requires a DNN to be adapted every iteration as well as decoding to be performed. Considering meetings from the RT07 dataset, the DNNCLU method takes around

Table 6.12 Overall RT07 results comparing the two scoring setups: SHEF (manual reference and collar of 0.05s) and NIST (specific portions of the evaluation reference with a collar of 0.25s).

System	RT07 (SHEF)				RT07 (NIST)			
	MS%	FA%	SE%	DER%	MS%	FA%	SE%	DER%
SHoUT	7.4	27.7	14.4	49.5	3.9	9.2	12.3	25.3
DNNCLU(A)	10.6	6.4	24.8	41.9	6.3	3.2	25.2	34.8

40% longer to complete than the time it takes for standard diarisation methods, in this case SHoUT. This was computed assuming a fixed number of 5 iterations. In the experiment where the method was continued for 20 iterations, this increased to 85% more time. When processing time matters, this method may be overlooked. However, with further work this method could be improved to reach an acceptable performance with fewer iterations, making the processing time more comparable to standard methods.

6.4 Summary

The chapter presents a novel approach to clustering for speaker diarisation in which DNNs are applied. Despite being discriminant classifiers and therefore not clusterers by definition, this proposed method of adapting on limited data indirectly causes the DNNs to behave like they cluster. DNNs have not been applied to clustering in a semi-supervised fashion, however they have been applied to other stages in a diarisation system: feature processing (Yella and Stolcke, 2015; Yella et al., 2014) and SAD (Dines et al., 2006).

An ssDNN is required and must be trained beforehand. This makes the approach semi-supervised in nature. This pretrained model is trained on any data which contains speaker labels. The ssDNN is trained to separate speakers and captures this information in the bottleneck layer. For each recording, a DNN is reconstructed from the ssDNN by removing the final layer and replacing with a randomly initialised final layer, where the number of target classes is the number of speakers in the input segments. An iteration of adaptation is performed before the segments are decoded to update their speaker labels. A stopping criterion is presented which achieves the same or better results over simply using a fixed number of iterations.

Experiments began by using reference segments to determine the optimum setup which gave the lowest SE. Methods were necessary to filter the segments used for the adaptation stage to use the more accurate segments and for the decoding stage to tune the output. Evaluating on both reference segmentation of RT07 and MGB datasets from different domains

shows how similar the best setup seems to be, however RT07 performed better with an extra filtering step. Using SAD output segments, for meeting data the method outperforms the toolkit baseline which is also semi-supervised. However, due to the lack of data for the pretrained ssDNN applied to the MGB data the clustering method does not perform well.

Chapter 7

Conclusions and Future Work

Contents

The goals of this research was to investigate the disadvantages of the current metrics and propose a new evaluation metric, and to investigate the performance of semi-supervised methods incorporating three types of supplementary data: transcripts, IHM channels and speaker models. This chapter summarises the main contributions of the thesis, notes the publications which support the research and outlines suggestions for future work on this topic.

7.1 Contributions

The introduction outlined three objectives for this thesis and these are detailed in Section 1.4. The thesis began by discussing the current state-of-the-art in the field of speaker diarisation in Chapter 2. This was followed by Chapter 3 in which the challenges to a system are investigated in terms of data domains and negative effects. The training and test sets were presented and evaluated using three described public domain toolkits. Chapter 4 focussed on Objective 1, which aimed to investigate a segment-based evaluation metric as

an alternative to the popular time-weighted DER. Objective 2 and Objective 3 focussed on semi-supervised diarisation techniques. The former is detailed in Chapter 5 and set out to investigate incorporating supplementary data in the form of transcripts and IHM channels into systems. Chapter 6 investigates the latter which focussed on speaker models as supplementary data for a DNN-based clustering technique.

7.1.1 Segment-oriented Evaluation

The first objective was investigated in Chapter 4 and an alternative evaluation metric was presented, the Segment F-measure (SEGF). Instead of being duration-based or focussing on boundaries, it considered the segments themselves. Hypothesis segments were correctly matched if their boundaries and cluster label corresponded to a reference segment. The PRC and RCL metrics were applied which led to the F-measure. A speaker mapping algorithm was presented which considered the lowest cost as opposed to a greedy search. It was shown that:

- The DER, DPC, BNDF and K metrics masked the segmentation quality by achieving high performance when the SEGF was low. The SEGF gave a deeper understanding of the quality of the segments from a segment-oriented perspective.
- The DER typically applies a collar of 0.25s whereas a smaller collar was required for the SEGF as the scores increase at a slower rate after 0.1s. The collar does not remove time for the evaluation which happens with the DER.
- A distribution around the reference boundaries is applied and, depending on the distribution, bounded by the specified collar. This gives a probability on the reference boundary which led to more leniency in the decision. Distributions investigated were the uniform, triangular and Gaussian.
- It was hard to achieve a high performance with the SEGF due to the difficult nature of matching segments as opposed to duration or boundary matching. When the segmentation quality was poor the scores did not reach above 1.5% which showed its limitations.
- The metric is sensitive to the segmentation quality, computes reproducible results and has an acceptable computation time.

7.1.2 Speaker Diarisation with Auxiliary Information

The second objective was investigated in Chapter 5. Five semi-supervised diarisation methods were presented incorporating supplementary data such as transcripts and IHM channels. Method 1 aligned transcripts to SDM channels. Method 2 performed SAD on SDM channels using DNN models before aligning the speech segments to IHM channels and used the resulting frame posterior probabilities for speaker labelling. Method 3 and 4 built DNNs trained on concatenated IHM channels and did not use SDMs as input. Lastly, Method 5 applied all three types of input. It combined the SAD from Method 2, transcript alignment from Method 1 and compared the speaker labelling in Method 2 with looking at energy across the IHM channels. Experiments investigated the performance of the methods and, when necessary, attempted to reduce the negative effects of crosstalk and overlapping speech. It was shown that:

- Transcript alignment in Method 1 outperformed the baseline on the SDM channels despite the known imperfections in the transcripts.
- Methods 1 and 2 struggled to give acceptable performance using IHM channels. The transcript alignment and the DNN-based SAD models suffered due to the negative effects of crosstalk in the data. It was possible to incorporate crosstalk features (Dines et al., 2006) into the trained DNN models which led to reductions in error.
- For Method 2, the DNN models trained for SAD worked well on the SDM channels. Two models trained using TBLTRAIN-SDM data, one of which included AMI-IHM data too, achieved a DER of 2.9% for the TBL dataset.
- When aligning speech segments on IHM channels, the resulting frame posterior probabilities were successful for speaker labelling. Furthermore, in Method 5 the posterior probabilities were shown to outperform the technique considering energy across the IHMs.
- Method 3 and 4 gave the best performances overall and did not include transcripts or SDM channels. DNNs were trained on concatenated features extracted from IHM channels. However, mixed results were seen when investigating training with or without overlap and with or without crosstalk features.
- For TBL data, Method 3 requiring a fixed number of channels outperformed Method 4, however the latter is applicable to more datasets.

- For Method 4, a bias against detecting nonspeech was applied and using simple counts, as opposed to posterior probabilities, for the frame decision step led to improved results.
- For Method 5, combining both transcripts, IHMs and SDMs as input gave better performance than either transcripts in Method 1 or IHMs with SDMs in Method 2. Incorporating more supplementary data leads to higher performance in this case.

7.1.3 DNN-based Speaker Clustering

The final third objective was investigated in Chapter 6. A DNN-based clustering technique was presented and investigated for both meeting and broadcast media datasets. The iterative method required a pretrained ssDNN from which a new DNN was reconstructed and adapted to filtered segments. A confidence-based stopping criterion was presented as opposed to stopping after a certain number of iterations. The technique was tuned using reference segments before applying SAD segments and evaluating. It was shown that:

- The structure of the ssDNN was important, and the best performance was seen with 4 layers of 1745 neurons and a bottleneck layer with 13 dimensions. This was the widest and deepest structure tested and the smallest bottleneck.
- Despite tuning on speaker-pure reference segments, resegmenting the data was seen to improve the performance as opposed to determining one speaker label per segment.
- More improvement was gained by filtering 25% of the time removing the shortest segments rather than filtering segments based on low-confidence.
- Applying a minimum state duration enforced segments to be of a certain duration which limited the number of detected segments and reduced over-segmentation.
- Further filtering was investigated by reducing the duration filter and removing the speech segments from the adaptation stage which were labelled with more than one speaker. This benefited the meeting domain but not the broadcast media domain.
- Changing the grammar scale factor did improve the performance and it was different for each domain.
- The confidence-based automatic stopping criterion achieved similar or better results to using fixed iterations. The former is the better choice as has the ability to achieve better results given perfect conditions.

- When considering SAD segments for RT07 and TBL data, the method outperformed the baseline. However, further research is necessary in terms of the broadcast media domain as the technique did not perform well for the MGB data. This was largely because of the lack of speaker labelled in-domain training data.

7.2 List of Publications

Research included in the thesis is presented chronologically. Research in (Milner et al., 2015) and (Saz et al., 2015) presents the diarisation and transcription systems submitted to the MGB challenge, from which came the DNN-based models for SAD applied in Section 6.3.3.7 specifically for the MGB datasets. Work in (Milner and Hain, 2016b) initially proposed the SEGF presented in Chapter 4 and work in (Milner and Hain, 2016a; Milner et al., 2015) began initial research into the DNN-based clustering technique presented in Chapter 6. Work in Hain et al. (2016) included the MGB diarisation system to an online platform known as webASR. Finally, the research in (Milner and Hain, 2017) presents Method 3 and 4 from Chapter 5, described in Section 5.3.2 and Section 5.3.3 respectively.

- R. Milner, O. Saz, S. Deena, M. Doulaty, R. Ng, and T. Hain, "The 2015 Sheffield System for Longitudinal Diarisation of Broadcast Media," in Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2015.

- O. Saz, M. Doulaty, S. Deena, R. Milner, R. Ng, M. Hasan, Y. Liu, and T. Hain, "The 2015 Sheffield System for Transcription of Multi–Genre Broadcast Media," in Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2015.

- R. Milner and T. Hain, "Segment-oriented evaluation of speaker diarisation performance," in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016.

- R. Milner and T. Hain, "DNN-based speaker clustering for speaker diarisation," in Proceedings of the 17th Annual Conference of the International Speech Communication Association (INTERSPEECH), 2016.

- T. Hain, J. Christian, O. Saz, S. Deena, M. Hasan, R. Ng, R. Milner, M. Doulaty and Y. Liu, "webASR 2 — Improved Cloud Based Speech Technology," in Proceedings of the 17th Annual Conference of the International Speech Communication Association (INTERSPEECH), 2016.

- R. Milner and T. Hain, "DNN approach to speaker diarisation using speaker channels," in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017.

7.3 Future Work

Chapter 4 proposed an alternative evaluation metric for scoring speaker diarisation performance. Distributions are applied around the reference boundaries and a collar is applied to account for the uncertainty in the reference. A straight-forward uniform distribution is compared to the Gaussian and triangular distributions. Further work into leniency on one side of the boundary is of interest. For example, it is arguably more important to detect all the speech rather than missing a region, therefore a nonspeech-speech boundary would need to be more lenient on the left-hand-side to respect this. Furthermore, padding is described in Section 4.2.2 as a way to compensate for uncertainty in the hypothesised boundaries, however, it was not implemented in the final metric used for the experiments. Adding a padding affects the segment matching equations detailed in Appendix A. An investigation into whether padding is necessary and whether distributions, other than a uniform, around the boundaries is beneficial is of interest. Finally, the speaker mapping algorithm relies on initially matched reference and hypothesised segments. If the segmentation is poor with few segments matched, then this speaker mapping technique will lead to an inaccurate mapping. A method is necessary to prevent this from happening and would greatly improve the metric.

In Chapter 5, five methods were investigated which incorporated supplementary data into speaker diarisation systems. Given the final results in Figure 5.17, it is clear that some methods work well and others need improving. Method 1 performs transcript alignment which is heavily dependent on the quality of the transcripts. This method would achieve better performance given more precise transcripts. Fox and Hain (2013) and Stan et al. (2016) have shown how transcripts can be aligned and repaired when data is missing or there is a lack of timing information. Method 2 uses DNN-based SAD models combined with an IHM-alignment technique to calculate posterior probabilities. The SAD models have worked well on the SDM channels but training on more data may lead to further improvements. An investigation into different frame scores could lead to improvement in the second stage. Method 3 and Method 4 concatenate IHM channels and train DNNs to detect nonspeech and channels. Method 4 is already an extension to Method 3, moving from requiring a fixed number of channels to any number of channels. The crosstalk features (Dines et al., 2006; Wrigley et al., 2005) applied to reduce this data effect have had a mixed success rate leading to further work required to consistently address its negative effects. Additionally, overlap

is a major contributor to poor performance (Huijbregts and Wooters, 2007; Knox et al., 2012). As well as training on nonspeech and the channels as speakers, introducing overlap into the model could help to detect the difficult regions on time (Huijbregts et al., 2009b). Furthermore, these two methods assume there is a single speaker per IHM, however datasets exist which contain a varying number per channel (Fox et al., 2013, 2016). Extending the technique to address this problem would help in multi-channel situations. Lastly, Method 5 combines aspects of the previous methods which achieves the best performance for the meeting data but not for the broadcast media data. A technique combining Method 4 with transcript data, with reasonable quality, may lead to a better performance across the domains.

Chapter 6 presented a DNN-based clustering method which allowed for speaker segmentation, resegmentation, to be performed. The broadcast media domain datasets from the MGB challenge have not been successful with this method. This was shown to be a cause of the lack of training data for the pretrained ssDNN. The majority of the training data was meeting data, a data mismatch. Further work is required to determine the necessary amount of in-domain training data for the system to be successful. The current ssDNNs are trained on speaker data. It would be interesting to add silence or nonspeech component to the model as occurred in the channel detection DNNs for Method 3 and 4 presented in Chapter 5. If detecting nonspeech as well as speakers, any FA detected in the SAD segmentation would be salvageable. Furthermore, to recover any MS, a ssDNN trained with nonspeech allows for both the nonspeech and speech segments to be presented to the system, as opposed to just the speech segments. As well as nonspeech, overlap models have shown promise when trained on data surrounding speaker changes (Huijbregts et al., 2009b) and previously mentioned. Incorporating an ability to detect overlap would allow prevention of overlap in the input segmentation as overlap is known to be detrimental to clustering (Huijbregts and van Leeuwen, 2012; Knox et al., 2012). Currently, the clustering and speaker segmentation stages are performed in parallel, causing the system to be integrated as opposed to a step-by-step style. A separate SAD system is necessary to provide the initial segmentation which contains segments with unique speaker labels, i.e. SPKR1, SPKR2, etc. However, the ssDNN, if trained to classify nonspeech and speakers, could act as a SAD model where any time detected as a specific speaker can be labelled as speech. This would allow for a full diarisation system to be based on a pretrained ssDNN.

7.4 Summary

This thesis addressed two aspects of the speaker diarisation field: evaluation and semi-supervised methods. As there are several issues with the DER and other metrics, an alternative

segment-based metric was proposed and evaluated which led to a deeper insight into the segmentation quality. Transcripts and IHMs are common across datasets so five methods incorporating a combination of different inputs were investigated. Speaker models such as the ssDNN are pretrained and usable for any data set and domain. The DNN-clustering technique was investigated as a semi-supervised diarisation method. Several contributions resulted from each objective were discussed and suggestions of future work in this topic were presented.

Appendix A

Segment Matching for SEG-F

This work was completed with Thomas Hain. To score speaker diarisation using the SEGF, reference speaker labelled segments and hypothesis speaker labelled segments are necessary. However, both may contain some uncertainty. There are two types of uncertainty given the two types of boundaries:

1. Uncertainty in the reference leading to a collar, c, around reference boundaries.
2. Uncertainty in the hypothesis leading to padding, w, around hypothesis boundaries.

The scoring can be formalised as:

$$P(correct|ref,hyp) \cong P[ref-c<hyp<ref+c] \tag{A.1}$$

where ref is the boundary in the reference and hyp is the boundary in the hypothesis. The collar, but not the padding, is considered in this representation. However, as both the ref and the hyp may contain uncertainty, the true, unknown boundaries are represented as r and h. Equation A.1 is redefined where independence is assumed:

$$P[r-c<h<r+c] = \int_{-\infty}^{\infty}\int_{r-c}^{r+c} P_R(r)P_H(h)drdh \tag{A.2}$$

$$= \int_{-\infty}^{\infty}\left[\int_{-\infty}^{r+c} P_R(r)P_H(h)drdh - \int_{-\infty}^{r-c} P_R(r)P_H(h)drdh\right] \tag{A.3}$$

$$= \int_{-\infty}^{\infty} P_R(r)\left[\int_{-\infty}^{r+c} P_H(h)dh - \int_{-\infty}^{r-c} P_H(h)dh\right]dr \tag{A.4}$$

$$= \int_{-\infty}^{\infty} P_R(r)\Big[F_H(r+c) - F_H(r-c)\Big]dr \tag{A.5}$$

Three different distributions are proposed to compensate for uncertainty around the reference boundaries. These are: uniform, SEGF, triangular, t-SEGF, and Gaussian, g-SEGF.

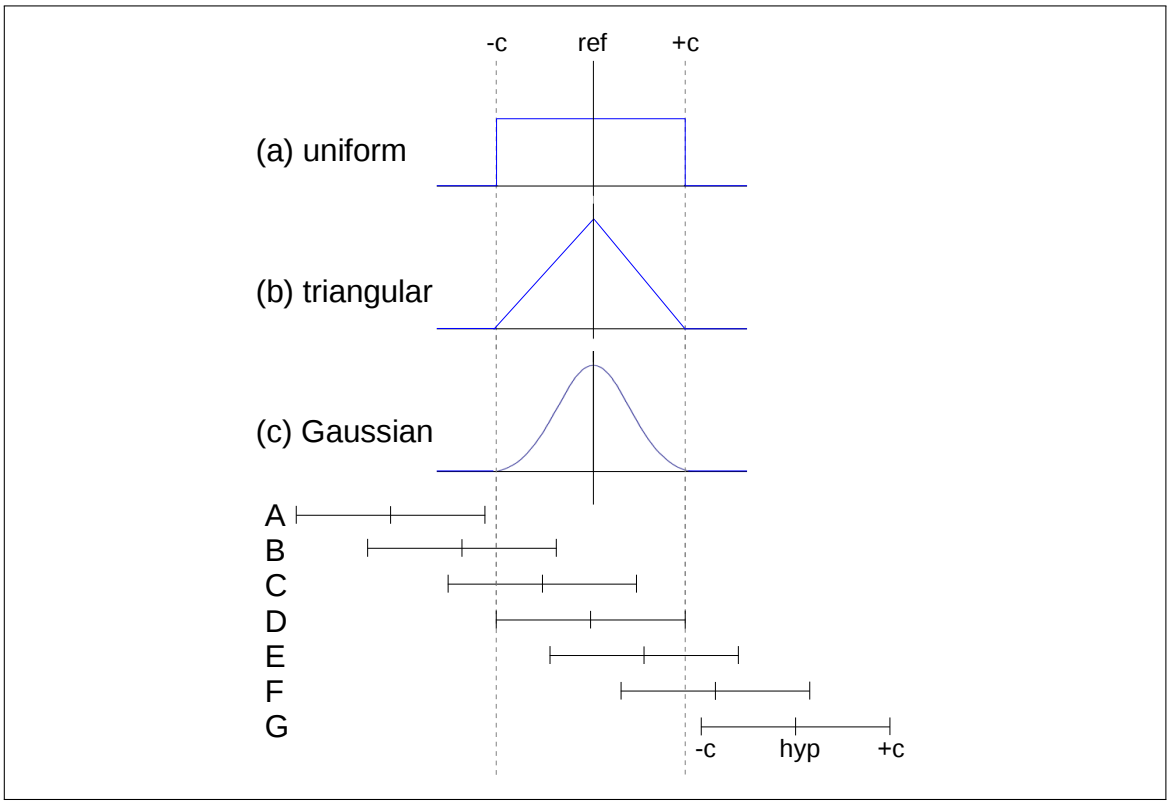

Fig. A.1 There are seven cases in which a segment boundary may fall around the reference when applying a collar and padding to compensate for uncertainty.

Figure A.1 displays an example of the distribution and the seven possible conditions for the position of the hypothesis boundary when compared to the reference. The equations for each are presented when considering uncertainty around the reference boundary using the collar. Uncertainty around the hypothesis is not applied.

Example 1: Uniform distribution

The uniform distribution around the reference boundary is defined to have a height of $\frac{1}{2c}$. The step function for F_H, defined in Equation A.5, can be expanded:

$$\int_{-\infty}^{\infty} P_R(r)\Big[F_H(r+c)-F_H(r-c)\Big]dr = \int_{hyp-c}^{\infty} P_R(r)dr - \int_{hyp+c}^{\infty} P_R(r)dr \tag{A.6}$$

$$= [1-F_R(hyp-c)] - [1-F_R(hyp+c)] \tag{A.7}$$

$$= F_R(hyp+c) - F_R(hyp-c) \tag{A.8}$$

where the true reference r is expressed as $hyp \pm c$. As seen in Figure A.1, the plot is equivalent to:

$$\text{Figure A.1(a)} = \begin{cases} 0, & r < ref - c \\ \frac{1}{2} + \frac{1}{2c}(r - ref), & ref - c < r < ref + c \\ 1, & r > ref + c \end{cases} \tag{A.9}$$

where ref is the reference boundary. This leads to the equations of the seven cases depicted in the figure:

A. $= \varnothing$

B.

$$\begin{aligned} F(hyp + c) &= \frac{1}{2} - \frac{1}{2c}((hyp + c) - ref) \\ &= \frac{1}{2} - \frac{1}{2c}(hyp - ref + c) \\ &= 1 - \frac{1}{2c}(hyp - ref) \end{aligned}$$

C. $\cong$ B.

D. $=$ B.

E.

$$\left.\begin{aligned} F(hyp + c) &= 1 \\ F(hyp - c) &= \frac{1}{2} + \frac{1}{2c}(hyp - c - ref) \\ &= \frac{1}{2c}(hyp - ref) \end{aligned}\right\} 1 - \frac{1}{2c}(hyp - ref)$$

F. $\cong$ E.

G. $= \varnothing$

This is subject to these three conditions:

$$hyp + c < ref - c: \qquad ref - hyp > 2c \tag{A.10}$$

$$hyp - c < ref + c: \qquad ref - hyp < -2c \tag{A.11}$$

$$else: \qquad 1 - \frac{1}{2c}|hyp - ref| \tag{A.12}$$

Example 2: Triangular distribution

For the triangular distribution, Equation A.6 from the uniform distribution example holds and the triangular plot in Figure A.1 can be represented by:

$$\text{Figure A1.(b)} = \begin{cases} \left[\begin{array}{l} \frac{1}{2c} + \frac{1}{2c^2}(r - ref) \\ \frac{1}{2c} - \frac{1}{2c^2}(r - ref) \end{array}\right] \frac{1}{2c} - \frac{1}{2c_2}|r - ref| \\ \varnothing \end{cases} \tag{A.13}$$

Two conditions exist for r. When $r < \varnothing$:

$$\int_{ref-c}^{r} P_R(r)dr = \frac{1}{2c}\int_{ref-c}^{r} 1 + \frac{1}{c}(r - ref)dr \tag{A.14}$$

$$= \frac{1}{2c}\left[\left(1 - \frac{ref}{c}\right)r + \frac{1}{c}\frac{r^2}{2}\right]_{ref-c}^{r} \tag{A.15}$$

$$= \frac{1}{2c^2}(c - ref)r + \frac{1}{4c^2}r^2 + \frac{1}{2c^2}(ref - c)^2 - \frac{1}{4c^2}(ref - c)^2 \tag{A.16}$$

$$= \frac{1}{4c^2}[r^2 + (ref - c)^2 - r(ref - c)] \tag{A.17}$$

$$= \frac{[r - (ref - c)]^2}{4c^2} \tag{A.18}$$

Secondly, when $r > \varnothing$:

$$\int_{-ref-c}^{r} P_R(r)dr \cong 1 - \frac{[-r - (ref - c)]^2}{4c^2} \tag{A.19}$$

This leads to $F_R(r)$ being represented as below given different situations for the boundaries:

$$F_R(r) = \begin{cases} 0 & r < ref - c \\ \dfrac{(r - (ref - c))^2}{4c^2} & ref - c < r < ref \\ 1 + \dfrac{(r + (ref - c))^2}{4c^2} & ref < r < ref + c \\ 0 & r > ref + c \end{cases} \tag{A.20}$$

The same division of cases is seen in Example 1 and this leads to the equations for the condition $r < \varnothing$:

A. $\varnothing$

B. $F(hyp+c) = \dfrac{(hyp+c-ref+c)^2}{4c^2} \qquad F(hyp-c) = \varnothing$

C. $F(hyp+c) = 1 - \dfrac{(hyp+c+ref-c)^2}{4c^2} \qquad F(hyp-c) = \varnothing$

D. 1

E. $F(hyp+c) = 1 \qquad F(hyp-c) = \dfrac{(hyp-c-ref-c)^2}{4c^2}$

F. $F(hyp+c) = 1 \qquad F(hyp-c) = 1 - \dfrac{(hyp-c+ref-c)^2}{4c^2}$

G. $\varnothing$

For the condition when $r > \varnothing$, the equations become:

B. $F(hyp+c) = 1 + \dfrac{(hyp-ref)^2 + 4c(hyp-ref)}{4c^2}$

C. $F(hyp+c) = 1 - \left[\dfrac{(hyp+ref)^2}{4c^2}\right]$

E. $F(hyp-c) = 1 - \left[\dfrac{(hyp-ref)^2 - 4c(hyp-ref)}{4c^2} + 1\right]$

F. $F(hyp-c) = 1 - \left[1 - \dfrac{(hyp+ref)^2 - 4c(hyp-ref)}{4c^2} - 1\right]$

Example 3: Gaussian distribution

For the Gaussian distribution, the same equations as in Example 1 hold. The mean is represented by the reference boundary, $\mu = ref$, and the variance by a multiple of the collar, $\sigma = 3c$. The Gaussian for $P_R(r)$ is seen in Figure A.1(c) and is formulated as:

$$F_R = \frac{1}{2}\left[1 + erf\left(\frac{r-\mu}{\sigma\sqrt{2}}\right)\right] \tag{A.21}$$

There are no boundaries for this distribution, and the final equation, for all of the seven cases, is:

$$P(ref-c<hyp<ref+c) = erf\left(\frac{hyp+c-ref}{3c\sqrt{2}}\right) - erf\left(\frac{hyp-c-ref}{3c\sqrt{2}}\right) \tag{A.22}$$

where erf represents the error function.

References

Adami, A. G., Burget, L., Dupont, S., Garudadri, H., Grézl, F., Hermansky, H., Jain, P., Kajarekar, S. S., Morgan, N., and Sivadas, S. (2002). Qualcomm-icsi-ogi features for ASR. In *7th International Conference on Spoken Language Processing, ICSLP*.

Ajmera, J., Bourlard, H., Lapidot, I., and McCowan, I. (2002). Unknown-multiple speaker clustering using HMM. In *7th International Conference on Spoken Language Processing, ICSLP*.

Ajmera, J., McCowan, I., and Bourlard, H. (2004). Robust speaker change detection. *IEEE Signal Processing Letters*.

Ajmera, J. and Wooters, C. (2003). A robust speaker clustering algorithm. *IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU*.

Ajot, J. and Fiscus, J. (2009). RT-09 Speaker Diarization Results RT-09 Evaluation Participants. *NIST Rich Transcription Meeting Recognition Evaluation*.

Anguera, X., Wooters, C., and Hernando, J. (2007). Acoustic beamforming for speaker diarization of meetings. *IEEE Transactions on Audio, Speech & Language Processing*.

Anguera, X., Wooters, C., and Pardo, J. M. (2006). Robust speaker diarization for meetings: ICSI rt06s evaluation system. In *9th International Conference on Spoken Language Processing, ICSLP*.

Anguera, X., Wooters, C., Peskin, B., and Aguiló, M. (2005). Robust speaker segmentation for meetings: The ICSI-SRI spring 2005 diarization system. In *Machine Learning for Multimodal Interaction, Second International Workshop, MLMI*.

Aronowitz, H. (2010). Unsupervised compensation of intra-session intra-speaker variability for speaker diarization. In *Odyssey: The Speaker and Language Recognition Workshop*.

Aronowitz, H. (2011). Speaker diarization using a priori acoustic information. In *12th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Astudillo, R. F., Kolossa, D., Abad, A., Zeiler, S., Saeidi, R., Mowlaee, P., da Silva Neto, J. P., and Martin, R. (2013). Integration of beamforming and uncertainty-of-observation techniques for robust ASR in multi-source environments. *Computer Speech & Language*.

Athineos, M. and Ellis, D. P. W. (2003). Frequency-domain linear prediction for temporal features. *IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU*.

Barker, J., Marxer, R., Vincent, E., and Watanabe, S. (2015). The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines. In *IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU*.

Barras, C., Zhu, X., Meignier, S., and Gauvain, J. (2006). Multistage speaker diarization of broadcast news. *IEEE Transactions on Audio, Speech & Language Processing*.

Bell, P., Gales, M. J. F., Hain, T., Kilgour, J., Lanchantin, P., Liu, X., McParland, A., Renals, S., Saz, O., Wester, M., and Woodland, P. C. (2015). The MGB challenge: Evaluating multi-genre broadcast media recognition. In *IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU*.

Boakye, K., Vinyals, O., and Friedland, G. (2011). Improved overlapped speech handling for speaker diarization. In *12th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Bonastre, J., Scheffer, N., Matrouf, D., Fredouille, C., Larcher, A., Preti, A., Pouchoulin, G., Evans, N. W. D., Fauve, B. G. B., and Mason, J. S. D. (2008). Alize/spkdet: a state-of-the-art open source software for speaker recognition. In *Odyssey: The Speaker and Language Recognition Workshop*.

Bozonnet, S., Evans, N. W. D., and Fredouille, C. (2010a). The lia-eurecom rt'09 speaker diarization system: Enhancements in speaker modelling and cluster purification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*.

Bozonnet, S., Evans, N. W. D., Fredouille, C., Wang, D., and Troncy, R. (2010b). An integrated top-down/bottom-up approach to speaker diarization. In *11th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Brümmer, N. and de Villiers, E. (2010). The speaker partitioning problem. In *Odyssey: The Speaker and Language Recognition Workshop*.

Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., McCowan, I., Post, W., Reidsma, D., and Wellner, P. (2005). The AMI meeting corpus: A pre-announcement. In *Machine Learning for Multimodal Interaction, Second International Workshop, MLMI*.

Chen, S. and Gopalakrishnan, P. (1998). Speaker, environment and channel change detection and clustering via the bayesian information criterion. *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*.

Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*.

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., and Ouellet, P. (2011). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech & Language Processing*.

Delacourt, P. and Wellekens, C. (2000). DISTBIC: A speaker-based segmentation for audio data indexing. *Speech Communication*.

Deléglise, P., Estève, Y., Meignier, S., and Merlin, T. (2005). The LIUM speech transcription system: a CMU sphinx iii-based system for french broadcast news. In *9th European Conference on Speech Communication and Technology, EUROSPEECH*.

Deng, L. and Dong, Y. (2015). *Deep Learning: Methods and Applications*. Springer.

Dines, J., Vepa, J., and Hain, T. (2006). The segmentation of multi-channel meeting recordings for automatic speech recognition. In *9th International Conference on Spoken Language Processing, ICSLP*.

Dong, Y. and Deng, L. (2015). *Automatic Speech Recognition: A Deep Learning Approach*. Springer.

Doulaty, M., Saz, O., Ng, R. W. M., and Hain, T. (2016). Automatic genre and show identification of broadcast media. *17th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Dupuy, G., Rouvier, M., Meignier, S., and Estève, Y. (2012). I-vectors and ILP clustering adapted to cross-show speaker diarization. In *13th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Evans, N. W. D., Bozonnet, S., Wang, D., Fredouille, C., and Troncy, R. (2012). A comparative study of bottom-up and top-down approaches to speaker diarization. *IEEE Transactions on Audio, Speech & Language Processing*.

Evans, N. W. D., Fredouille, C., and Bonastre, J. (2009). Speaker diarization using unsupervised discriminant analysis of inter-channel delay features. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*.

Ferguson, T. S. (1973). A bayesian analysis of some nonparametric problems. *The annals of statistics*.

Ferras, M. and Bourlard, H. (2012). Speaker diarization and linking of large corpora. In *IEEE Spoken Language Technology Workshop, SLT*.

Fiscus, J. G., Ajot, J., Michel, M., and Garofolo, J. S. (2006). The rich transcription 2006 spring meeting recognition evaluation. In *Machine Learning for Multimodal Interaction, Third International Workshop, MLMI*.

Fox, C. and Hain, T. (2013). Lightly supervised learning from a damaged natural speech corpus. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*.

Fox, C., Liu, Y., Zwyssig, E., and Hain, T. (2013). The Sheffield wargames corpus. In *14th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Fox, C., Liu, Y., Zwyssig, E., and Hain, T. (2016). The Sheffield Wargame Corpus — Day Two and Day Three. In *17th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Fox, E. B., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. (2008). An HDP-HMM for systems with state persistence. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference ICML*.

Fox, E. B., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. (2011). A sticky HDP-HMM with application to speaker diarization. *The Annals of Applied Statistics*.

Fredouille, C., Bozonnet, S., and Evans, N. (2009). The LIA-EURECOM RT'09 Speaker Diarization System. *NIST Rich Transcription Meeting Recognition Evaluation*.

Friedland, G., Janin, A., Imseng, D., Miró, X. A., Gottlieb, L. R., Huijbregts, M., Knox, M. T., and Vinyals, O. (2012). The ICSI RT-09 speaker diarization system. *IEEE Transactions on Audio, Speech & Language Processing*.

Friedland, G., Vinyals, O., Huang, Y., and Müller, C. A. (2009). Prosodic and other long-term features for speaker diarization. *IEEE Transactions on Audio, Speech & Language Processing*.

Galibert, O. and Kahn, J. (2013). The first official REPERE evaluation. In *Proceedings of the First Workshop on Speech, Language and Audio in Multimedia*.

Galliano, S., Geoffrois, E., Gravier, G., Bonastre, J.-F., Mostefa, D., and Choukri, K. (2006). Corpus description of the ester evaluation campaign for the rich transcription of french broadcast news. In *Proceedings of LREC*.

Galliano, S., Gravier, G., and Chaubard, L. (2009). The ester 2 evaluation campaign for the rich transcription of french radio broadcasts. In *10th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Gauvain, J., Lamel, L., and Adda, G. (1998). Partitioning and transcription of broadcast news data. In *The 5th International Conference on Spoken Language Processing, Incorporating The 7th Australian International Speech Science and Technology Conference*.

Ghaemmaghami, H., Dean, D., Vogt, R., and Sridharan, S. (2012). Speaker attribution of multiple telephone conversations using a complete-linkage clustering approach. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*.

Gish, H., Siu, M.-H., and Rohlicek, R. (1991). Segregation of speakers for speech recognition and speaker identification. In *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*.

Gravier, G., Betser, M., and Ben, M. (2010). *AudioSeg: Audio Segmentation Toolkit, release 1.2*. IRISA.

Hain, T., Burget, L., Dines, J., Garner, P. N., Hannani, A. E., Huijbregts, M., Karafiát, M., Lincoln, M., and Wan, V. (2010). The AMIDA 2009 meeting transcription system. In *11th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Hain, T., Christian, J., Saz, O., Deena, S., Hasan, M., Ng, R. W. M., Milner, R., Doulaty, M., and Liu, Y. (2016). webasr 2 - improved cloud based speech technology. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*.

Han, K. J. and Narayanan, S. S. (2008). Novel inter-cluster distance measure combining GLR and ICR for improved agglomerative hierarchical speaker clustering. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*.

Harper, M. (2015). The automatic speech recogition in reverberant environments (aspire) challenge. In *IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU*.

Hasan, M., Doddipatla, R., and Hain, T. (2015). Noise-matched training of CRF based sentence end detection models. In *16th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *the Journal of the Acoustical Society of America*.

Hermansky, H. and Sharma, S. (1998). TRAPS - classifiers of temporal patterns. In *The 5th International Conference on Spoken Language Processing, Incorporating The 7th Australian International Speech Science and Technology Conference*.

Hieu, N. T. (2012). *Speaker diarization in meetings domain*. PhD thesis, Nanyany Technology University, Singapore.

Hinton, G., Deng, L., and Yu, D. (2012). Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*.

Huijbregts, M., Ordelman, R., van der Werff, L., and de Jong, F. M. G. (2009a). Shout, the university of twente submission to the n-best 2008 speech recognition evaluation for dutch. In *10th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Huijbregts, M. and van Leeuwen, D. A. (2012). Large-scale speaker diarization for long recordings and small collections. *IEEE Transactions on Audio, Speech & Language Processing*.

Huijbregts, M., van Leeuwen, D. A., and de Jong, F. M. G. (2009b). Speech overlap detection in a two-pass speaker diarization system. In *10th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Huijbregts, M., van Leeuwen, D. A., and Wooters, C. (2012). Speaker diarization error analysis using oracle components. *IEEE Transactions on Audio, Speech & Language Processing*.

Huijbregts, M. and Wooters, C. (2007). The blame game: performance analysis of speaker diarization system components. In *8th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Huijbregts, M. A. H. (2008). *Segmentation, diarization and speech transcription: surprise data unraveled*. PhD thesis, University of Twente, Netherlands.

Huijbregts, M. A. H. and van Leeuwen, D. A. (2010). Towards automatic speaker retrieval for large multimedia archives. In *In proceedings of Automated Information Extraction in Media Production*.

Imseng, D. and Friedland, G. (2009). Robust speaker diarization for short speech recordings. In *IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU*.

Imseng, D. and Friedland, G. (2010). Tuning-robust initialization methods for speaker diarization. *IEEE Transactions on Audio, Speech & Language Processing*.

Jain, P. and Hermansky, H. (1999). Improved mean and variance normalization for robust speech recognition. In *IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS SPEECH AND SIGNAL PROCESSING*.

Janin, A., Baron, D., Edwards, J., Ellis, D., Gelbart, D., Morgan, N., Peskin, B., Pfau, T., Shriberg, E., Stolcke, A., and Wooters, C. (2003). The ICSI meeting corpus. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*.

Jin, Q. and Schultz, T. (2004). Speaker segmentation and clustering in meetings. In *8th International Conference on Spoken Language Processing, ICSLP*.

Johnson, M. J. and Willsky, A. S. (2010). The hierarchical dirichlet process hidden semi-markov model. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, UAI*.

Jothilakshmi, S., Ramalingam, V., and Palanivel, S. (2009). Speaker diarization using autoassociative neural networks. *Engineering Applications of Artificial Intelligence*.

Kemp, T., Schmidt, M., Westphal, M., and Waibel, A. (2000). Strategies for automatic segmentation of audio data. In *IEEE International Conference on Acoustics, Speech, and Signal Processing. ICASSP*.

Kenny, P., Reynolds, D. A., and Castaldo, F. (2010). Diarization of telephone conversations using factor analysis. *Journal of Selected Topics in Signal Processing, J-STSP*.

Kim, H., Ertelt, D., and Sikora, T. (2005). Hybrid speaker-based segmentation system using model-level clustering. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*.

Kinoshita, K., Delcroix, M., Gannot, S., Habets, E. A. P., Haeb-Umbach, R., Kellermann, W., Leutnant, V., Maas, R., Nakatani, T., Raj, B., Sehr, A., and Yoshioka, T. (2016). A summary of the REVERB challenge: state-of-the-art and remaining challenges in reverberant speech processing research. *EURASIP Journal of Advanced Signal Processing*.

Knox, M. T., Mirghafori, N., and Friedland, G. (2012). Where did I go wrong?: Identifying troublesome segments for speaker diarization systems. In *13th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Konig, Y., Heck, L., Weintraub, M., Sonmez, K., and E, R. E. (1998). Nonlinear discriminant feature extraction for robust text-independent speaker recognition. In *In Proceedings Speaker Recognition and its Commercial and Forensic Applications*.

Kotti, M., Benetos, E., and Kotropoulos, C. (2008a). Computationally efficient and robust bic-based speaker segmentation. *IEEE Transactions on Audio, Speech & Language Processing*.

Kotti, M., Moschou, V., and Kotropoulos, C. (2008b). Speaker segmentation and clustering. *Signal Processing*.

Lamel, L., Gauvain, J., and Canseco-Rodriguez, L. (2004). Speaker diarization from speech transcripts. In *8th International Conference on Spoken Language Processing, ICSLP*.

Le, V.-B., Mella, O., and Fohr, D. (2007). Speaker diarization using normalized cross likelihood ratio. In *8th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

LeBlanc, J. P. and Leon, P. L. D. (1998). Speech separation by kurtosis maximization. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*.

Liu, Y., Zhang, P., and Hain, T. (2014). Using neural network front-ends on far field multiple microphones based speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*.

Luque, J., Anguera, X., Temko, A., and Hernando, J. (2007). Speaker diarization for conference room: The UPC RT07s evaluation system. In *Multimodal Technologies for Perception of Humans, International Evaluation Workshops CLEAR and RT*.

Meignier, S., Bonastre, J., Fredouille, C., and Merlin, T. (2000). Evolutive HMM for multi-speaker tracking system. In *IEEE International Conference on Acoustics, Speech, and Signal Processing. ICASSP*.

Meignier, S., Bonastre, J., and Igounet, S. (2001). E-HMM approach for learning and adapting sound models for speaker indexing. In *2001: A Speaker Odyssey - The Speaker Recognition Workshop*.

Meignier, S. and Merlin, T. (2010). LIUM_SpkDiarization: An Open Source Toolkit for Diarization. *in Proceedings of CMU SPUD Workshop*.

Meignier, S., Moraru, D., Fredouille, C., Bonastre, J., and Besacier, L. (2006). Step-by-step and integrated approaches in broadcast news speaker diarization. *Computer Speech & Language*.

Milner, R. and Hain, T. (2016a). Dnn-based speaker clustering for speaker diarisation. In *17th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Milner, R. and Hain, T. (2016b). Segment-oriented evaluation of speaker diarisation performance. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*.

Milner, R. and Hain, T. (2017). DNN approach to speaker diarisation using speaker channels. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*.

Milner, R., Saz, O., Deena, S., Doulaty, M., Ng, R. W. M., and Hain, T. (2015). The 2015 Sheffield system for longitudinal diarisation of broadcast media. In *IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU*.

Milner, R. M. (2012). *Multi-Recording Diarisation for BBC Broadcasts*. PhD thesis, University of Sheffield, UK.

Mirghafori, N. and Wooters, C. (2006). Nuts and flakes: a study of data characteristics in speaker diarization. In *IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP*.

Miró, X. A. (2006). *Robust Speaker Diarization for meetings*. PhD thesis, Universitat Politecnica de Catalunya, Spain.

Miró, X. A., Bozonnet, S., Evans, N. W. D., Fredouille, C., Friedland, G., and Vinyals, O. (2012). Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech & Language Processing*.

Moraru, D., Besacier, L., and Castelli, E. (2004a). Using a priori information for speaker diarization. In *ODYSSEY: The Speaker and Language Recognition Workshop*.

Moraru, D., Meignier, S., Besacier, L., Bonastre, J., and Magrin-Chagnolleau, I. (2003). The ELISA consortium approaches in speaker segmentation during the NIST 2002 speaker recognition evaluation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*.

Moraru, D., Meignier, S., Fredouille, C., Besacier, L., and Bonastre, J. (2004b). The ELISA consortium approaches in broadcast news speaker segmentation during the NIST 2003 rich transcription evaluation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*.

Mori, K. and Nakagawa, S. (2001). Speaker change detection and speaker clustering using VQ distortion for broadcast news speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*.

Nguyen, T. H., Chng, E., and Li, H. (2008). T-test distance and clustering criterion for speaker diarization. In *9th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Nguyen, T. H., Sun, H., Zhao, S., Khine, S. Z. K., Tran, D. H., Ma, T. L. N., Ma, B., Chng, E. S., and Li, H. (2009). The IIR-NTU speaker diarization systems for RT 2009. *NIST Rich Transcription Meeting Recognition Evaluation*.

Noulas, A. K., Englebienne, G., and Kröse, B. J. A. (2012). Multimodal speaker diarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, TPAMI*.

Otterson, S. (2008). *Use of speaker location features in meeting diarization*. PhD thesis, University of Washington, USA.

Pardo, J. M., Anguera, X., and Wooters, C. (2007). Speaker diarization for multiple-distant-microphone meetings using several sources of information. *IEEE Transactions on Computers*.

Pardo, J. M., Barra-Chicote, R., Segundo, R. S., de Córdoba, R., and Martínez-González, B. (2012). Speaker diarization features: The UPM contribution to the RT09 evaluation. *IEEE Transactions on Audio, Speech & Language Processing*.

Ramirez, J., Górriz, J. M., and Segura, J. C. (2007). *Voice activity detection. fundamentals and speech recognition system robustness*. InTech Open Access Publisher.

Reynolds, D. A. (2002). An overview of automatic speaker recognition technology. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*.

Reynolds, D. A., Kenny, P., and Castaldo, F. (2009). A study of new approaches to speaker diarization. In *10th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Reynolds, D. A., Quatieri, T. F., and Dunn, R. B. (2000). Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*.

Rouvier, M., Dupuy, G., Gay, P., el Khoury, E., Merlin, T., and Meignier, S. (2013). An open-source state-of-the-art toolbox for broadcast news diarization. In *14th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Rouvier, M. and Meignier, S. (2012). A global optimization framework for speaker diarization. In *Odyssey: The Speaker and Language Recognition Workshop*.

Saz, O., Doulaty, M., Deena, S., Milner, R., Ng, R. W. M., Hasan, M., Liu, Y., and Hain, T. (2015). The 2015 Sheffield system for transcription of multi-genre broadcast media. In *IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU*.

Senoussaoui, M., Kenny, P., Stafylakis, T., and Dumouchel, P. (2014). A study of the cosine distance-based mean shift for telephone speech diarization. *IEEE/ACM Transactions on Audio, Speech & Language Processing*.

Shriberg, E., Stolcke, A., and Baron, D. (2001). Observations on overlap: findings and implications for automatic processing of multi-party conversation. In *7th European Conference on Speech Communication and Technology EUROSPEECH*.

Shum, S., Campbell, W. M., and Reynolds, D. A. (2013). Large-scale community detection on speaker content graphs. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*.

Shum, S., Dehak, N., Chuangsuwanich, E., Reynolds, D. A., and Glass, J. R. (2011). Exploiting intra-conversation variability for speaker diarization. In *12th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Shum, S., Dehak, N., and Glass, J. (2012). On the use of spectral and iterative methods for speaker diarization. In *13th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Siegler, M. A., Jain, U., Raj, B., and Stern, R. M. (1997). Automatic segmentation, classification and clustering of broadcast news audio. *Proceedings of DARPA Speech Recognition Workshop*.

Sinclair, M. and King, S. (2013). Where are the challenges in speaker diarization? In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*.

Sinha, R., Tranter, S. E., Gales, M. J. F., and Woodland, P. C. (2005). The Cambridge University March 2005 speaker diarisation system. In *9th European Conference on Speech Communication and Technology, EUROSPEECH*.

Stafylakis, T., Katsouros, V., Kenny, P., and Dumouchel, P. (2012). Mean shift algorithm for exponential families with applications to speaker clustering. In *Odyssey: The Speaker and Language Recognition Workshop*.

Stan, A., Mamiya, Y., Yamagishi, J., Bell, P., Watts, O., Clark, R. A. J., and King, S. (2016). ALISA: an automatic lightly supervised speech segmentation and alignment tool. *Computer Speech & Language*.

Sun, H., Ma, B., Khine, S. Z. K., and Li, H. (2010). Speaker diarization system for RT07 and RT09 meeting room audio. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical dirichlet processes. *Journal of the American Statistical Association*.

Tran, V., Le, V. B., Barras, C., and Lamel, L. (2011). Comparing multi-stage approaches for cross-show speaker diarization. In *12th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Tranter, S. E. and Reynolds, D. A. (2004). Speaker diarisation for broadcast news. In *ODYSSEY: The Speaker and Language Recognition Workshop*.

Tranter, S. E. and Reynolds, D. A. (2006). An overview of automatic speaker diarization systems. *IEEE Transactions on Audio, Speech & Language Processing*.

Valente, F. and Wellekens, C. (2005a). Variational bayesian methods for audio indexing. In *Machine Learning for Multimodal Interaction, Second International Workshop, MLMI*.

Valente, F. and Wellekens, C. (2005b). Variational bayesian methods for audio indexing. In *Machine Learning for Multimodal Interaction, Second International Workshop, MLMI*.

van Leeuwen, D. A. (2010). Speaker linking in large data sets. In *Odyssey: The Speaker and Language Recognition Workshop*.

van Leeuwen, D. A. and Konecný, M. (2007). Progress in the AMIDA speaker diarization system for meeting data. In *Multimodal Technologies for Perception of Humans, International Evaluation Workshops CLEAR and RT*.

van Vuuren, V. Z., ten Bosch, L., and Niesler, T. (2013). A dynamic programming framework for neural network-based automatic speech segmentation. In *14th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Vaquero, C., Ortega, A., and Lleida, E. (2011). Partitioning of two-speaker conversation datasets. In *12th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Veselý, K., Burget, L., and Grézl, F. (2010). Parallel training of neural networks for speech recognition. In *11th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Vijayasenan, D. and Valente, F. (2012). Diartk : An open source toolkit for research in multistream speaker diarization and its application to meetings recordings. In *13th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Vijayasenan, D., Valente, F., and Bourlard, H. (2009). An information theoretic approach to speaker diarization of meeting data. *IEEE Transactions on Audio, Speech & Language Processing*.

Vijayasenan, D., Valente, F., and Bourlard, H. (2010). Multistream speaker diarization beyond two acoustic feature streams. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*.

Vijayasenan, D., Valente, F., and Bourlard, H. (2011). An information theoretic combination of MFCC and TDOA features for speaker diarization. *IEEE Transactions on Audio, Speech & Language Processing*.

Vijayasenan, D., Valente, F., and Bourlard, H. (2012). Multistream speaker diarization of meetings recordings beyond MFCC and TDOA features. *Speech Communication*.

Vinyals, O. and Friedland, G. (2008). Modulation spectrogram features for improved speaker diarization. In *9th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Wan, V. (2003). *Speaker verification using support vector machines*. PhD thesis, University of Sheffield, UK.

Wang, H. and Cheng, S. (2004). METRIC-SEQDAC: a hybrid approach for audio segmentation. In *ICSLP, 8th International Conference on Spoken Language Processing, ICSLP*.

Willsky, A. and Jones, H. (1976). A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Transactions on Automatic control*.

Wölfel, M., Yang, Q., Jin, Q., and Schultz, T. (2009). Speaker identification using warped MVDR cepstral features. In *10th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Wooters, C., Fung, J., Peskin, B., and Anguera, X. (2004). Towards robust speaker segmentation: The ICSI-SRI fall 2004 diarization system. *RT-04F Workshop*.

Wooters, C. and Huijbregts, M. (2007). The ICSI RT07s speaker diarization system. In *Multimodal Technologies for Perception of Humans, International Evaluation Workshops CLEAR and RT*.

Wrigley, S. N., Brown, G. J., Wan, V., and Renals, S. (2005). Speech and crosstalk detection in multichannel audio. *IEEE Transactions on Speech and Audio Processing*.

Yang, Q., Jin, Q., and Schultz, T. (2011). Investigation of cross-show speaker diarization. In *12th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Yella, S. H. and Bourlard, H. (2014). Overlapping speech detection using long-term conversational features for speaker diarization in meeting room conversations. *IEEE/ACM Transactions on Audio, Speech & Language Processing*.

Yella, S. H. and Stolcke, A. (2015). A comparison of neural network feature transforms for speaker diarization. In *16th Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Yella, S. H., Stolcke, A., and Slaney, M. (2014). Artificial neural network features for speaker diarization. In *IEEE Spoken Language Technology Workshop, SLT*.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X. A., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. (2006). *The HTK Book 3.4*. Cambridge University Engineering Department.

Zelenák, M., Schulz, H., and Hernando, J. (2012a). Speaker diarization of broadcast news in albayzin 2010 evaluation campaign. *EURASIP Journal of Audio, Speech and Music Processing*.

Zelenák, M., Segura, C., Luque, J., and Hernando, J. (2012b). Simultaneous speech detection with spatial features for speaker diarization. *IEEE Transactions on Audio, Speech & Language Processing*.

Zhang, P., Liu, Y., and Hain, T. (2014). Semi-supervised DNN training in meeting recognition. In *IEEE Spoken Language Technology Workshop, SLT*.