THE UNIVERSITY
of EDINBURGH

Speech Segmentation and Speaker Diarisation for Transcription and Translation

Mark Sinclair

Doctor of Philosophy
Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh
2015

Abstract

This dissertation outlines work related to Speech Segmentation – segmenting an audio recording into regions of speech and non-speech, and Speaker Diarization – further segmenting those regions into those pertaining to homogeneous speakers.

Knowing not only what was said but also who said it and when, has many useful applications. As well as providing a richer level of transcription for speech, we will show how such knowledge can improve Automatic Speech Recognition (ASR) system performance and can also benefit downstream Natural Language Processing (NLP) tasks such as machine translation and punctuation restoration.

While segmentation and diarization may appear to be relatively simple tasks to describe, in practise we find that they are very challenging and are, in general, ill-defined problems. Therefore, we first provide a formalisation of each of the problems as the sub-division of speech within acoustic space and time. Here, we see that the task can become very difficult when we want to partition this domain into our target classes of speakers, whilst avoiding other classes that reside in the same space, such as phonemes. We present a theoretical framework for describing and discussing the tasks as well as introducing existing state-of-the-art methods and research.

Current Speaker Diarization systems are notoriously sensitive to hyper-parameters and lack robustness across datasets. Therefore, we present a method which uses a series of oracle experiments to expose the limitations of current systems and to which system components these limitations can be attributed. We also demonstrate how Diarization Error Rate (DER), the dominant error metric in the literature, is not a comprehensive or reliable indicator of overall performance or of error propagation to subsequent downstream tasks. These results inform our subsequent research.

We find that, as a precursor to Speaker Diarization, the task of Speech Segmentation is a crucial first step in the system chain. Current methods typically do not account for the inherent structure of spoken discourse. As such, we explored a novel method which exploits an utterance-duration prior in order to better model the segment distribution of speech. We show how this method improves not only segmentation, but also the performance of subsequent speech recognition, machine translation and speaker diarization systems.

Typical ASR transcriptions do not include punctuation and the task of enriching transcriptions with this information is known as ‘punctuation restoration’. The benefit is not only improved readability but also better compatibility with NLP systems that expect sentence-like units such as in conventional machine translation. We show

how segmentation and diarization are related tasks that are able to contribute acoustic information that complements existing linguistically-based punctuation approaches.

There is a growing demand for speech technology applications in the broadcast media domain. This domain presents many new challenges including diverse noise and recording conditions. We show that the capacity of existing GMM-HMM based speech segmentation systems is limited for such scenarios and present a Deep Neural Network (DNN) based method which offers a more robust speech segmentation method resulting in improved speech recognition performance for a television broadcast dataset.

Ultimately, we are able to show that the speech segmentation is an inherently *ill-defined* problem for which the solution is highly dependent on the downstream task that it is intended for.

Acknowledgements

I would like to thank the following people who have helped me along the way:-

- My supervisor Prof. Simon King, for his mentorship and support, for always maintaining confidence in my ability and encouraging me to achieve my potential.
- Dr. Peter Bell, Prof. Steve Renals, Dr. Alexandra Birch and Dr. Mike Lincoln, for all the additional technical expertise, scientific debate and co-operation that contributed to the outcome of this thesis.
- My friends and colleagues from CSTR, past and present, all of whom have create such a wonderful working environment that has always been a pleasure to be part of.
- Erin, for all her support and kindness, for helping me to stay positive, for all the cups of coffee and keeping me fed, and for her truly incredible feat of patience over these years.
- My parents, for all the sacrifices they made to provide me with such great opportunities, for buying me that old BBC Micro all those years ago that started this adventure, and for never telling me to get a “proper job”!

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Mark Sinclair*)

Table of Contents

Notation and Nomenclature

Throughout this thesis we have endeavoured to use common notation and nomenclature from the speech and language processing community where possible. The following is a list of examples where this has either not been possible or where they pertain exclusively to the topics of speech segmentation or speaker diarization.

α	Acoustic feature scaling factor
δ	Max. segment length
$\ell(x)$	Function: acoustic model likelihood of observation x
η	Punctuated word-sequence
$\mathbb{B}$	Boolean domain $\{0,1\}$
$\mathbb{S}$	A set of segmentation sets
$\mathbb{T}$	Set of all real numbers on interval $[0,T]$
$\mathscr{L}$	Language vocabulary
$\mu_L()$	Function: Lebesgue measure
$\xi(S)$	Function: sum of duration of each segment in S
B	An utterance-break sequence
C	Punctuation token vocabulary
D	Sequence of durations between segments
DER	Diarization Error Rate
E_{clst}	Speaker-to-cluster difference

E_{spch}^{fa} Speech segmentation error - false alarm time

E_{spch}^{miss} Speech segmentation error - missed speech time

E_{spkr} Speaker segmentation error - speaker error time

E_{spkr}^{fa} Speaker segmentation error - false alarm time

E_{spkr}^{miss} Speaker segmentation error - missed speaker time

$I(S)$ Function: Converts segmentation set to equivalent interval set

K Number of clusters/speakers

S A segmentation set

SAD Speech Activity Detection; Voice Activity Detection (VAD); Speech Segmentation

T Total time

Chapter 1

Introduction

We begin by describing the tasks of Speech Segmentation – segmenting an audio recording into regions of speech and non-speech, and Speaker Diarization – further segmenting those regions into those pertaining to homogeneous speakers. These tasks form the basis for all of the research in this work. The general concept is illustrated in Figure 1.1 and each task is introduced in more detail in the subsequent sections.

1.1 What is Speech Segmentation?

The Task

The task of Speech Segmentation, also referred to as Speech Activity Detection (SAD) or Voice Activity Detection (VAD), is to segment a given speech recording into the regions that contain speech and those that contain non-speech.

The Implication

Speech Segmentation is an important precursory step that exists for almost every spoken language technology domain as it allows us to identify the data in a recording that is of most interest to tasks such as speech recognition, speech synthesis model training, discourse analysis, etc. By extracting contiguous segments of speech from a recording we are then able to apply processing where and when it is required, in a parallel fashion if desired, and disregard all other irrelevant data.

The Challenge

There are often many different ways to segment speech to achieve the same downstream performance. As a result, Speech Segmentation can be considered an ill-defined problem. Human transcribers, for example, may have conflicting perceptions regarding the start and end points of a segment or how much non-speech 'gap' there should be between segments for them to be considered distinct. Indeed, many such decisions regarding this task are subjective, meaning there is no hard consensus on what constitutes a 'segment'. The *ideal* speech segmentation may also depend on the subsequent task that it will be used for. For example, we may prefer short segments for presenting subtitles such that they fit on the screen, but longer segments for passing to an ASR system such that the language model can take more effect.

There are also acoustic challenges including environmental noise, channel conditions and reverberation. All of these aspects can affect the robustness and the

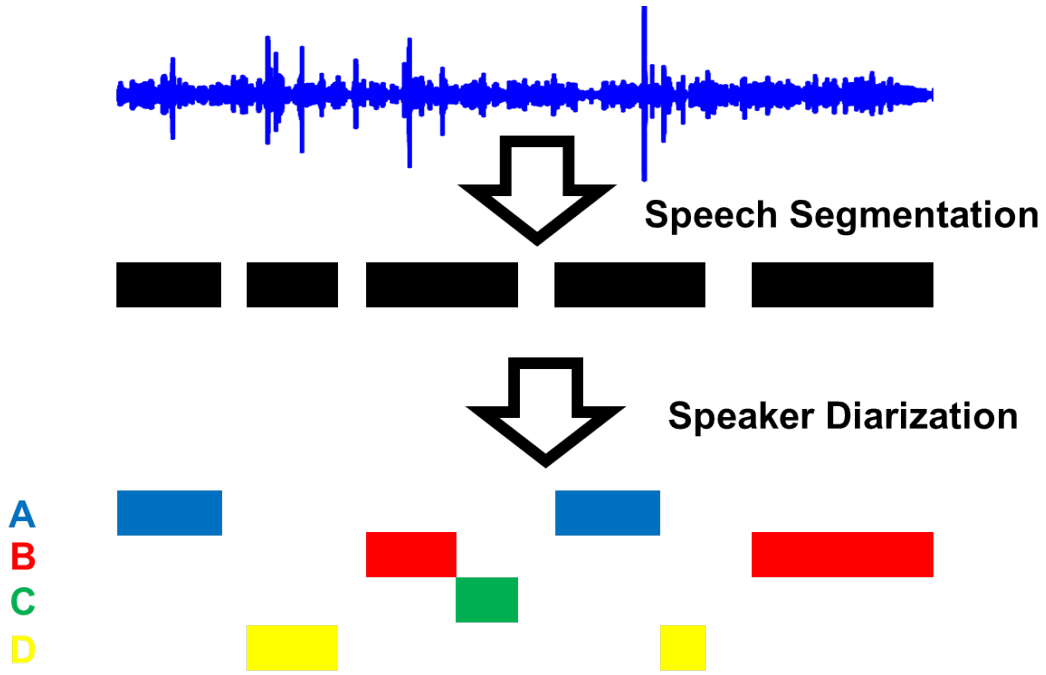


Figure 1.1: *Given an audio recording, we first segment it into the regions that contain speech (Speech Segmentation), then we further segment into the regions belonging to each speaker (Speaker Diarization).*

performance of acoustic models for speech and non-speech. We may also have to contend with speech-like noise, such as background speech, and audible non-speech, such as music. The decision of how to consider these special cases may depend on the task.

1.2 What is Speaker Diarization?

The Task

While Automatic Speech Recognition (ASR) systems continue to improve in accuracy and efficiency for the task of recognising word sequences, so too develops a demand for a richer transcription of the spoken dialogue. One of the most significant aspects for broadening the suite of information we can extract includes speaker diarization: the process of partitioning speech into homogeneous segments which belong to each speaker. Recent interest in speaker diarization among the research community is also evident, such as in associated tasks and evaluation campaigns as well as an increased presence of related work in publications and conference proceedings.

The Implication

Knowing not only *what?* was said but also *who?* said it and *when?*, has many significant applications. It can, for example, be extremely useful for colourising subtitles for broadcast media or for annotating the dialogue of meetings with the identity of each speaker. Such information can be used to search speech archives for quotations from specific speakers. Additionally, this knowledge can also benefit the speech recognition process as it allows for an informed adaptation or selection of models in order to tailor the system behaviour for individual speakers or speaker categories.

The Challenge

The canonical speaker diarization task is made difficult as the following knowledge typically cannot be exploited *a priori*:-

- The number of speakers
- Speaker-specific models/data
- Environmental conditions (room acoustics, noise, channel, etc.)

These conditions lead us towards an autonomous and robust solution. However, they can often be relaxed for experimental and investigative reasons, and may not always be strictly necessary for many practical applications.

1.3 Contribution

Now that we have outlined the motivations for working on the tasks of speech segmentation and speaker diarization, as well offering a brief description of each, we will present the main contributions of this thesis.

Speech segmentation and speaker diarization are very related problems but are not normally discussed with any common language. We present a formal theoretical framework that represents and relates both problems generally without assuming any particular practical method or application. This allows concepts and ideas for solutions to be discussed within the framework.

When we began working on this topic there were many examples of the lack of robustness for speaker diarization systems across datasets, even within the same domain. We presented a method to expose the specific challenges and short-falls of state-of-

the-art speaker diarization systems through a series of oracle experiments. This serves to inform future work and better target research to improve performance.

From this result, we found that speech segmentation has a significant impact on speaker diarization performance. In contrast to most conventional state-of-the-art speech segmentation methods, we present a novel approach that exploits a non-acoustic utterance-break prior to improve segmentation. We show that this improves not only speech segmentation performance but can be tuned to improve downstream tasks such as speech recognition, machine translation and speaker diarization.

We make a connection between acoustic segmentation and the task of punctuation restoration – adding punctuation marks to speech transcription – and present novel ways to combine such information with state-of-the-art linguistically motivated methods.

Finally, we explore the use of DNNs to improve the robustness of speech segmentation models for challenging audio environments such as broadcast television. We show that such methods can improve speech segmentation as well as downstream ASR performance.

The following peer-reviewed academic conference papers are direct products of the contents of this thesis (for full papers see Appendix A):-

- Sinclair and King (2013). Where are the challenges in speaker diarization? In *Proc. ICASSP*.

- Sinclair et al. (2014). A semi-Markov model for speech segmentation with an utterance-break prior. In *Proc. Interspeech*.

In addition, work from this thesis contributed to system components of the following (for full papers see Appendix B):-

- Driesen et al. (2013). Description of the UEDIN System for German ASR. In *Proc. IWSLT*.

- Bell et al. (2013). The UEDIN English ASR System for the IWSLT 2013 Evaluation. In *Proc. IWSLT*.

- Bell et al. (2015b). A system for automatic broadcast news summarisation, geolocation and translation. In *Proc. Interspeech*.

1.4 A Theoretical Framework

Essentially, all of the tasks we are trying to do revolve around the concept of segmenting an audio recording *session* – a contiguous event in time such as a meeting or a television programme – in various different ways. Here we will introduce a theoretical framework that establishes some fundamental concepts and notation that will facilitate all subsequent discussion of segmentation throughout the thesis.

1.4.1 Segmentations as Sets

Given a recording session of total time duration T, then $\mathbb{T}$ is the set of all real numbers that exist on the interval $[0,T]$. We can define t as a point on this continuous time-line where,

$$0 \leq t \leq T, t \in \mathbb{T} \subset \mathbb{R} \tag{1.1}$$

We can then consider an arbitrary function $f(t)$ that decides if a given condition is true or false at time t. In practice, the condition is typically whether or not certain acoustic phenomena are observed at that time – e.g. speech, a speaker, a noise condition, a channel condition, etc. This function should be a member of a set of functions F which itself is a subset of all functions that map from the domain $\mathbb{T}$ to the boolean co-domain $\mathbb{B} = \{0,1\}$.

$$f(t) \in F \subset \mathbb{T} \rightarrow \mathbb{B} \tag{1.2}$$

When implementing a segmentation system, the function set F can be thought of as all the possible functions $f(t)$ permitted by a given method.

For example, a very simple segmentation system method may consider the posterior likelihood of a distribution model represented by model parameter set θ, given some observation feature vector at time t, x_t. In this case, $f(t)$ could simply make a decision based on some threshold λ e.g.,

$$f(t) = \begin{cases} 1 & p(\theta|x_t) > \lambda \\ 0 & \text{otherwise} \end{cases} \tag{1.3}$$

Assuming the observations are fixed, then the function set F would therefore be all possible values of θ and λ. However, for the purposes of this framework we will think of $f(t)$ in more abstract terms and not consider in any detail methods that could produce the function in practice.

The actual segmentation derived from such a function, S, can then be defined as the set of all values of t where $f(t)$ is true:

$$S = \{t | f(t) = 1\} \tag{1.4}$$

As we are dealing with segments of a time-line, this can be equivalent to a set of disjoint intervals representing the start and end points of each segment (a_n, b_n), where $1 \leq n \leq N$ and N is the number of segments. Therefore, consider that we have a function $I(S)$ that converts a given segmentation set to its equivalent interval set,

$$S \equiv I(S) = \bigcup_{n=1}^{N} (a_n, b_n) \tag{1.5}$$

Often, we will need to know the total amount of time for which a segmentation function is true. We will define $\xi(S)$ to be a function that takes the Lebesgue measure $\mu_L()$ of the interval set equivalent of S,

$$\xi(S) = \mu_L(I(S)) = \sum_{n}^{N} (b_n - a_n) \tag{1.6}$$

This can be interpreted simply as the sum of the duration of all the individual segments.

1.4.2 Comparing Segmentation Sets

We will, of course, be interested in comparing and contrasting different segmentations of the same session. This is easily done with the proposed framework as we can make use of set operations. For example, if we had two segmentations S_A and S_B, and we wanted only the part that is unique to S_A then we could simply take the relative complement $S_A \setminus S_B$. We could also consider how much the pair differ by calculating their symmetric difference:

$$S_A \triangle S_B = (S_A \setminus S_B) \cup (S_B \setminus S_A) \tag{1.7}$$

If we then wanted to know how much time this constitutes, we can use the function from Eq. 1.6:

$$\xi(S_A \triangle S_B) \tag{1.8}$$

As we will see in Section 1.6, such operations become very useful when formulating metrics for evaluating system performance.

Figure 1.2: Speech Segmentation vs. Speaker Segmentation. The speech segmentation is the union of all speaker segmentations.

1.5 A Formal Definition of the Tasks

In this section we will use the framework established in 1.4 to formally define the tasks of speech segmentation and speaker diarization. We will begin with an illustrative example and then we will take each task in turn and make a formal definition using the framework established in Section 1.4.

1.5.1 Speech vs. Speakers

Before we discuss the tasks of both speech segmentation and speaker diarization in more detail, we will begin by making a distinction between what we will term *speech* and *speakers*. Consider the example shown in Fig. 1.2. Here we see the segmentation of a discussion between two speakers, each of which is represented by the green and blue blocks respectively. The speakers take turns but also overlap in time. We will refer to this as *speaker segmentation*, which is the goal of speaker diarization.

From this speaker segmentation we can infer the *speech segmentation*. This is simply the union of all speaker segments i.e. we only consider if there is speech or not, irrespective of the number of speakers. Note that we cannot inherently infer the speaker segmentation from a given speech segmentation alone.

1.5.2 Speech Segmentation

For speech segmentation we desire a function that will produce a segmentation of regions where speech and non-speech are observed. Consider a function $f_{spch}(t)$ where,

$$f_{spch}(t) = \begin{cases} 1 & \text{if speech at } t \\ 0 & \text{if non-speech at } t \end{cases} \tag{1.9}$$

We can therefore define S_{spch}, the set of all values of t where we observe speech,

$$S_{spch} = \{t | f_{spch}(t) = true\} \tag{1.10}$$

And the total speech time T_{spch} is simply,

$$T_{spch} = \xi(S_{spch}) \tag{1.11}$$

Which is bound by the total session time T as follows,

$$T_{spch} \leq T \tag{1.12}$$

The actual task of speech segmentation can therefore be defined as finding a function that produces an optimal segmentation according to some objective criteria. A typical example may be when we have a reference segmentation S_{ref} derived from a manual transcription of the session. In this case our objective is to find the best function $\hat{f}(t)$ that minimises the symmetric difference between S_{ref} and the segmentation implied by that function S,

$$\hat{f}(t) = \underset{f(t) \in F}{\arg\min} \xi(S \triangle S_{ref}), f(t) \implies S \tag{1.13}$$

As we mentioned in Section 1.4.1, the function space F will depend on the method chosen and any subsequent parameters that are available.

This is just one example of an objective criteria. We may be more concerned with how hypothesised speech segmentation propagates through a larger system to affect downstream tasks. In the case of speech segmentation that precedes speech recognition it may not matter how well the system matches manual transcriptions if it is able to provide comparable speech recognition performance. Indeed, we will show throughout this thesis that the optimal segmentation can differ depending on the ultimate task and in general it is an ill defined problem.

Given a development dataset of multiple sessions with reference segmentations we could attempt to select a function from this space that minimises the error globally.

1.5.3 Speaker Segmentation

The main difference with speaker segmentation is that we want to be able to produce a segmentation S_k for each speaker k, where $0 \leq k \leq K$, and K is the total number of speakers present in a meeting. Therefore, a given speaker segmentation can be thought of as being a set of segmentation sets $\mathbb{S} = \{S_1, \ldots, S_k\}$.

We can define the function $f_{spkr}(t,k)$ as,

$$f_{spkr}(t,k) = \begin{cases} 1 & \text{if speaker } k \text{ at } t \\ 0 & \text{otherwise} \end{cases} \tag{1.14}$$

And the set S_k of all values of t where we observe speaker k,

$$S_k = \left\{ t | f_{spkr}(t,k) = 1 \right\} \tag{1.15}$$

Where the total *speaker* time T_{spkr} is,

$$T_{spkr} = \sum_{k=0}^{K} \xi(S_k) \tag{1.16}$$

As we have seen in Fig. 1.2, speaker segments may overlap so the total speaker time can range from 0 (no speech/speakers) to $K \times T$ (all speakers overlapping at all times). Therefore the total *speaker* time is bounded as follows:

$$T_{spch} \leq T_{spkr} \leq K \times T \tag{1.17}$$

We will also introduce the notion of total speaker *overlap* time, T_{OL}, which we will define as the difference between T_{spkr} and T_{spch}.

$$T_{OL} = T_{spkr} - T_{spch} \tag{1.18}$$

It may also be useful to derive the associated speech segmentation from a speaker segmentation. This is simply the union of all speaker segmentations:

$$S_{spch} = \bigcup_{k=1}^{K} S_k \tag{1.19}$$

Speaker diarization can then be defined as the joint task of estimating the best value of K as well as the associated function $f_{spkr}(t,k)$ according to some objective criteria. Finding the optimal K can be thought of as a clustering problem – we have to cluster the data in a given session into homogeneous speakers. Whereas, finding the optimal $f_{spkr}(t,k)$ can be thought of as a segmentation problem – we want to attribute our speaker clusters to the correct regions of the session in time. For the canonical speaker diarization scenario, the target for both of these problems is generally unknown. This means that they need to be solved simultaneously and, as we will discover, there is often a trade-off between the optimisation of either problem. For example, getting the number of speakers wrong may result in an *overall* better segmentation function, or vice-versa.

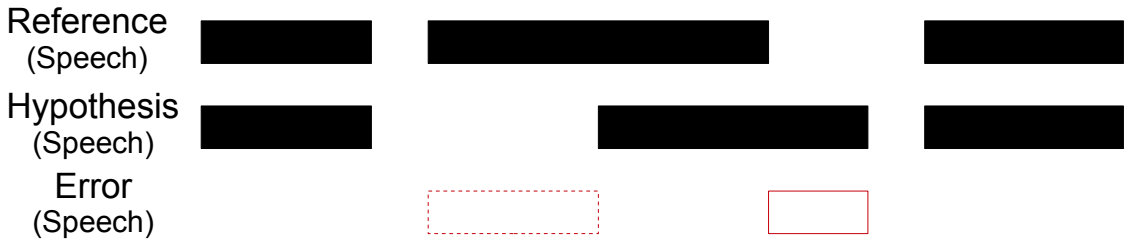


Figure 1.3: An example of speech segmentation error. The dashed-outline unfilled box represents missed speech while the solid-outline unfilled box represents a false alarm.

Similar to speech segmentation, the objective criteria could be to match a manual segmentation or it could be to optimise the performance of a downstream task such as speech recognition. Throughout the thesis, we will show that the ideal solution to this problem is also not always well defined and can vary significantly depending on the task.

1.6 Objective Criteria and Evaluation Metrics

As we have discussed in Section 1.4, there are many potential objective criteria for the application of speech segmentation and speaker diarization. Here we will provide an explanation of some objectives and evaluation metrics that can be used in the development of segmentation systems. Throughout the thesis, we will make use of these metrics to compare and contrast system performance.

1.6.1 Speech Segmentation: Speech Activity Detection (SAD) Error

If we have a reference speech segmentation available then we may be interested in knowing the difference between a given reference speech segmentation S^{ref} and a system hypothesis speech segmentation S^{hyp}. We can express this as the symmetric difference between the two segmentations, which can be further subdivided into two components: missed speech and false alarm speech.

$$S^{hyp} \triangle S^{ref} = (S^{ref} \backslash S^{hyp}) \cup (S^{hyp} \backslash S^{ref}) \tag{1.20}$$

Figure 1.3 shows an example of speech segmentation error, illustrating the difference between missed speech and false alarm.

The missed speech error component is the set of all value of time t where speech was observed in the reference but non-speech was hypothesised by the system. We can

define this as the relative complement of S^{ref} in S^{hyp}, that is,

$$S^{ref} \backslash S_{hyp} \tag{1.21}$$

This error is normally expressed as a fraction of the total speech time in the reference,

$$E^{miss}_{spch} = \frac{\xi(S^{ref} \backslash S^{hyp})}{T^{ref}_{spch}} \tag{1.22}$$

Conversely, the false alarm error component is the set of all values of time t where the system hypothesised speech but non-speech was observed in the reference. We can define this as the relative complement of S^{hyp} in S^{ref}, that is,

$$S^{hyp} \backslash S^{ref} \tag{1.23}$$

$$E^{fa}_{spch} = \frac{\xi(S^{hyp} \backslash S^{ref})}{T^{ref}_{spch}} \tag{1.24}$$

The total *Speech Error* E^{spch}, is simply the sum of these components.

$$E_{spch} = E^{miss}_{spch} + E^{fa}_{spch} = \frac{\xi(S^{ref} \triangle S^{spch})}{T^{ref}_{spch}} \tag{1.25}$$

These errors are referred to as *speech* time errors in the results computed by the NIST tools[1].

1.6.2 Speaker Segmentation: Diarization Error Rate (DER)

The main metric for speaker segmentation evaluation is the Diarization Error Rate (DER) which is a sum of three contributing factors: speaker error, false alarm speaker and missed speaker.

$$DER = E_{spkr} + E^{fa}_{spkr} + E^{miss}_{spkr} \tag{1.26}$$

Figure 1.4 shows an example of DER, illustrating how it differs from SAD error.

If we have a reference speaker segmentation set $\mathbb{S}^{ref}$ and a system hypothesis speaker segmentation set $\mathbb{S}^{hyp}$, then we are able to evaluate the system performance. Each set is actually a set of speaker segmentation sets, one for each speaker. For example, a reference speaker segmentation with K_{ref} speakers would be $\mathbb{S}^{ref} = \left\{ S^{ref}_1, \ldots, S^{ref}_{K_{ref}} \right\}$.

One aspect that makes evaluation of speaker segmentation more difficult than speech segmentation is that we may have a different number of reference speakers K_{ref} and

[1] http://www.itl.nist.gov/iad/mig/tools

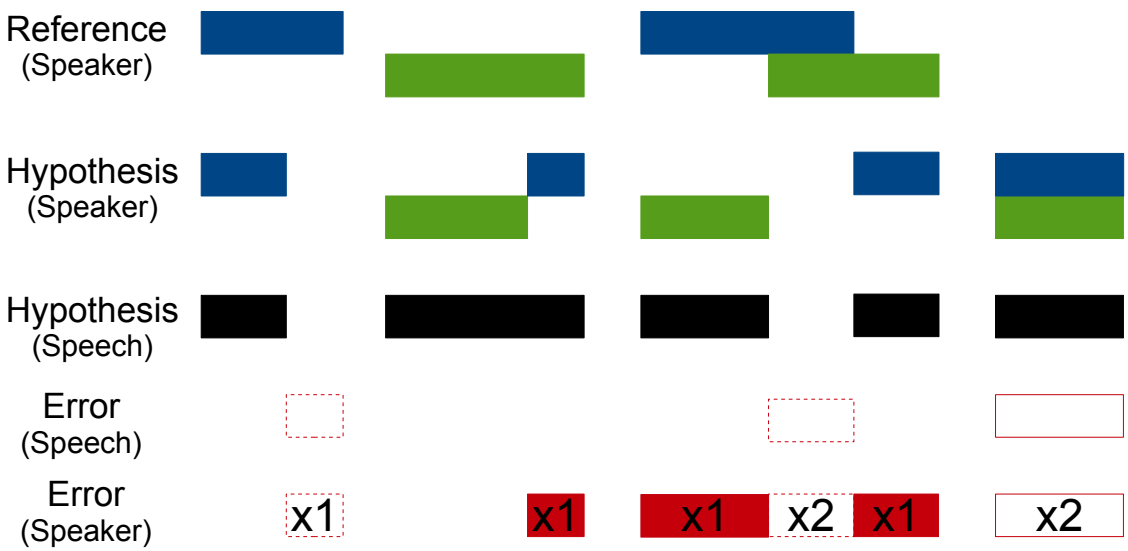


Figure 1.4: An example of speaker segmentation error. The dashed-outline unfilled boxes represent misses; the solid-outline unfilled boxes represent false alarms; and the filled boxes represent speaker errors. Note that, unlike SAD errors, DER errors can stack and this is represented by the multipliers shown inside the boxes i.e. the dotted box containing $\times 2$ is a *miss* error that is counted twice because 2 speakers were missed.

hypothesis speakers K_{hyp}. We also do not necessarily know which speakers from the hypothesis should be paired with which speakers from the reference. Therefore, before evaluating, we need to find the *optimal* one-to-one assignment of each hypothesis speaker to a reference speaker. This optimal mapping should be that which minimises the total symmetric difference between the reference speakers and those mapped to them.

Consider that we have the following function to perform such a mapping:-

$$G(k \in K_A, S^A, S^B) = \begin{cases} S^B_{k^*} & \text{if optimal mapping exists} \\ \emptyset & \text{otherwise} \end{cases} \text{, where } k^* \in K_B \quad (1.27)$$

Put simply, this function takes a speaker k from S^A and returns the speaker segmentation that results from the most optimal mapping to a speaker k^* in S^B. If there is no optimal mapping for a particular speaker, such as in the case where $K^A > K^B$, then the empty set is returned.

The DER sub-components can then be calculated. First, the false alarm speaker error:

$$E^{fa}_{spkr} = \frac{\sum_k^{K_{hyp}} \xi(S^{hyp}_k \backslash S^{ref}_{spch})}{T_{spkr}} \quad (1.28)$$

Where S^{ref}_{spch} is the associated speech segmentation as derived from S^{ref} (see Eq. 1.19). This represents any time that a speaker was hypothesised and but no speech was observed in the reference.

The missed speaker error can be calculated in similar fashion:

$$E^{miss}_{spkr} = \frac{\sum_k^{K_{ref}} \xi(S^{ref}_k \backslash S^{hyp}_{spch})}{T_{spkr}} \quad (1.29)$$

This is essentially the converse of false alarm speaker error in that it represents any time when there is a speaker in the reference and no speaker in the hypothesis.

Finally, the speaker error can be calculated as follows:

$$E_{spkr} = \frac{\sum_i^{K_{hyp}} \sum_j^{K_{ref}} \xi(S^{hyp}_i \cup (S^{ref}_j \backslash G(i, S^{hyp}, S^{ref}))}{T_{spkr}} \quad (1.30)$$

This represents all the time when a hypothesised speaker is observed when its mapped reference speaker was not observed, yet another reference speaker was.

All of these errors differ from *speech segmentation* errors as they can 'stack' when presented with overlapping speech. For example, if for a given time, the system hypothesises three speakers and the reference contains only one then, assuming it did get

one right, the attributed false alarm speaker error is double to account for getting an extra two speakers wrong at that time.

For references that do not contain overlapping speakers there is no difference between E_{spch}^{miss} and E_{spkr}^{miss}. For systems which do not hypothesise overlapping speakers, there is no difference between E_{spch}^{fa} and E_{spkr}^{fa}.

These errors are referred to as *speaker* time errors by the results of the NIST tools.

1.6.3 DER Considerations

While DER has been adopted as the main evaluation metric among the diarization research community, it is worth noting that it does not offer a fully comprehensive representation of system performance for the task. In particular, it fails to describe well the following conditions.

Short Segments

DER does not represent the attribution of short segments well as scoring is based on time. Thus, the mis-classification of many short segments may not have a significant effect on overall DER.

Speaker/Cluster Purity

Often during meetings some speakers dominate the discourse more than others. In such cases a system may have highly pure clusters which model the dominant speakers, as well as several weak clusters that either poorly represent or even completely miss the remaining speakers. This behaviour may not be evident from the DER alone so we should also consider the purity of a system's output clusters to be of interest.

Number of Speakers

The DER also fails to represent how close a system was to estimating the correct number of speakers. For example, there may be a number of hypothesised clusters greater than the number of actual speakers but if some only have low temporal representation they will not influence DER substantially. Therefore, it is also important to consider the ratio/difference between the true number of speakers and a system's hypothesis (see Section 1.6.4).

There is also some contention about how overlap should be considered. Some authors (Huijbregts and Wooters, 2007) choose to report E_{spch}^{fa} and E_{spch}^{miss} errors from the SAD error and only E_{spkr} from the true DER formulation. This essentially ignores

overlap and results in a lower overall DER. Others (Miró et al., 2012) choose to report the E_{spkr}^{fa} and E_{spkr}^{miss} errors inclusive of overlap, e.g. a segment which contains two speakers that has been completely missed by the system will have twice the error. This is in fact the default formulation of the *overall speaker diarization error* for the NIST RT evaluation tools and, unless otherwise stated, will be the version presented for all results in this thesis.

1.6.4 Speaker-to-Cluster Error

As mentioned in Section 1.6.3, Diarization Error Rate (DER) does not account for the difference between a hypothesised number of speakers, K_{hyp}, and the reference number of speakers, K_{ref}. Therefore, we should also consider the following metric:-

$$E_{clst} = K_{ref} - K_{hyp} \tag{1.31}$$

This measure can also show whether a particular method has under-clustered ($E_{clst} > 0$) or over-clustered ($E_{clst} < 0$).

In some cases we may simply be interested in knowing the absolute difference:-

$$|E_{clst}| = |K_{ref} - K_{hyp}| \tag{1.32}$$

This can be useful when averaging the results across a large dataset as averaging E_{clst} alone can be deceptive if there is a mixture of over- and under-clustering.

1.6.5 Speech Recognition: Word Error Rate (WER)

The primary metric for evaluating Automatic Speech Recognition (ASR) system performance is Word Error Rate (WER) (Hunt, 1990)(Wang et al., 2003). This is measures how well a system hypothesised word sequence matches with a reference transcript. The hypothesis is optimally aligned with the reference, allowing the following formulation to be calculated:

$$WER = \frac{S+D+I}{N} \tag{1.33}$$

Where S, D, and I correspond to substitutions, deletions and insertions respectively. Substitutions represent cases where a hypothesised word is aligned with a different word in the reference. Deletions represent cases where a reference word is not aligned with any corresponding word in the hypothesis. Insertions, conversely, represent cases when a hypothesised word is not aligned with any corresponding word in the reference.

N is the total number of words in the reference. Often the metric is expressed as a percentage, although it is important to note that it is possible to exceed 100% if the number of insertions is large enough.

There are many other metrics and variants of WER that are used in the speech recognition community. Some, for example, attribute less weight to errors where the system hypothesised a similar word to that in the reference. Others may take into consideration the confidence attributed by the system to its hypothesis. However, for the purpose of this thesis we will use the standard WER as described above.

1.6.6 Machine Translation: Bilingual Evaluation Understudy (BLEU)

Machine translation is the task of converting text from one natural language to another e.g. from English to French. Such tasks are notoriously difficult to evaluate automatically as the performance is in many regards subjective. This means that human evaluations can provide a better representation of how well a system performs. However, such evaluations are expensive and time consuming.

As a result, for this thesis, we will primarily be using the BLEU score (Papineni et al., 2001). This is the most commonly used automatic evaluation metrics for machine translation. It works by taking candidate translated word sequences, usually at a sentence level, and comparing them with one or more possible reference translations. The BLEU score for a given word sequence can be formulated as follows:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{1.34}$$

Here, N is total number of potential reference translations for the given word sequence. p_n is the *modified unigram precision*. It is calculated by counting the number of words in the hypothesis that have a corresponding word in the reference, however after each match the reference candidate is 'exhausted'. This prevents high precision for cases where the hypothesis over-estimates a word more times than necessary. w_n represents a weighting for the given reference translation. This means a system can be rewarded for selecting a reference translation that, for example, is more common or had greater inter-annotator agreement. The brevity penalty, BP, accounts for hypotheses which under-estimate the number of words. This prevents cases where the hypothesis gets all of its words correct (high precision), but has missed out some words.

The brevity penalty is calculated as follows:

$$BP = \begin{cases} 1 & \text{if} c > r \\ e^{1-r/c} & \text{if} c \leq r \end{cases} \tag{1.35}$$

Where c is the length of the hypothesised translation and r is the effective reference length – the length of the best matched translation from the references.

1.7 Relationship to ASR: Sequences of Speech Units

The tasks of Automatic Speech Recognition, Speech Segmentation and Speaker Diarization are, in many ways, very similar. Ignoring for a moment the use of linguistic information such as language modelling, and considering it from a purely acoustic perspective, we can view the task of Speech Recognition as that of correctly identifying a sequence of phonemes given an audio recording containing speech. In the same way, we can think of Speech Segmentation as finding the sequence of speech and non-speech in such a recording (speech segments), and Speaker Diarization as finding the sequence of Speakers (labelled speaker segments). As such, all three tasks attempt to sub-divide recordings into sequences of different *units* of speech.

It is no surprise therefore that many tools and techniques are common across the various state-of-the-art methods for approaching all three of these tasks. However, the similarities between the tasks can make it difficult to find solutions that are appropriately distinctive to each. For example, Speech Recognition systems are typically required to be highly phoneme discriminative while being independent of speaker variation. Speaker Diarization systems, on the other hand, are required to do exactly the opposite: to be highly speaker discriminative while being independent of phoneme variation.

We therefore need to tailor each of our systems to tend towards finding their respective acoustic units. In this section we will look at the relationship between these acoustic units and how they are sub-divided in order to formalise the tasks of Speech Segmentation and Speaker Diarization.

1.7.1 Temporal Space: The Duration of Speech Units

One of the fundamental ways in which the three units – phonemes, speaker segments and speech segments – differ is in their respective temporal durations. We can assert that the general distributions of relative durations for each unit is as follows (here we

have also included a fourth unit 'word' to further illustrate the context of the speech units):-

$$\text{Phoneme} \leq [\text{Word}] \leq \text{Speaker Segment} \leq \text{Speech Segment}$$

From here we can see that in the most extreme case a speech segment can consist of a single speaker segment that in turn consists of a single phoneme word (consider the word 'a'). In general, phonemes do not typically exceed 500ms in duration, and while there are no explicit limits on the duration of any of the units they do typically exhibit behaviour that can be modelled by a probability distribution that is a result of linguistic and physical constraints.

We will see in Section 4.2 that this inherent difference in duration can be exploited to encourage, for example, longer units to be discovered over shorter units.

1.7.2 Feature Space: Speech Unit Clustering

As with ASR, the most practical way to parametrise a given recording of speech for segmentation or diarization is to divide the raw audio signal into a uniformly-spaced sequence of short-term analysis windows (frames) which are typically calculated every 10ms. These frames are then converted to the frequency domain using standard methods such as the Fast Fourier Transform (FFT). The information contained in the frequency domain is further compressed by transforming it into features borrowed from the ASR domain such as Mel-frequency Cepstral Coefficients (MFCCs), Perceptual Linear Prediction Coefficients (PLPCs) or simply filter-bank coefficients.

While features derived from ASR in this way can indeed be suitable for modelling speaker variability as well as speech/non-speech classes, some care must be taken as they are typically inherently designed to maximise the discrimination of phoneme variability. Speaker Diarization, for example, typically requires an un-supervised clustering of the acoustic space into homogeneous speaker clusters. However, there is no guarantee that such a clustering will not instead find clusters other than speakers e.g. phone classes, channels, noise classes, etc.

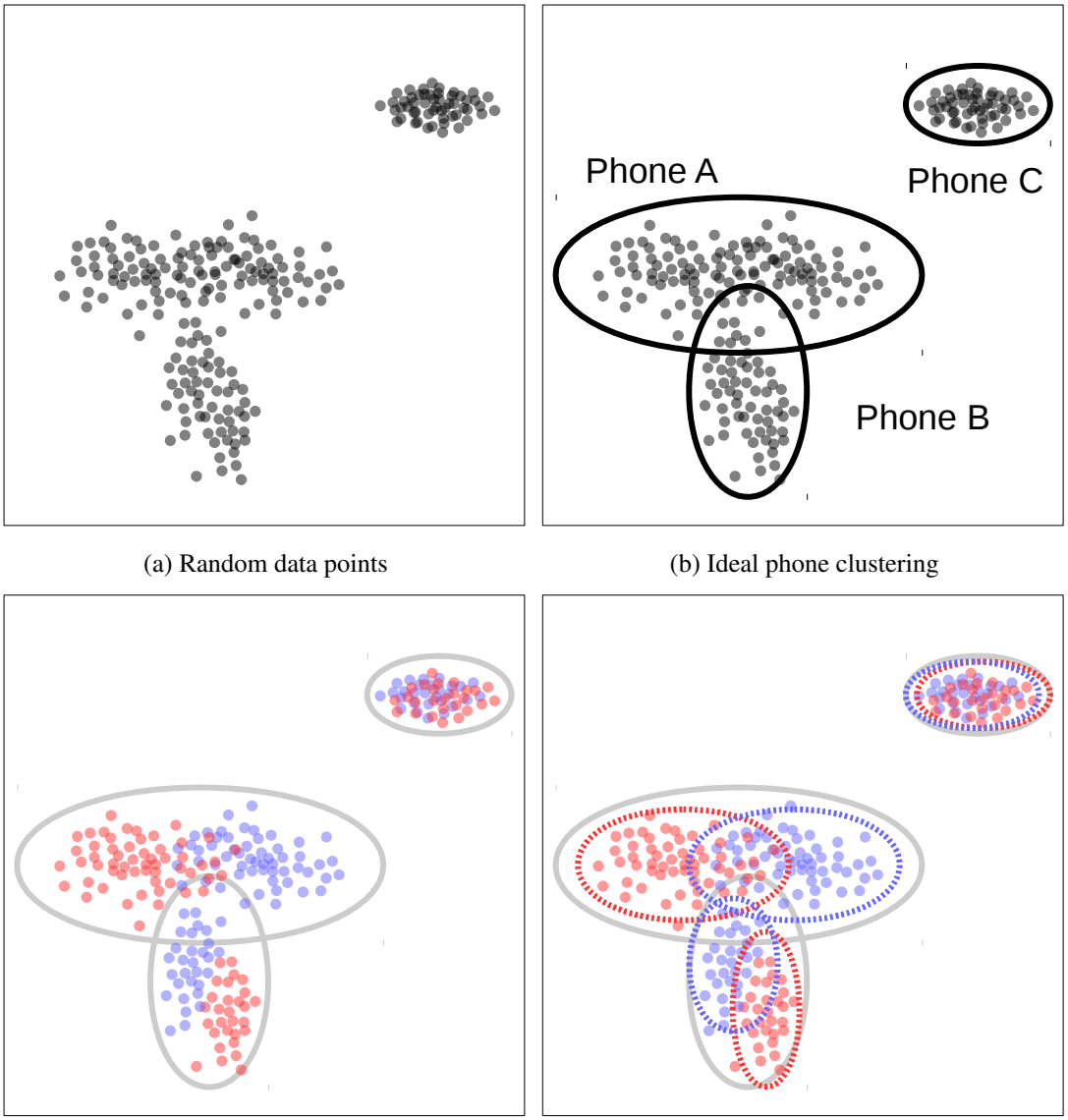
An illustrative example of this is shown in Figure 1.5. The rectangular areas in each subplot represent some hypothetical feature space that is derived from Short-term Fourier Analysis (STFT). Figure 1.5a shows some random data points that could have come from parametrising the frames of a given recording. If we perform an un-supervised clustering technique on these data points, such as K-means, we may

find clusters as shown in Figure 1.5b. If our feature space is particularly phoneme-discriminative then we may also find that these clusters correspond to actual phonemes, in this case we have found 3 different phones; one which is quite distinct (Phone C) and two which have some degree of overlap (Phones A and B). In a real world scenario this could be analogous to the differences between a single consonant and a pair of somewhat similar vowels. In this case we are able to model, with some degree of success, the phone variability. Such a clustering could be ideal for ASR or even applicable to speech segmentation, but if we are interested in clustering speakers this may not be suitable.

Consequently, consider now that these data points in fact came from frames pertaining to two different speakers and that their corresponding speaker labels are exposed such as in Figure 1.5c. Here we see that if we were in fact trying to cluster speakers, as in the case for Speaker Diarization, then the example clustering would have failed. This is related to the fact that it is often the *intra*-phoneme variability that actually constitutes the differences between speakers but this can be difficult to ascertain in an unsupervised manner. Figure 1.5d shows an example of what could be an ideal speaker clustering. We see that for some phonemes the speaker variability is low: they contain little speaker-discriminative information – such as in our hypothetical 'consonant' Phone C. For the other phones however, the speaker variability is greater and this represents an ideal sub-space for speaker discrimination.

We could potentially capture this variability if our models are complex enough – such as a Gaussian Mixture Model (GMM) with a large enough number of mixture components. However, there is still no guarantee of ideal clustering and the similarities between speaker models could still dominate over the differences. In practice, this is also made more difficult by the fact that, typically, we do not initially know the true speaker turn sequence (the points in time where the speaker changes). This means we may not have a reliable sequence to initialise with, resulting in impure clusters (containing more than one speaker) and data sparsity issues that limit our initial model complexity.

We therefore need to consider our feature space and how to make it appropriate to the task i.e. speech/non-speech discriminative for Speech Segmentation and speaker discriminative for Speaker Diarization.



(a) Random data points

(b) Ideal phone clustering

(c) Speaker labels

(d) Ideal speaker clustering

Figure 1.5: *Illustrating the problem of speaker clustering in phone-discriminative feature space. The colours, blue and red, represent 2 different speakers. The solid lines represent phone clusters and the dashed lines represent speaker clusters.*

1.7.3 Non-Speech Units, Acoustic and Channel Effects

In addition to speech, in practical applications we may also have to contend with non-speech noise such as environmental noise, music, and sound effects; channel effects such as telephone vs. wideband, microphone conditions and recording quality; and acoustic conditions such as reverberation.

All three tasks, ASR, Speech Segmentation and Speaker Diarization, should ideally be able to function independently of such signal corruption. However, in practice we find that the performance of ASR and Speaker Diarization systems can be greatly affected by such variation e.g. our speaker clustering may end up clustering channel effects or noise conditions. The performance of Speech Segmentation can also be affected, however it is worth noting that a speech segment does not have to exclusively comprise frames that contain audible speech as, for example, there can often be acoustic pauses during speech that should still be grouped together into the same segment. We will discuss this further in Chapter 5.

1.8 Datasets

In this section, we will list the datasets that will be used throughout the thesis.

NIST RT06/07/09

The NIST Rich Transcription (RT) evaluation campaign (NIS, 2006) ran annually between 2002 and 2009, focusing on promoting Metadata Extraction (MDE) for speech. For some of the years, the campaign included a dedicated speaker diarization task and the use of the associated datasets and evaluation tools have come to form the standard for developing and comparing most current systems. The NIST RT challenges have probably been the most significant driving force for community interest and support for speaker diarization.

This thesis will make use of data from the RT06, RT07, and RT09 campaigns each of which comprise 9, 8, and 7 meetings respectively. For all intents and purposes, there is no scientific reason to split the data into its RT campaign of origin as it is all of a similar interest and design. Therefore, while there are only a small number of meetings per campaign, by combining the data from all three campaigns together we can obtain results that offer greater statistical significance. For historical comparisons however, system results are often shown separately for each campaign in publications.

The NIST RT corpus is primarily being used here for system evaluation as results on this data are often presented for competing systems in speaker diarization literature

AMI Corpus

The AMI corpus (Mccowan et al., 2005) is a collection of over 100 hours of meetings data gathered under similar (often identical) conditions to the RT data. For each meeting it contains multiple audio/video sources and a rich transcription.

This is a rich data resource but results are currently not greatly represented in current diarization publications. This may change in the future as the AMI corpus offers a chance to provide a broader variety of results as well as opportunities for supervised system development and the gathering of more robust global statistics.

The AMI corpus is primarily being used here as a training and development dataset as evaluation results for speaker diarization are not well represented in literature.

IWSLT 2010/11/12/13

The International Workshop for Spoken Language Translation (IWSLT) has involved a yearly evaluation campaign focusing on the automatic transcription and subsequent translation of TED talks. TED (Technology, Entertainment and Design) is a global conference which invites experts from a variety of disciplines to give short talks. These talks, which are primarily given in English, are then shared freely online and are translated by the user community into multiple different languages. This allows for the creation of a dataset which contains spoken audio and video aligned with punctuated, multi-lingual manual transcriptions. The IWSLT workshop has built an evaluation campaign around this resource which includes challenges in ASR and MT as well as standardised scenarios and scoring metrics. This dataset typically only includes a single speaker per session and is therefore not particularly useful for speaker diarization-based experiments. As such, this dataset is used for speech segmentation experiments and to explore the effects on downstream NLP tasks such as MT and punctuation restoration.

Multi-Genre Broadcast (MGB) 2015

	NIST RT	AMI	IWSLT	MGB
Scenario	meetings	meetings	lectures	broadcast
Multiple speakers	X	X	-	X
Segmentation (manual)	(RT06 only)	-	X	(dev. only)
Segmentation (force aligned)	X	X	-	X
Transcription (manual)	X	X	X	(dev. only)
Translations (manual)	-	-	X	-

Table 1.1: Dataset information and available references

This data set is derived from 7 weeks of BBC television recordings across multiple channels and genres. The resultant corpus comprises over 1600 hours of audio with meta-data including the original subtitles.

An associated evaluation campaign (Bell et al., 2015a) was derived that included different tracks for tasks such as speech recognition, alignment and longitudinal speaker diarization (across a series of episodes from a given show). The campaign was associated with the Automatic Speech Recognition and Understanding Workshop (ASRU) 2015.

This dataset and the resultant work and publications from the workshop are a rich resource for research into speech segmentation and speaker diarization as well as other tasks. However, as the release of the dataset was near the end of the research period for this thesis, its representation is limited.

Table 1.1 provides an overview of some information and statistics on the datasets. Here we can see that some characteristics are not common across the datasets and this can make the comparison of certain conditions difficult or impossible. For example, we do not have translation references for a dataset with multiple speakers – this means we cannot examine the effects of speaker diarization on translation.

Aside from these examples, there are many other datasets that already exist, in particular deriving from the ASR field, for scenarios such as telephone conversations and broadcast news. Emerging and future application scenarios of diarization that will increasingly become of interest include web videos (Clement et al., 2011), court rooms, parliament/council, press conferences, TV/film, etc. However, the focus of this research will be on the datasets described above.

Chapter 2

Acoustic Features for Speech Segmentation and Speaker Diarization

2.1 Feature Extraction

In this section we will discuss some of the most common acoustic features used in state-of-the-art speech segmentation and speaker diarisation systems. Many of these features have been adopted from the speech recognition field, the most common of which are Mel Frequency Cepstral Coefficients (MFCC) and Perceptual Linear Prediction (PLP) (Milner, 2002). These features generally are motivated by how speech is produced and perceived by the human vocal and auditory systems respectively.

Voiced human speech is produced by the excitation of the vocal folds to create a pulse-train with a fundamental frequency (F0). The resultant signal is then filtered by the resonances of the vocal tract: the position and shape of the larynx, naval cavity, and mouth cavity (tongue, lips, teeth, etc.). This process is often described with the source-filter speech production model – the pulse train is the source and the vocal tract is the filter.

It is the filter component that creates the different units of speech (phonemes) that we use to encode information during the speech process. For non-tonal languages such as English, the source does not play a significant part in the transmission of this information, other than adding intonation (change in F0) for prosodic effect. However, the source does contain information about a given speaker – each speaker will, to some extent, have a unique source characteristic giving their voice a certain *timbre*. One of the most significant differences between speakers is their own fundamental frequency – averaging around 120Hz for adult males and 210Hz for adult females (Traunmüller and Eriksson, 1995). However, even speakers with the same fundamental frequency can be distinguished as there are other acoustic artefacts which are present with the pulse train.

2.1.1 Mel Frequency Cepstral Coefficients (MFCC)

One of the most popular feature extraction method in the speech processing community is Mel Frequency Ceptral Coefficient (MFCC) analysis. MFCC features are widely used for many tasks, such as speech recognition, because they are:-

- Functionally uncorrelated
- Easy and cheap to model e.g. with diagonal covariance GMMs
- Highly compact – can represent a second of speech with only a few hundred parameters

The process for extracting MFCC features is as follows:-

Pre-emphasis

The pulse-train produced by the source (vocal folds) has more energy in the lower frequencies than in the higher frequencies. This is known as *spectral tilt* and is a result of physiological aspects of the vocal folds. If we do not compensate for this non-uniform energy distribution then lower frequencies may appear more dominant in our analyses.

In practice, this can be achieved by applying a simple first-order filter in the time-domain:

$$x[n] = x[n] - \alpha x[n-1] \tag{2.1}$$

Where $x[n]$ is a sample at time n and α is a filter coefficient that is typically set in the range $0.95 < \alpha < 0.99$.

Windowing

It is not practical to consider the whole input signal at once as it is fundamentally non-stationary in nature. Instead, we adopt a strategy of *short-term analysis* whereby we further discretise the signal into short analysis 'windows' comprising a fixed number of samples each. These windows are typically 20-30 milliseconds in length as we can make the assumption that most natural signals of interest to our task, particularly speech, will be stationary across this sort of time. This allows us to essentially convert the whole non-stationary input signal into a piecewise-stationary sequence of 'frames'. The frames are usually multiplied by a window function, such as a Hamming or Hanning window, which helps with edge-effects during spectral analysis (see next item). After each frame is processed we shift the start time of the next frame by a *frame shift* that is an amount of time less than the length of a frame (typically 10ms). This creates a temporal overlap between frames and allows us to have a more fine-grained analysis i.e. a higher number of frames for a given amount of time.

Spectral Analysis

In order to garner the most discriminative information from the input signal, we need to move to the spectral domain. For MFCC extraction, this is done by means of a Discrete Fourier Transform (DFT) which is most commonly realised by means of the Fast Fourier Transform (FFT). The FFT is formulated as follows:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi k \frac{n}{N}} \tag{2.2}$$

Here, X_k is a given frequency bin and x_n is a sample at position n in the analysis window.

Mel-scale Filter-bank

Human hearing is subject to a non-linear perception of frequency – we are not as sensitive to differences in high frequencies as we are to those in low frequencies. As a likely consequence of evolution, there is a resultant correlation with spectral distribution of information in speech. Speech information is more densely packed into the lower audible frequencies. It is pertinent, therefore, to apply a non-linear strategy to our spectral analysis. We can do this by warping the linear frequency spectrum to a resultant space that better matches the response of the human auditory system. The most common example of a suitable warping is the *Mel-scale* function:

$$M(f) = 1127 \ln(1 + f/700) \tag{2.3}$$

Where f is the linear-spaced frequency. As this is a heuristically derived approximation, there are many variants and alternatives, such as the Bark function, that attempt to improve upon it. However, in practice the choice does not have a significant impact on down-stream performance.

Normally for MFCC computation such Mel-scaling is not applied directly. Instead is realised through the application of a *Mel-scaled* filter-bank to the output of our initial spectral analysis. The filter-bank comprises a series of triangular filters and, when ordered from low to high frequency, each filter's pass-band begins at the mid-point of the previous filter. The filters are typically linearly spaced below 1000Hz and logarithmically above 1000Hz. This results in coarser spectral representation for high frequencies and finer representation for low frequencies, emulating the response of the human auditory system.

The configuration of the filter-bank means the resultant output serves jointly to compress the information from the spectral domain as well as mapping it onto the Mel-scale.

Log Power Spectrum

We compute the log magnitude squared of the energy from each filter in the Mel-scaled filter-bank. This compresses the dynamic range and also mimics the

logarithmic sensitivity of the human auditory system to sound pressure levels (energy) – we are more sensitive to differences at low energies than high energies.

Discrete Cosine Transform (DCT)

The next step involves converting to the *cepstral* domain. This can be considered as the 'spectrum of the log power spectrum' and has the effect of approximately deconvolving the source and filter components of the speech production model.

We produce the cepstrum by applying a Discrete Cosine Transform (DCT) to the log power spectrum.

$$y_t[k] = \sum_{m=1}^{M} \log(|Y_t(m)|) \cos(k(m-0.5)\frac{\pi}{M}) \tag{2.4}$$

By applying a so-called lifter (filter in the cepstral domain), we can find the low and high-frequency components of the cepstrum, which roughly correspond to the filter and source speech components respectively. The output from such a lifter provides our cepstral coefficients.

The log energy outputs from the filters are highly correlated which can be difficult to model efficiently. An additional benefit of the cepstral coefficients is that they are functionally de-correlated. This means they can be easily modelled by Gaussian Mixture Models (GMMs) that have only diagonal-covariance matrices, greatly reducing the number of parameters needed to model the information.

2.1.2 Perceptual Linear Prediction (PLP)

An alternative to MFFC features is Perceptual Linear Prediction (PLP) parametrization (Hermansky, 1990). This method takes a similar motivation as to that of MFCCs – to emulate the human auditory process – however the realisation is achieved in a different way. PLPs have been shown to be more noise robust than their MFCC counterparts.

The windowing and the spectral analysis are the same as for MFCCs, after which the process differs:-

Critical-band Integration

Similar to the function of the mel-filter bank for MFCCs we want to map the spectral analysis onto a space that better represents the non-linear perception of human hearing to pitch. This is done by first warping the power spectrum onto

the bark scale by use of an approximation function. This is then convolved with the critical-band masking curve (an empirically-derived mapping of the sensitivity of human hearing to pitch). The curve is usually approximated through the use of a piece-wise function. As the mel-filter bank method can be considered an *over*-approximation of this curve, critical-band integration allows for a more faithful representation.

Equal Loudness Pre-emphasis

Just as per the MFCC extraction process, we need to compensate for the spectral-tilt of the source. We apply an approximation function of the equal-loudness curve (an empirically-derived mapping of the sensitivity of human hearing to intensity) to the output of the critical-band integration. This is closer to the actual human auditory system than the simple time-domain filter used for MFCCs.

Intensity to Loudness Compression

In order to compress the dynamic range, we apply cube-root to the resultant energy from the equal loudness pre-emphasis. This is a better approximation than the log energy used for MFCC extraction.

Discrete Cosine Transform (DCT)

This step is the same as for MFCCs and has the effect of moving into the cepstral domain.

Linear Predictive Coding (LPC)

In order to produce the final PLP coefficients we we apply Linear Predictive Coding (LPC) to the cepstral domain.

2.1.3 Dynamic Features

Many speech processing systems will use static MFCC or PLP features but will also append derivatives of them to the overall feature vector for a given frame (Furui, 1986; Hanson and Applebaum, 1990). These derivatives normally constitute the velocity and acceleration (delta and delta-delta) of the static features over some window of context. This adds dynamic information that can be useful for discerning certain phone units.

For example, some phones are realised through a filter that changes over time. This results in dynamic formant behaviour in the spectrum. In order to capture this we need to know not only the characteristics of a stationary snapshot of the filter but how it is changing over time.

This extra information can be useful for identifying speech-like dynamic behaviour in a recording, which can contribute to speech segmentation systems. However, it does not inherently contain much speaker discriminative behaviour, so is often neglected from speaker diarization systems as it can either have a negligible effect or even hamper performance by introducing statistical noise.

2.2 Speaker Diarization Experiments

In order to investigate the effect of different feature extraction techniques on speaker diarization, we performed some experiments on the NIST RT06/07/09 datasets combined. To control for the influence of speech segmentation, we used the reference speech segmentation. Therefore, SAD error is zero in all cases so we do not report it.

The speaker diarization system we used is described in Section 3.3.3 and is allowed to both cluster and re-segment speaker segments. The analysis window in all cases was 30ms with a step-size of 10ms.

2.2.1 MFCC Dimension

As we described in Section 2.1.1, MFCC parametrization can approximate the source-filter model of speech production whereby the higher coefficients represent the source and the lower coefficients represent the filter. In general, the filter component of this model is predominantly phone-dependent, whereas the source component is predominantly speaker-dependent.

For speech recognition, we do not need to model the source component in detail as we are only interested in discriminating between phonemes, therefore it becomes increasingly redundant to include higher coefficients. It is widely accepted that for ASR tasks a feature vector of dimensionality 13 (12 MFCC coefficients + energy), is sufficient to describe enough information about the filter component of speech. In practice dynamic feature derivatives are also augmented to the static vector for each frame, making the final feature dimensionality up to 39 (13 static, plus delta and delta-delta), as this can help to improve phone discrimination further. However, adding coefficients to the initial static vector does not add any more useful information and can actually degrade ASR performance as it can effectively add ‘noise’ to the phone-discriminative information in the lower coefficients.

For speaker diarization, the problem is essentially inverted – we are more interested

in the variability in the source component for each speaker than the inter-phone variability in the filter component. For this reason, we would like to be able to describe the source component in more detail than for speech recognition. To do this we can simply increase the number of coefficients in our MFCC parametrization. Many speaker diarization systems use 19 MFCC coefficients. Energy is typically not appended to this vector as it is not inherently speaker discriminative. We also do not use any dynamic features as this does not typically add much speaker-discriminative information for the task and also requires more complex models. During clustering, particularly in the initial iterations of agglomerative clustering-based speaker diarization (see Section 3.2.1), there may not be a substantial amount of data available to estimate models for clusters. This means we also need to take care not to use too many feature dimensions as it can cause model instability and affect the trajectory of the cluster merges.

Table 2.1 shows results obtained from using different MFCC feature dimensions for our baseline speaker diarization system (see Section 3.3.3 for system details) on NIST RT data. In all cases we used the reference speech segmentation. We would expect the E^{fa}_{spckr} error to be zero in this case, however a small rounding error is introduced because the reference has a fidelity of 1ms and our system operates with 10ms frames. We consider this to have a negligible effect on the results.

The first row shows results for 12 MFCC coefficients, typical in ASR applications, and the second row shows results for 19 MFCC coefficients. Here, we can see that there is a substantial gain in performance with the DER reducing from 25.41% to 21.93% and the absolute cluster difference reducing from 1.71 to 1.13. Interestingly, we also see that the relative cluster difference changes from a tendency to under cluster (+0.46) towards a tendency to over-cluster (-0.71), so the cluster termination points are also affected. This performance gain can likely be attributed to the increase in information that the extra coefficients provide about the source component.

The remaining rows in Table 2.1 show what happens when we begin to drop the first N coefficients from the MFCC vector. We ran the system for $N = 1$ to $N = 4$ i.e. dropping from the first, up to the first four coefficients. We find that the DER actually improves slightly for $N = 1$ to $N = 3$, particularly for the case when we drop the first coefficient (21.92% to 21.10%). When we reach $N = 4$, the performance suddenly degrades. This may represent the point where MFCC coefficients move from describing mostly filter components to describing a mixture of source and filter components and should therefore be kept for the speaker diarization task.

We also note that while some values of N improve DER, the absolute cluster differ-

ence degrades slightly. Often, this can actually result in a better clustering for downstream tasks such as ASR. However, as the changes are only slight, we can make a broader assumption that the first 3 MFCC coefficients have a negligible effect on speaker diarization performance for either better or worse. For this reason, we will use the full 19 static MFCCs for all speaker diarization results, unless stated otherwise, as this corresponds with the features used in the majority of systems found in the speaker diarization literature.

Table 2.1: MFCC feature dimension results on NIST RT06/07/09 data.

	$E_{spkr}^{miss}(\%)$	$E_{spkr}^{fa}(\%)$	$E_{spkr}(\%)$	DER(%)	E_{clst}	$\lvert E_{clst}\rvert$
MFCC12	5.02	0.06	20.33	25.41	0.46	1.71
MFCC19	5.03	0.03	16.86	21.93	-0.71	1.13
MFCC19_drop1	5.03	0.03	16.05	21.10	-0.54	1.29
MFCC19_drop2	5.03	0.05	16.56	21.63	0.13	1.38
MFCC19_drop3	5.03	0.06	16.77	21.84	0.71	1.54
MFCC19_drop4	5.03	0.07	20.10	25.19	1.54	2.29

2.2.2 MFCC vs. PLP

We also compared the use of MFCC features against PLP features for the speaker diarization task. Table 2.2 shows the results for a dimensionality of 12 and 19 for both MFCC and PLP parametrizations of the data. All other aspects of the system remained constant.

We find that, as in Section 2.2.1, 19 coefficients significantly out-performs 12 for both techniques. This is likely due to the increased description of the source component in both cases.

The like-for-like performance between MFCCs and PLPs seems to be broadly similar. PLPs slightly out perform MFCCs for 12 coefficients (25.41% DER vs. 24.85%), but MFCCs show a gain at 19 coefficients (21.93% DER vs. 22.22%). We can confirm that 19 MFCC coefficients is a good choice for speaker diarization as it shows the best performance overall.

Table 2.2: Comparing MFCC and PLP features on NIST RT06/07/09 data.

	$E_{spkr}^{miss}(\%)$	$E_{spkr}^{fa}(\%)$	$E_{spkr}(\%)$	DER(%)	E_{clst}	$\lvert E_{clst}\rvert$
MFCC12	5.02	0.06	20.33	25.41	0.46	1.71
MFCC19	5.03	0.03	16.86	21.93	-0.71	1.13
PLP12	5.03	0.05	19.78	24.85	-0.13	1.38
PLP19	5.03	0.02	17.17	22.22	-0.88	1.13

Chapter 3

State-of-the-art Speech Segmentation and Speaker Diarization Systems

In this chapter we will look at some of the state-of-the-art methods for speech segmentation and speaker diarization. We will present some of the standard methods that are used for each task as well as providing some examples of real systems that have been presented by the scientific community.

3.1 Speech Segmentation Methods

Speech segmentation methods can be broadly classed into two different categories *Energy-based* methods and *Model-based* methods, into which the majority of systems can be placed. There are other system types, such as those which incorporate multi-modal features from video/facial analysis, however for the purposes of this thesis we will focus only on the two aforementioned categories.

3.1.1 Energy-based Methods

We will define an Energy-based speech segmentation method as any method which takes a short-term observation window of the audio data, measures the spectral energy within that window and makes a decision based on defined thresholds as to whether or not speech or non-speech is present. Many of these methods make the assumption that regions of speech in a recording are likely to have a higher absolute energy than regions of non-speech. This can work for scenarios such as telephone conversations due to the close proximity of the speaker to the microphone. However, for distant microphone scenarios, such as meetings, the variation of the distance between speaker and the microphone can make setting a threshold difficult. When significant environmental noise is introduced then this method breaks down even more. For example, if a meeting is taking place in a room with a loud air-conditioner then such a method may not be able to distinguish based on energy alone.

Some methods attempt to mitigate for such scenarios by using noise-reduction techniques or using adaptive thresholding. They may also look at specific spectral bands that are more likely to contain speech content.

Energy-based methods also tend to over-segment as they make decisions with little or no context. This can be controlled by including temporal factors such as a delay factor between the detection and attribution of non-speech after speech. However, this introduces more parameters to be tuned and indeed if such parameters are learned then it could arguably be considered a model-based method.

While Energy-based models lack robustness, they are often very simple and computationally inexpensive. They also lend themselves well to applications requiring a low-latency response as they do not need to consider much temporal history before making a decision.

3.1.2 Model-based Methods

Model-based speech segmentation methods use machine learning techniques to find regions of speech within a recording. Typically, models will be trained for a variety of acoustic classes using a training corpus of labelled examples. The most simple methods will use two models: one for speech and one for non-speech. However, more complex systems may include multiple non-speech classes – e.g. music, noise, silence, etc. They may also sub-divide the speech class to cover inter-speaker variability by including models for different genders, languages, speaker ages, etc.

Current state-of-the-art methods normally make use of Gaussian Mixture Models (GMMs) to model the observation likelihoods of a Hidden Markov Model (HMM) and perform a Viterbi decode to find the optimal state sequence. Much of the variation between such methods is found in the choice of features, model complexity and HMM network topology, however, the underlying concept is often very similar.

More recently, we are starting to see neural network methods achieving prominence for the task of speech segmentation. We will consider these to be part of Model-based Methods as they have many fundamental similarities.

3.1.3 Example Systems

3.1.3.1 SHOUT

The SHOUT Toolkit (v0.3)[1](Huijbregts, 2008) is a widely-used off-the-shelf speech segmentation system. The tool uses a GMM-HMM-based Viterbi decoder, with an iterative sequence of parameter re-estimation and re-segmenting.

Fundamentally, it is a GMM-HMM-based Viterbi decoder but also includes an iterative sequence of parameter re-estimation and re-segmenting. The process is as follows:-

1. The audio is first segmented using *speech* and *non-speech* 'bootstrap' models.

[1]http://shout-toolkit.sourceforge.net/download.html

2. The *non-speech* frames are split to form 2 new models:-
    - *silence* - frames with low mean energy
    - *sound* - denoting audible non-speech. Frames with high mean energy *and* low zero-crossing rate.
3. The *silence* and *sound* models only are re-estimated a fixed number of times on high-confidence frames.The number of Gaussian components is increased at each iteration.
4. A new speech model is trained on all 'speech' segments.
5. All 3 models are re-estimated a fixed number of times on high-confidence frames. The number of Gaussians is increased at each iteration.
6. *silence* and *speech* models only are re-estimated until the *speech* and *sound* models diverge according to the Bayesian Information Criterion (BIC).

Minimum speech and silence duration constraints are enforced by the number of emitting states in the respective HMMs.

3.1.3.2 QIO-Aurora Toolkit

The QIO-Aurora Toolkit (Adami et al., 2002) comprises various front-end tools for speech processing and includes a Voice Activity Detection (VAD) component. This method uses a Multi-layer Linear Perceptron (MLP) trained using a back-propagation algorithm to discriminate between speech and non-speech frames. It uses a 9-frame window, each of which contains 6 cepstral coefficients of low-pass filtered log-energies of 23 Mel filters. A forward-pass over the data is used to determine the most likely speech/non-speech frame sequence.

3.2 Speaker Diarization Systems

Speaker Diarization systems can be broadly classed into two different categories: *Top-down* methods and *Bottom-up/Agglomerative* methods. The distinction between these categories is attributed to the way they perform clustering. Top-down approaches begin by including all speech data into one cluster and iteratively splitting the cluster. Bottom-up, often referred to as agglomerative, methods begin with a large number of

clusters created by an initialisation step and iteratively merge clusters based on some similarity measure. Both methods must terminate their respective clustering schemes when an optimal number of clusters is reached.

Similar to segmentation methods, clusters will typically be modelled using GMMs. As discussed in Section 1.5, speaker diarization systems are jointly tasked with both clustering and segmenting speakers. A GMM-HMM approach is normally used to re-segment the data between iterations.

In Evans et al. (2012), some typical bottom-up and top-down speaker diarization techniques were compared and showed that while overall performance is broadly similar:-

- bottom-up clustering methods tend to produce speaker models with higher purity, and
- top-down clustering methods tend to be more robust to nuisance variability.

The study also presented a method for system combination that improved overall performance – showing that top-down and bottom-up methods can be intrinsically complementary.

3.2.1 Bottom-up methods

As Bottom-up, or Agglomerative, methods are the most common approaches to speaker diarization, we chose to use them as the foundation for our own work. In this section we present a general overview of the typical architecture and also present some state-of-the-art system examples. The majority of bottom-up methods follow some form of the following process:-

1. Initialise K clusters
2. Train GMMs for each cluster
3. Segment data using GMM-HMM Viterbi decode
4. Re-train clusters with new segmentation
5. If still doing initial pre-merge iterations, go to Step 3
6. Merge closest pair of clusters according to similarity measure
7. Stop if termination criteria is met, else go to Step 3

The main differences between competing methods tend to be found in the realisation of the **Initialisation Step**, **Similarity Measure** and **Termination Criteria**.

3.2.1.1 Initialisation Step

The bottom-up clustering approach needs to start somewhere, so we need a way to create K initial clusters in order to begin the process. There are several possible ways to do this initialisation step. In any case, K should ideally always be greater than the number of true speakers, so it is good practice to set it much greater than the likely number of speakers for a given scenario if possible.

An intuitive method may be to make use of an unsupervised clustering technique such as k-means, where we take all frames of data as data-points and attempt to partition the resultant space. The number of clusters, K, can be explicitly defined or it can be based on some kind of criterion, such as when the cluster assignments no longer change in a k-means update. However, such methods have been shown to be unreliable as they do not take in to account the temporal relationship between frames i.e. adjacent frames are more likely to belong to the same speaker. It may also result in clustering relating to other speech units besides speakers (see Section 1.7).

Many systems instead use a form of initialisation that splits all the speech data into short chunks and assigns them to a pre-determined number of clusters. In its simplest form this may involve uniformly segmenting the speech data into chunks of a fixed length, such as a few seconds. Each chunk is then assigned to one of K clusters by means of a round-robin approach. There are two main assumptions here which allow this to work:-

- The individual chunks are short enough that they are likely to be homogeneous i.e. mostly contain data from a single speaker
- It is unlikely that the speakers will have a uniform distribution – after the allocation of the chunks to the initial clusters, some clusters will be more pure than others.

3.2.1.2 Similarity Measure

During the merge step of each clustering iteration, we need to know which pair of clusters are most similar. Similarity measures, or distance metrics, allow us to create a matrix of all possible candidate pairs of clusters and their respective scores. The pair

with the greatest similarity, or lowest distance, can then be selected as the best pair to merge. There are many potential distance metrics but we will list some of the more prominent examples used for Speaker Diarization.

Generalised Likelihood Ratio (GLR)

The Generalised Likelihood Ratio (GLR) test (Solomonoff et al., 1998), is formulated as follows:-

$$d_{GLR}(X_0, X_1) = \frac{l(X_0|M(X_0))l(X_1|M(X_1))}{l(X_{0,1}|M(X_{0,1}))} \tag{3.1}$$

Here, X_0 and X_1 are the data points (frames) attributed to cluster 0 and 1 respectively. $X_{0,1}$ is the concatenation of all data points attributed to both clusters. $M(X)$ represents a model trained on data X and $l(X|M)$ is the likelihood of data X, given model M. If the data points from both clusters have different distribution then the combined model (denominator) will likely give a low likelihood score as it tries to model more spread out data. As a result the GLR score would be high, suggesting the models should not be merged as they are more *distant*. Conversely, if the data points are close together, then the GLR score would be low, suggesting good candidates for merging.

Cross Entropy

The Cross Entropy metric is formulated as follows:-

$$d_{CE}(X_0, X_1) = \log \frac{l(X_0|M(X_0))}{l(X_0|M(X_1))} + \log \frac{l(X_1|M(X_1))}{l(X_1|M(X_0))} \tag{3.2}$$

Here, if data X_0 and X_1 are similar then their corresponding models should provide more similar likelihoods for one another. That is, that the likelihood of data from one cluster, given the model from the other, should still be high. In this case, the Cross Entropy score would be greater. Conversely, if the opposite models and data points are mismatched then the terms on the right-hand side will be lower resulting in a lower Cross Entropy.

Bayesian Information Criterion (BIC)

The Bayesian Information Criterion (BIC), for the purposes of Speaker Diarization, measures the difference between the log-likelihood of a combined model and data with the sum of the log-likelihood of each separate model and data. It is defined as follows:-

$$BIC(X_0, X_1) = \log l(X_{0,1}|M(X_{0,1})) - (\log l(X_0|M(X_0)) + \log l(X_1|M(X_1))) - P \tag{3.3}$$

In order to account for the fact that the three models under analysis could potentially have a different number of parameters, e.g. GMMs with a different number of mixtures, we introduce a penalty term P. This penalty term should be some function of the number of free parameters in the models. This can prevent the likelihood for any data given a model from being greater simply because the model has greater complexity. However, this penalty factor typically needs to be tuned heuristically by means of a hyper-parameter.

It is possible to eliminate the need for such a penalty factor if the number of parameters in the combined model is set to be the sum of the number of parameters in each of the candidate pair models. Such a method was introduced by Ajmera and Wooters (2003) and by removing the need for a hyper-parameter, it was shown that the BIC method could be robust across different application scenarios.

3.2.1.3 Termination Criteria

Once we have a similarity measure, we are able to find the best candidate cluster pair to merge and are able to iterate around the sequence described in Section 3.2.1. However, we also need some kind of termination criterion so that we know when to stop clustering.

The most simple termination criterion is simply to stop clustering when we reach some desired number of clusters. This could be quite appropriate if we have a known number of speakers. Although, as we will discover in Section 4.3.2, this may not necessarily result in the most optimal speaker segmentation. However, for most canonical applications of speaker diarization, the number of speakers is unknown.

The most common solution is to threshold the similarity metrics. For example, if we are using the BIC metric, we could simply stop when the scores for all candidate cluster pairs are less than zero. This does not necessarily guarantee an optimal solution, so the choice of threshold can cause problems for performance and robustness. Note that, although they are usually the same, the choice of metric used to make a termination decision does not necessarily need to be the same as that used for the clustering decision.

Some methods (Nguyen et al., 2008) continue the clustering sequence until only one cluster remains. They then assess the segmentation that resulted from each iteration and select the most optimal based on some criteria. This can help to find a globally

optimal solution but comes at the expense of potentially greater computational cost.

3.2.2 Example Systems

3.2.2.1 ICSI RT09

A good example of a state-of-the-art speaker diarization system is the International Computer Science Institute (ICSI) submission for for the NIST RT09 competition (Friedland et al., 2011). This system is capable of using multiple distant microphones as well as visual data streams, however we will concentrate on the single distant microphone variant for the purposes of comparison with the work in this thesis.

The ICSI RT09 system is primarily an agglomerative 'bottom-up' clustering method. However, it also included some novel methods such as a pre-clustering algorithm to better initialise the agglomerative clustering process and the use of prosodic long-term features to augment the MFCC vector.

The long-term prosodic features comprise a vector of 12 pitch and formant based features that were heuristically selected for their speaker discrimination. These features are calculated over larger windows than would be used for conventional parametrisations such as MFCCs (1000ms vs. 10ms).

The prosodic features are then used to perform an initial clustering. This is used to estimate the number of initial clusters k for the agglomerative clustering algorithm as well as the number of mixture components in each cluster's GMM, g. A GMM is used to model the data, then segments are assigned to the mixture component they best match. A 10-fold cross validation of the data is performed where the different clusters are compared with each other. If there is an increase in log-likelihood then it suggests there is a lack of intra-mixture homogeneity somewhere. In this case another component is added to the GMM and the process repeats until no increase in log-likelihood is observed. At this point the number of initial clusters K can be determined as the number of mixture components in the GMM.

The clusters will then be modelled with a GMM *each* such that the standard agglomerative clustering process can begin. However, at this point we still have to decide the number of mixture components for these GMMs, g. The solution proposed is to derive it based on a formula termed Adaptive Seconds Per Gaussian (ASPG), which is defined as follows:

$$s_{mix} = 0.01 \cdot s_{spch} + 2.6 \quad (3.4)$$

$$g = \frac{s_{spch}}{s_{mix} \cdot k} \quad (3.5)$$

Where s_{mix} is the number of seconds of data per Gaussian mixture component and s_{spch} is the number of seconds of speech data.

The remaining part of the process is just a standard agglomerative clustering process with a BIC similarity measure and 19 MFCC coefficients as outlined in Section 3.2.1.

For the RT09 dataset, under the single distant microphone (SDM) condition, the ICSI RT09 system reports a DER score of 31.3%.

3.2.2.2 IDIAP

The system proposed by the IDIAP research institute, Martigny, still follows the same general formula for an agglomerative speaker diarization system (Vijayasenan and Valente, 2012; Vijayasenan et al., 2007, 2009). However, it does differ from conventional GMM-HMM systems by offering a novel clustering technique based on the Information Bottleneck (IB) principal (Tishby et al., 2000).

The method begins by considering that the problem requires a set of segments $X = \{x_1, \ldots, x_T\}$ be mapped into a set a of clusters $C = \{c_1, \ldots, c_K\}$. This is difficult to do directly, so a set of relevance variables, Y, are constructed that contain relevant information about the problem. Each segment is then mapped onto Y, giving $p(Y|X)$. In principal, Y could be any relevant variables, but in practice they are the components of a Universal Background Model (UBM) estimated on the data from the whole session recording.

The IB principal then allows us to consider that the optimal clustering should be able to compress the input variables while still preserving as much mutual information as possible about the relevance variables. This can be formulated as the minimization of the following objective function:-

$$\mathcal{F} = I(X,C) - \beta I(C,Y) \quad (3.6)$$

Where β is a Lagrange multiplier used to control the amount of information preserved, $I(C,Y)$, against the amount of compression of $I(X,C)$.

The algorithm begins with the trivial clustering of each segment into its own cluster. At each iteration of the agglomerative clustering stage, the cluster pair that minimises

the loss of mutual information is merged and the data is re-segmented. The process repeats until a threshold based on the Normalized Mutual Information (NMI) is met. One of the main advantages of this method over the conventional GMM-HMM systems is that it has a lower computational complexity and, as a result, can execute faster.

In Vijayasenan and Valente (2012), the IDIAP system is reported to have the results on NIST RT data as shown in Table 3.1, for the case that does not use Time Delay of Arrival (TDOA) features derived from a microphone array, however it is not clear if it is strictly using data from a single distant microphone.

Table 3.1: IDIAP Information Bottleneck (IB) speaker diarization system results on NIST RT data.

Dataset	$E_{spkr}^{miss} + E_{spkr}^{fa}$	E_{spkr}	DER (%)
RT06	6.60	15.60	22.25
RT07	3.70	11.30	15.03
RT09	12.70	21.30	33.98

3.3 Baseline Systems

As many of the existing systems are either unavailable or difficult to modify, we designed our own proprietary systems to establish baselines and foundations to build our experiments on. Both systems are compatible with feature files from either the Kaldi toolkit or HTK.

3.3.1 Viterbi Decoder

As Viterbi decoding is common to both systems we designed a decoder that could be used universally. It is written in Cython, which makes it possible to write optimised C code with Python-like syntax and easily integrate the resultant methods within other Python code. The advantage is that significant speed and memory optimisations can be made.

Rather than creating a generic Viterbi decoder we designed a dedicated decoder for the HMM topology shown in Figure 3.1 which allowed for some further optimisations.

Later, for the speech segmentation system only, we started using a custom OpenFST (Allauzen et al., 2007) finite state transducer (FST) as this allowed us to use the

Kaldi decoder and better integrate with the Kaldi toolkit in general (Povey et al., 2011). However, this version and our initial python version are functionally identical.

3.3.2 Speech Segmentation System

The initial baseline speech segmentation system was a GMM-HMM based system written in Python. We use the scikit-learn package to handle machine learning aspects. The system allows for iterations of MAP adaptation and re-segmentation.

Unless otherwise stated, the following parameters/options are used:-

- Kaldi feature extraction
- 39-dimension feature vector (12 PLPs + energy, with deltas and delta-deltas)
- 30ms window
- 10ms step-size
- cepstral mean normalisation.
- 100ms minimum segment duration (speech and non-speech)
- 1 iteration of MAP adaptation

The HMM topology is shown in Figure 3.1. Each of the two classed (speech and non-speech) is represented by an initial state, a number of feed-forward only states and a final state that allows both for self-transition and for a transition to the initial state of the other class. All the within-class states are assigned the same GMM for computing acoustic likelihoods.

The purpose of the feed-forward states is to enforce a minimum duration. This is necessary to prevent the state sequence from fluctuating too frequently between classes leading to 'choppy' segmentation. For the same reason we also give greater weight to the final state self-transitions than the class-transitions (0.9 vs. 0.1). The minimum duration constraint and the transition weights can be set independently for each class but are often identical.

3.3.3 Speaker Diarization System

The baseline speaker diarization system is also a GMM-HMM agglomerative system (see 3.2.1) written in Python with the scikit-learn package.

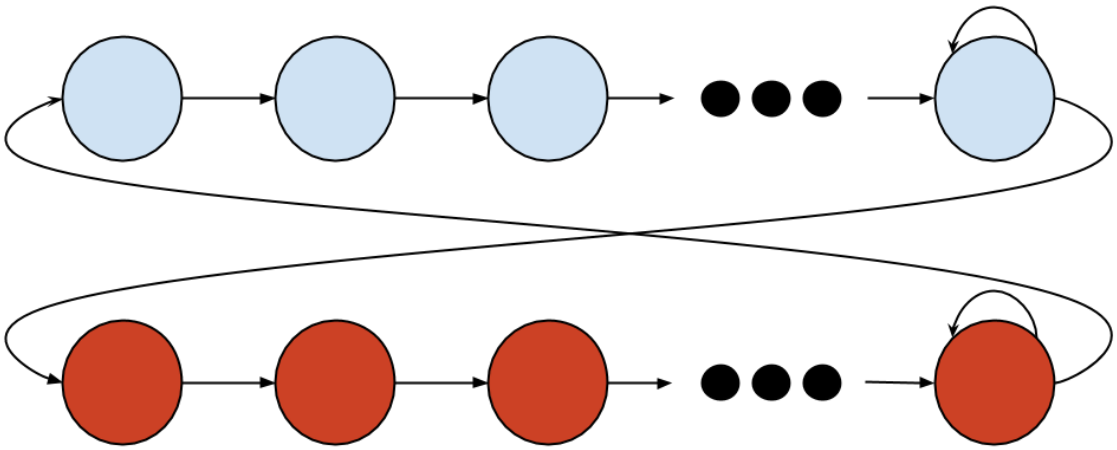


Figure 3.1: The standard HMM topology for our baseline speech segmenter. The blue states represent the non-speech class and the red states represent the speech class. The feed-forward only states enforce a minimum duration for each class. The final state for each class can self-transition or transition to the initial state of another class – these transitions can be weighted.

Clusters are initialised by a uniform chunking of the data into 2.5s second segments. Each segment is then assigned to a cluster using a 'round-robin' strategy. We perform an initial round of re-train/re-segment iterations before merging is allowed. Merging iterations then begin whereby candidates are selected with the BIC criterion. Merging stops when all cluster pairs have a BIC score less than zero.

Unless otherwise stated, the following parameters/options are used:-

- HTK/Kaldi feature extraction
- 19-dimension feature vector (19 MFCCS, no energy or dynamics)
- 30ms window
- 10ms step-size
- no cepstral normalisation
- 250ms minimum segment duration
- 4 initial pre-merge iterations

- BIC merging criterion
- 5 initial mixture components per GMM
- 16 initial clusters

The HMM topology is similar in concept to that of the speech segmentation baseline system (see Figure 3.1), except that we have a class for each speaker cluster instead of just speech/non-speech and the minimum duration is typically longer.

Initially we used the HTK toolkit (Young et al., 2006) for feature extraction and later switched to the Kaldi toolkit for this purpose. The features produced by equivalent parametrization set-ups are not strictly identical but have a negligible effect on system performance. This choice was purely to facilitate easier integration with wider end-to-end systems.

Chapter 4

An Oracle Investigation into the Shortcomings of Speech Segmentation and Speaker Diarization Systems

In this chapter we present our initial work into finding the challenges faced in speech segmentation and diarization, as well as the shortfalls of current-state-of-the art systems. This work served to motivate and inform the research directions that we could take in order to best tackle the problems.

4.1 Motivation

The more recent NIST RT campaigns (RT05/06/07/09) focused on diarization of meetings data and several systems were entered into the challenge by different institutions. However, across all entries the system performance on this task was found to be notoriously meeting-dependent and hyper-sensitive to system parameters (Mirghafori and Wooters, 2006).

Diarization systems based on agglomerative clustering generally involve an initialisation step, followed by interleaved iterations of re-segmenting the speech, re-estimating the speaker models, and merging models, to gradually converge on the correct number of speakers and the best segmentation and speaker assignments. This architecture means that the final system performance is a complex function of the performance of the individual parts, making it very difficult to identify the causes of error. As a consequence, it was not clear where to begin improving the shortfalls of such systems. The work we present here was motivated by the need for a better understanding of the system component factors that contribute to diarization error. Our ultimate goal was to identify where improvements are needed and, conversely, which parts of the system already work well.

An investigation along similar lines was conducted in (Huijbregts and Wooters, 2007). Our investigation is complementary to that work. We investigate several aspects of the system performance in more detail – e.g. how much pure data is desirable to initialise a speaker model; how significant the number of overlapping speakers is. We also expand on the conclusions about where efforts should be focussed in order to reduce diarization error rate. Our methodology is broadly similar to theirs – we start with a diarization system that is capable of good performance in the standard fully-unsupervised mode, and then conduct various ‘oracle’ experiments to isolate the effects of various components.

4.2 System Description

There are several diarization systems with competitive state-of-the-art performance such as ICSI (Friedland et al., 2011), IDIAP (Vijayasenan et al., 2011), LIAEURECOM (Bozonnet et al., 2010) and I^2R (Sun et al., 2010). However, we have developed our own speaker diarization system. This system is written in Python (and Cython for the Viterbi decoder) and is designed to be more modular and flexible than existing available systems.

By choosing similar parameters and methods it is possible to closely emulate the performance of the ICSI system e.g. for single distant microphone (sdm) RT09 data, ICSI has an average of 31.3%DER (Friedland et al., 2011) vs. our 31.8%. The slight difference in performance may be due to the fact that the ICSI results include pre-processing the audio with dynamic range compression followed by Weiner noise filtering.

Unlike many other systems (e.g., (Anguera et al., 2006b)) we choose not to use a beam-formed signal from multiple channels of a microphone array and instead opt for single distant microphone data. A beam-formed signal typically improves DER results for systems that ignore overlap (Anguera et al., 2005), but could be a poor choice if we wish to detect a number of simultaneous speakers. This is because the nature of beam-forming could result in focusing on the most dominant speaker during overlap, effectively *filtering* out the weaker speaker(s).

Speech Activity Detection (SAD) is performed by the QIO-Aurora tool-kit (Adami et al., 2002) as this has proven to work well with the RT datasets (Zwyssig, 2013). For the regions labelled as speech, feature extraction is performed using HTK (Young et al., 2006). We use the first 19 MFCCs computed from a bank of 26 Mel-scaled triangular filters with a pre-emphasis coefficient of 0.97 and cepstral lifting coefficient of 22. We used an analysis window of 30ms and a time-shift of 10ms.

The system uses a GMM-HMM framework whereby 16 clusters (states) are initialised with speech data by dividing the speech frames uniformly into 32 chunks and using 2 chunks (from different points in the data) to initialise each of the 16 GMMs. For example, the first cluster will be assigned data from the first chunk and the seventeenth chunk. Given these models, the system then segments all speech using the Viterbi algorithm with a forced minimum duration constraint of 250ms. After segmentation, the models are retrained, and this is followed by a clustering step in which the most similar clusters are merged – the choice of which clusters to merge is based on

the Bayesian Information Criterion (BIC). The putative merged model has a complexity (i.e., number of model parameters) equal to the sum of the complexity of the models being merged, which means that a penalty factor parameter is not required. Details of this technique can be found in (Ajmera and Wooters, 2003).

The process of segmentation and clustering is then iterated until a termination criteria is met: for example, all BIC scores for putative cluster merges are negative.

4.3 Experiments

We used the data from RT06, RT07 and RT09 in a series of experiments designed to control for the influence of separate system components by replacing them with oracle or *ideal* equivalents. Often, in the literature, we see that results on the RT corpora are presented by campaign year. However there are no inherent differences in terms of task or conditions and, while inter-meeting variations are observed in results, no inter-campaign variations are. Therefore, results for all meetings are presented here together as a single set. Fig.4.1 shows an outline of the system design and also illustrates information at each stage that can be replaced by oracle knowledge.

4.3.1 End-to-End

This is the fully automatic unsupervised system. The system is not provided with any oracle knowledge. Apart from a few heuristically-selected parameters (as is the case for all diarization systems), it is completely unsupervised. Speech segmentation is done automatically using the QIO-Aurora tool-kit. These are the standard conditions for speaker diarization and forms the baseline to which all experiments in this chapter are compared.

As we can see from Table 4.1 and Fig.4.2 the performance of this system is not consistent across meetings and the average results (33.58% DER across RT06/07/09) are far from acceptable for most practical applications.

4.3.2 Number of Speakers

One of the canonical conditions for the speaker diarization task is that the number of speakers in the audio to be diarised is unknown. This is key information during the clustering stages of diarization as it can dictate when to stop merging or splitting clusters. For example, in the case of agglomerative clustering, over-merging will lead

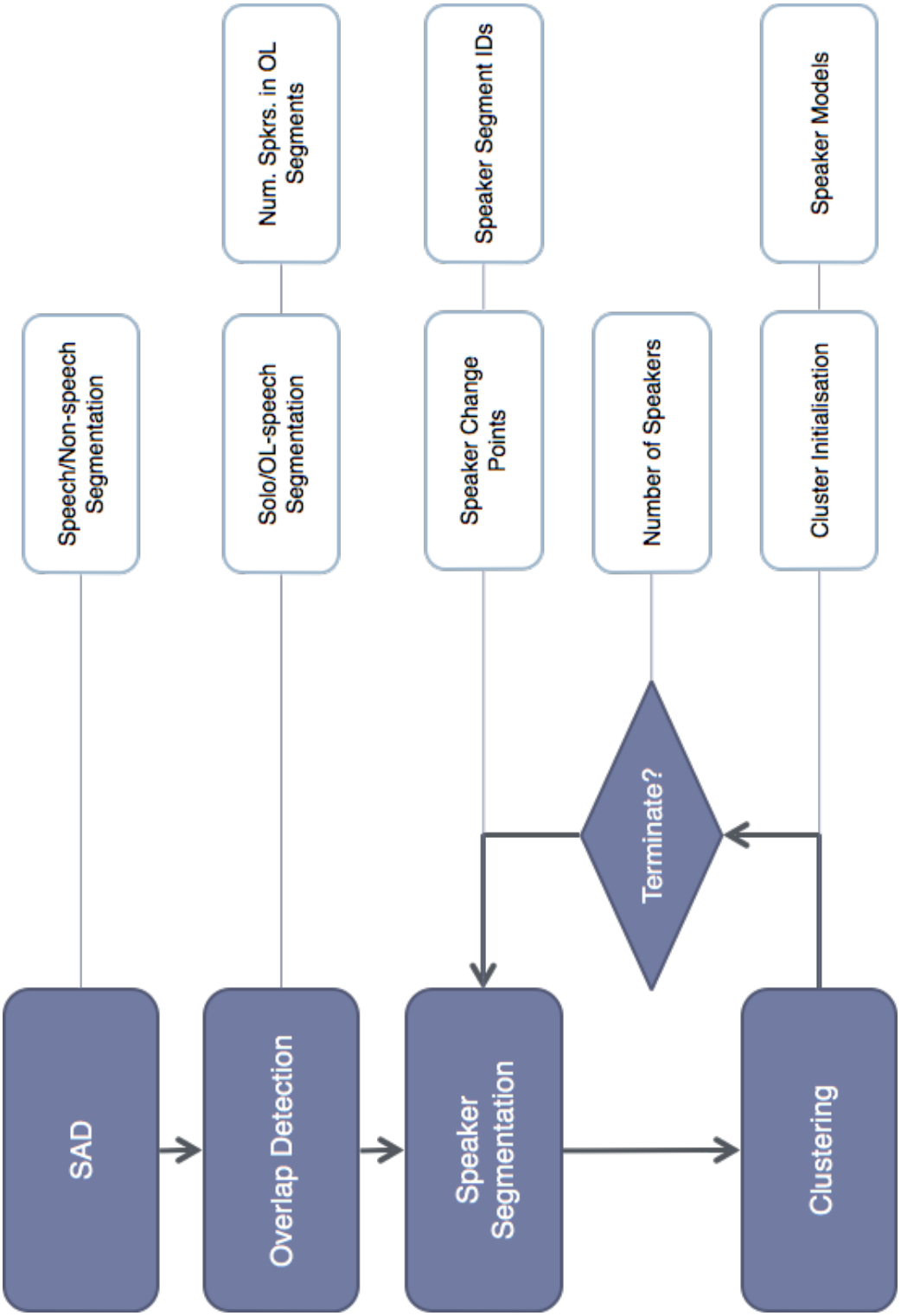


Figure 4.1: A typical speech segmentation and speaker diarization system. The blue shaded boxes show system components and the un-shaded boxes show the different attributes that are associated with each of them. This is the information we can with provide or derive in our oracle experiments.

Table 4.1: Oracle speaker diarization results averaged over NIST RT06/07/09.

Oracle Components	E_{spkr}^{miss}	E_{spkr}^{fa}	E_{spkr}	DER(%)
None (End-to-End system)	14.86	2.34	16.38	33.58
numspks	14.86	2.34	16.68	33.88
SAD	10.50	0.00	17.58	28.09
SAD+idealclust	10.50	0.00	15.27	25.78
SAD+numspks+idealclust	10.50	0.00	15.46	25.96
SAD+idealmodels	10.50	0.00	6.78	17.29
SAD+idealmodels+OLtop0	19.04	0.0	0.07	19.11
SAD+idealmodels+OLtop0	10.2	0.0	0.99	11.19
SAD+idealmodels+OLtop0	1.35	0.0	5.21	6.56
SAD+idealmodels+OLall	0.0	0.0	5.87	5.87

to the speech being labelled with too few speakers and typically this results in a sharp increase in DER.

In order to investigate this, the baseline system was provided with the true number of reference speakers and this was used as the termination point for clustering. All other parts of the system remained the same. In this condition the clustering stops at exactly the known number of speakers per meeting.

However as the results in Table 4.1 show, oracle knowledge of the number of speakers alone *numspks* does not necessarily translate into better performance and in many cases actually degrades it. One reason for this is that slightly too many clusters can actually be better, if each speaker is well represented – i.e., speaker-attributed clusters have high purity and the *extra* clusters are small. Continuing until the oracle number of speakers is reached may result in incorrect cluster merges and a breakdown of speaker homogeneity.

4.3.3 Speech Activity Detection (SAD)

The majority of speaker diarization systems are preceded by an SAD segmentation. In doing so, we are able to exclude most of the data that does not contain speech which helps to prevent an overall clustering into speech/non-speech classes. The effect of an ideal SAD can be illustrated by providing the baseline system with an oracle speech/non-speech segmentation derived from the reference transcription. The system

is then run as normal. As the results in Table 4.1 show, the E^{fa}_{spkr} error is reduced to zero and the E^{miss}_{spkr} error is reduced to the remaining contribution of overlapping speech which the system does not consider. This alone provides a substantial reduction in overall DER. However, it is also worth noting that this does not propagate on to a better E_{spkr} as this in fact increases. This suggests that while a good SAD is important, the function of the diarization algorithm is not highly dependent on it. Importantly, this also indicates that it is safe to use oracle SAD when investigating other components of a diarization system.

4.3.4 Clustering

Normally the initial seed clusters to the algorithm are derived by uniformly dividing the data and attributing a portion of it to each cluster. By the nature of the discourse during a meeting, for example, the initial clusters are likely to contain a mix of data for different speakers. The clustering process relies on some of these clusters being more pure for one speaker than others, and that the cluster purity will increase with each iteration of segmentation and merging.

An oracle experiment to investigate this was devised whereby the clusters were instead each initialized with homogeneous data belonging to only one speaker. In order to maintain a similar amount of data in each cluster as in the End-to-End condition, each speaker's data is split across a number of clusters based on the proportion of his or her speaking time. The number of initial clusters is the same as in the End-to-End condition. Ideally, at each iteration, the algorithm should choose to merge clusters in such a way as to maximise cluster purity – that is, belonging to the same speaker. This condition allows us to check how sensitive the clustering process is to initialization.

While the average E_{spkr} shows a reasonable reduction in Fig.4.2, the inclusion of ideal cluster initialisations has greatest effect for meetings where the End-to-End system gave a high E_{spkr}. This suggests that poor cluster initialisation, whereby initial clusters all have a low purity, may be non-recoverable.

There already exists some work that aims to mitigate for poor initialisation by exploiting cluster purity during model retraining (Anguera et al., 2006a; Bozonnet et al., 2010). These methods retrain cluster models at each iteration in the clustering stage by using only the top-N frames of data assigned to that cluster in terms of model likelihood. This can help to assure that the resultant clusters have a higher purity, however it does not guarantee that they are *speaker*-pure. It may also require the introduction

of further heuristic parameters such as the percentage of data to use.

4.3.5 Models

The speaker models are rather simple: Gaussian mixture models with simple duration modelling. It is likely that these models were originally chosen due to their similarity with typical ASR models. Therefore, it is reasonable to ask whether these are adequate for the diarization task. In this experiment, the reference transcription was used to create optimal speaker models by creating a number of clusters equal to the known number of speakers and training each with data from one speaker only. This way, we can discover whether the models themselves and the associated acoustic feature set have sufficient speaker discrimination power for the task.

We also varied the amount of data used to train these ideal models, from 10% of the available data per speaker up to 100%. We examine the effect of further iterations of re-segmentation and re-training (no merging) too, from 1 iteration (i.e., segmentation with ideal models) up to 5 iterations of re-segmenting and re-training. These iterations *should* improve the models (or, in the 100% data case, do nothing).

As we can see from the results in Table 4.1, providing the system with ideal models trained on each speaker's data substantially reduces E_{spkr}, confirming the models do work. Fig.4.3 shows the effect of varying the amount of data used to train the models. As little as 10% improves performance over the baseline. Interestingly, more iterations *degrades* performance. This suggests cluster purity is critical to the clustering process: impurities introduced at each iteration cannot be accommodated, and the models do not recover.

4.3.6 Overlapping Speech

The diarization task can face significantly greater difficulty when speakers overlap. This is, however, a very common phenomena that appears in many settings. An example of a typical meeting with overlapping speakers is shown in Figure 4.4. Here we can see that there are some long parts when only a single speaker takes precedence, but there are also other parts of dense discourse where, at times, all 4 speakers may overlap.

Figure 4.5 shows the distribution of solo speech and overlap speech across the RT meetings. Here we can see that almost all meetings contain a substantial amount of overlap (an average total of 18.3%), the majority of which comprises *two*-speaker

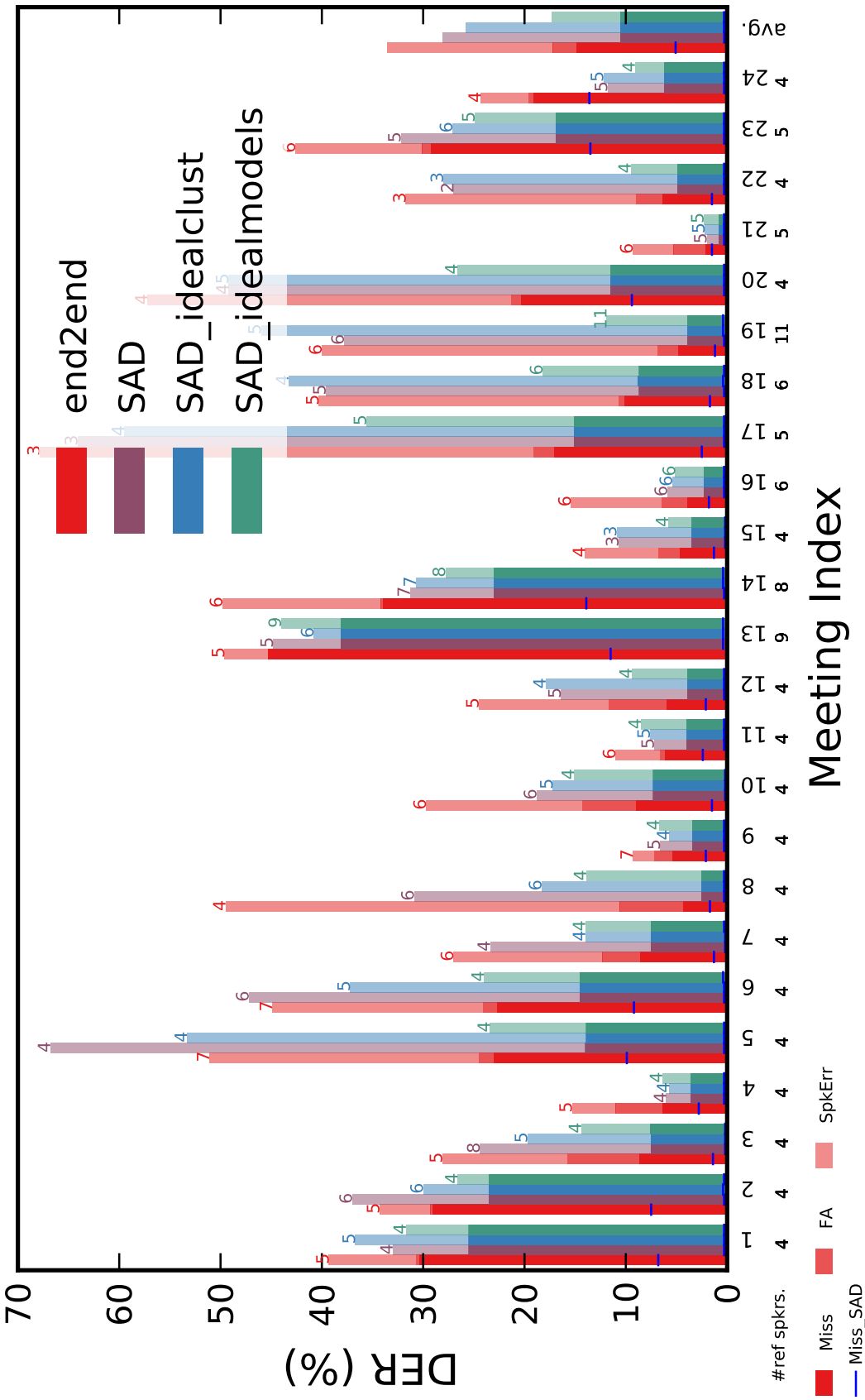


Figure 4.2: DER results for NIST RT06/07/09 meetings. The decaying opacity of the bars show how the error is composed of E^{miss}_{spkr} and E^{fa}_{spkr} speaker time (inclusive of overlap) as well as E_{spkr} (due to speaker id misclassification). The blue horizontal bar indicates the amount of missed speech error contributed by SAD, E^{miss}_{spch}. Above each bar the number of hypothesised speakers is shown and the reference is provided parallel to the x-axis.

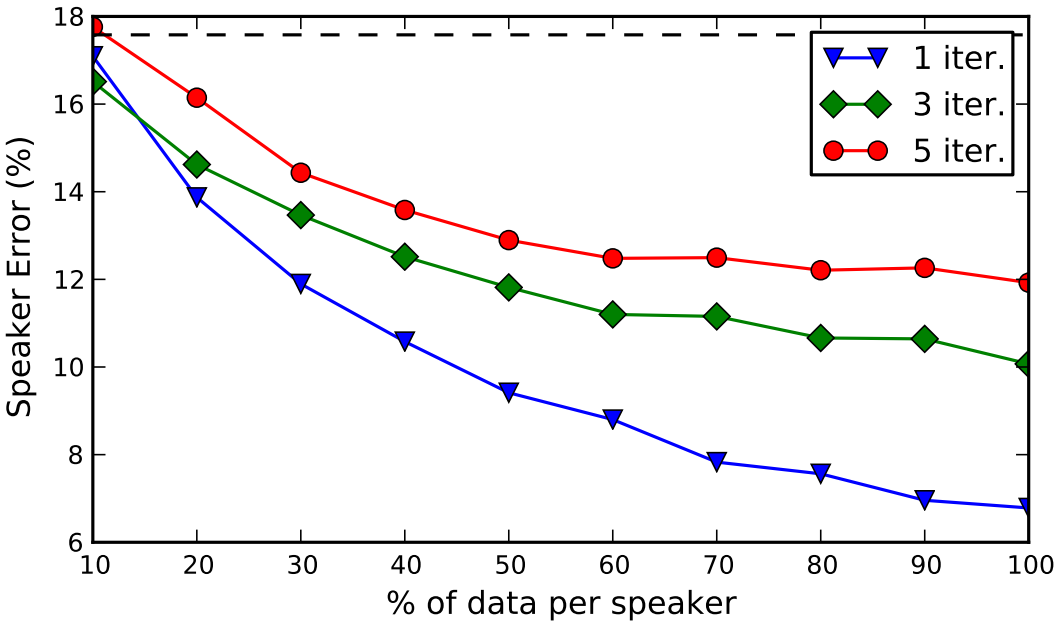


Figure 4.3: The E_{spkr} obtained when creating ideal speaker models using varying amounts of data, up to and including all the data. The three lines show the results for segmentation using these models directly ('1 iter.') and when re-segmenting and re-training the models on all the data in the usual iterative fashion ('3 iter.' and '5 iter.'). The dashed line is the DER of the Oracle SAD system *SAD*.

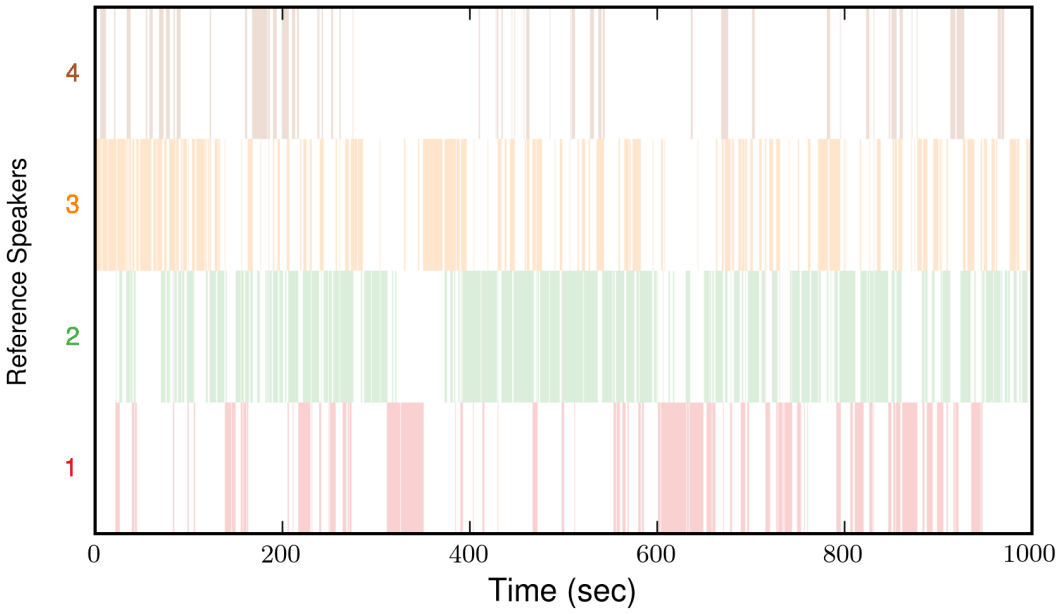


Figure 4.4: Example of typical discourse in an RT meeting. Here there are 4 speakers who overlap with each other regularly. In some cases, this overlap can include all 4 speakers simultaneously.

overlap. For a system which does not consider such occurrences, the associated error will always remain.

While SAD is used prior to diarization to classify the signal into speech and non-speech (e.g. silence, music, noise, etc.), we could also benefit from knowing if each speech segment contains one speaker (solo/non-overlap speech) or multiple (overlap speech). This experiment employs such a three-class segmentation derived from reference transcriptions. We first use the ideal models to select the most likely speaker for each solo speech segment. Then we revisit overlapping segments and attribute more speakers to them, based on the top N most likely models. Thus, overlap speech is ignored during model training, but is still labelled with speaker ids.

As we see in Table 4.1, ignoring overlap (SAD+idealmodels+OLtop0) is costly (19.11% DER in this case), this is primarily due to the extra E^{miss}_{Spkr}. By attempting to get at least 1 speaker correct per overlap region (SAD+idealmodels+OLtop1), we nearly halve that error (11.19% DER vs. 19.11% DER). An average minimum 10.50% DER is always incurred if only 1 speaker at a time is possible, but by getting at least the 2nd speaker correct (SAD+idealmodels+OLtop2), we nearly halve the error again (6.56% DER vs. 11.19% DER). By getting all overlap speakers, we see a less significant gain

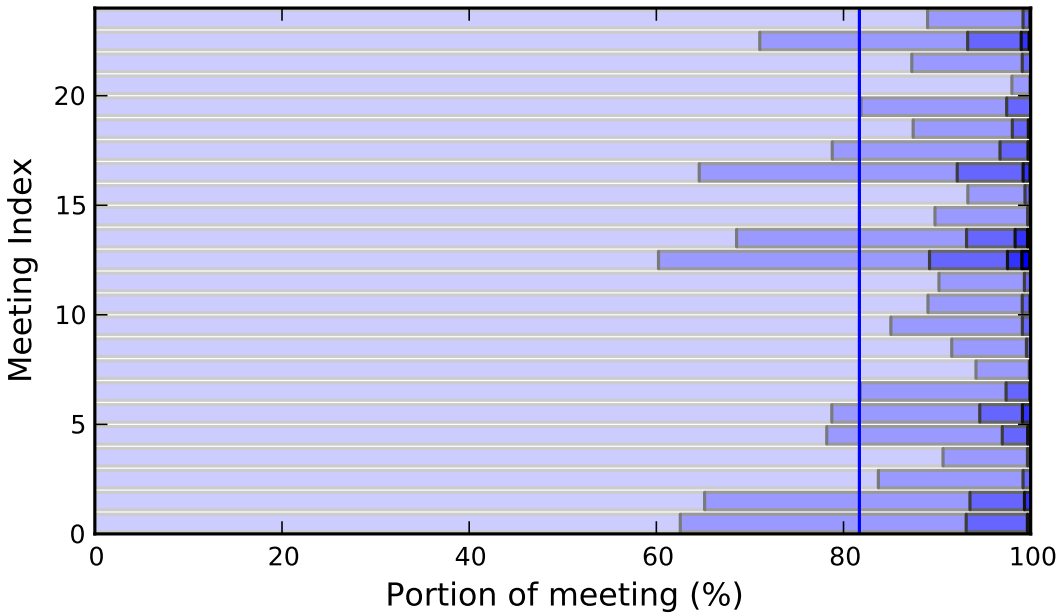


Figure 4.5: The distribution of overlapping speech in the RT06/07/09 meetings. The least opaque represents a single speaker and each subsequent level of opacity represents one more speaker overlapping. The vertical blue line shows the position of the average amount of total overlap (18.3%)

(5.87% DER vs. 6.56% DER). This shows that by estimating at least 2 speakers in overlap regions, we can cover most of the overlap distribution shown in Figure 4.5. Indeed, it may be better to limit an overlap-estimation systems estimated speakers to a maximum of 2, in order to avoid incurring extra false alarm errors, E^{fa}_{spkr}. We can also see from Figure 4.6 that this performance gain is consistent across all meetings, which suggests that overlap detection and attribution is of universal benefit.

4.4 Conclusions

4.4.1 Relation to Huijbregts and Wooters (2007)

As Huijbregts and Wooters (2007) also found, results are highly dependent on the evaluation data (i.e., high variation in DER between meetings) and some system components can be sensitive to the performance of preceding ones. Like them, we found that SAD can be a major contributor to DER by directly contributing E^{miss}_{spch}, but we would add that subsequent components actually have little dependence on its performance.

4.4.2 New Findings

One of the key findings from our experiments is the importance of estimating the speaker models on pure data: speech from just one speaker. If this could be achieved, dramatic reductions in DER would result (Fig.4.3). Even if only a fraction of the data for each speaker could be reliably identified, free from the polluting effects of data from other speakers, than large improvements would still be expected. Methods for estimating some form of *confidence* in speaker homogeneity when seeding clusters with data should therefore work well, even if that entails rejecting a large proportion of the data.

Our ideal models are strong enough to allocate multiple speakers to overlap regions. So another focus of future research should be in overlap-speech detection. Systems which do not consider overlap will always concede substantial error. Current attempts to do this, e.g. (Boakye et al., 2008), normally do not involve novel methods and simply exploit GMM classifiers. This yields a performance that is not yet sufficient for practical application. Having either a good overlap detector or at least being able to extract enough non-overlap speech segments with high precision would benefit this process.

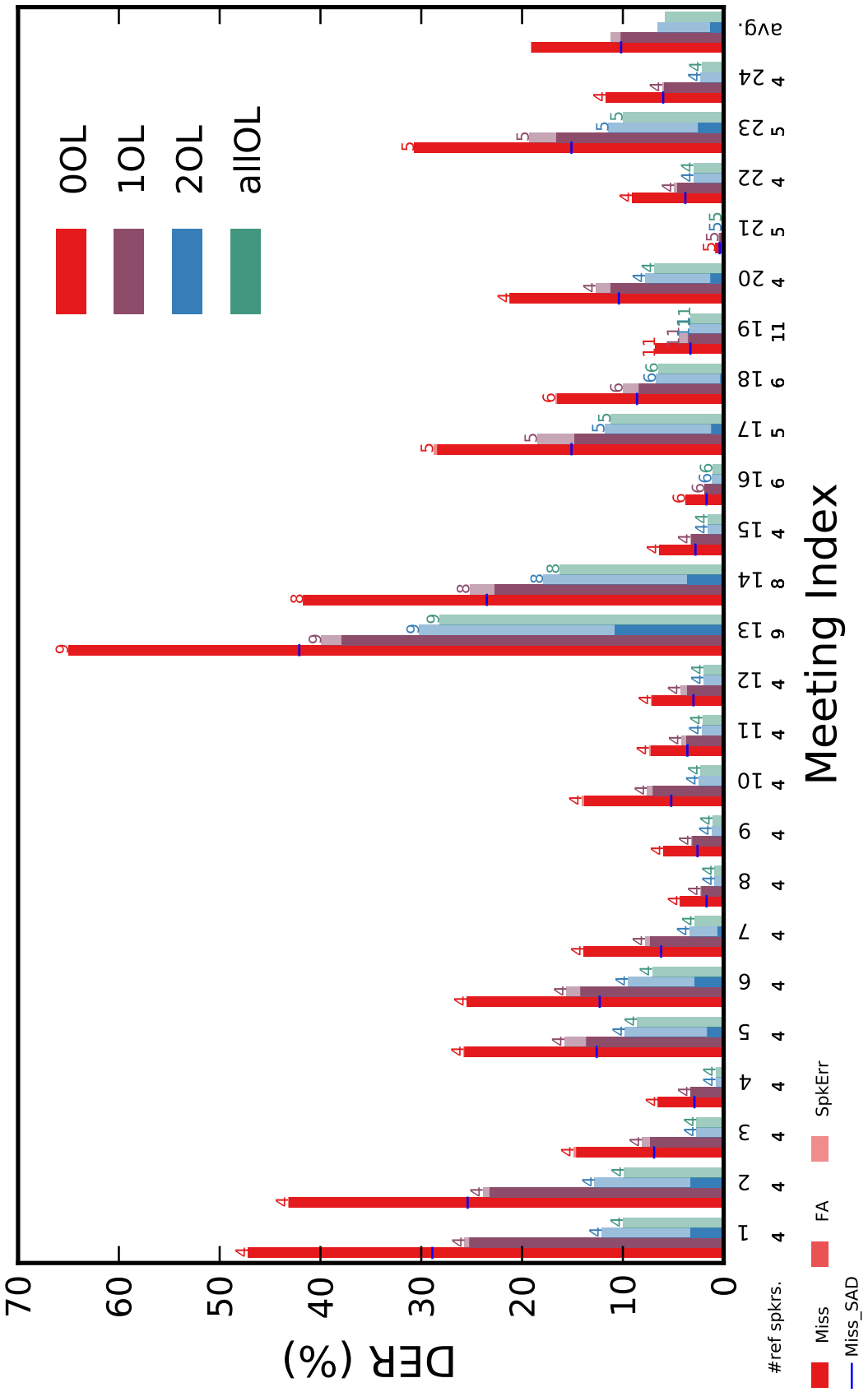


Figure 4.6: DER results for NIST RT06/07/09 meetings when perfect models are used to score known oracle segmentations (see Figure 4.2 for plot interpretation).

One of the take-home messages, given that further iterations degrade models that were initially pure (Fig.4.3), is that the final set of speaker models should not necessarily be trained on all data to be diarised, but only on reliably-identified pure data. The models have been shown to be very sensitive to how well they are initialisation and are often unable to recover if clustering tends towards a sub-optimal solution. It would therefore also be beneficial to at least be able to seed the models with data that has a high confidence of being speaker homogeneous.

Chapter 5

Exploiting Non-acoustic Information to Improve Speech Segmentation for Speech Recognition and Machine Translation

5.1 Introduction

From our oracle experiments in Chapter 4, we found that the performance of speech segmentation contributes significantly to Diarization Error Rate. We also know that it has a direct impact on speech recognition performance. As it is the often one of the first components in any speech processing system, errors made during speech segmentation are often non-recoverable. For this reason, we decided to look into how we could improve the performance of speech segmentation. We began by simply asking the question: *what is a speech segment?*

In Chapter 1 we defined speech segmentation as the problem of finding when speech is observed in a given recording. While this may at first seem like a relatively simple goal it is in fact a non-trivial problem to define. As speech does not strictly follow the same rules we find in written language, such as sentence breaks, it can often be highly subjective as to what constitutes an appropriate segmentation of speech for a given task. The vagueness and high-order decision processes that surround these concepts make it challenging to design an effective automatic speech segmentation system.

Most automatic speech segmentation methods work by identifying speech and non-speech regions based on acoustic evidence alone e.g. contrasting energy levels or spectral behaviour (Boyd and Freeman, 1994; Ramırez et al., 2004; Tucker, 1992). Some more recent research has improved upon this foundation by using richer feature sets that are more suited to the task or include long-term dependences (Ye et al., 2013; Graciarena et al., 2013; Tsiartas et al., 2013). Others have begun to apply deep learning techniques which can garner more discriminative features and improve robustness (Ryant et al., 2013; Zhang and Wu, 2013; Eyben et al., 2013). However, all of these methods still only consider the acoustics and are not necessarily exploiting the underlying structure of the spoken language.

Human transcribers, on the other hand, are capable of segmenting speech by exploiting a greater wealth of prior information such as syntax, semantics and prosody in addition to such acoustic evidence. As a result, human transcribers may opt to ignore acoustically motivated ‘breaks’ in speech in favour of maintaining longer segments based on semantic knowledge – just because there is a short pause between words, it does not mean that we should break the utterance. Such an informed segmentation can greatly influence subsequent system components that have been explicitly designed to exploit the patterns and structure of natural language, e.g. the language models used in

automatic speech recognition (ASR) or machine translation (MT).

Previous work on segmentation of sentence like units has looked at using features such as prosody (Stolcke et al., 1998), language model scores (Stolcke and Shriberg, 1996) (Matusov et al., 2006), translation model scores (Matusov et al., 2007) and syntactic constituents (Roark et al., 2006). (Ostendorf et al., 2008) presents a review of some of this work and also motivates tuning the segmentation of speech to the task at hand as we do in this thesis. There has been some previous work which attempts to exploit some of these cues (Rybach et al., 2009). However, for the work in this chapter, we have limited our focus to the use of statistics of utterance durations and present a novel way of exploiting this to select a globally optimal sequence from acoustically motivated 'break candidates'. We find that this yields an advantage over the use of local acoustic information alone at putative pauses in the speech. We also find indications that the optimal setting of the segmentation parameters varies with the ultimate downstream task – transcription (ASR), translation or speaker diarization.

In order to demonstrate the influence of segmentation on transcription and translation we present experimental results on TED talks[1] from the IWSLT campaign. We then present experimental results on the NIST RT dataset to show how segmentation influences speaker diarization.

5.2 Utterance-break Modelling

While the automatic speech segmentation methods that we initially used are only able to segment on an acoustic basis, they actually perform this task very well at a frame level. When compared to the manual segmentation we found the False Alarm rate to be very low (2-3%) while the dominant error is Missed Speech. We found that often this missed speech was actually pauses inside utterances which, from a purely acoustic perspective, are indeed non-speech. This can be problematic for speech recognition in particular as missed speech cannot be recovered for this task and breaking on such pauses can affect the contingency of the language model. False alarms, on the other hand, may result in word insertions, but if the ASR models have non-speech states (such as a null, noise, silence, etc.) then it is possible that it will just be ignored and no downstream error will be introduced.

It follows that if empirical evidence shows automatic segmenters work very well at frame-wise classification, the main issue is being able to distinguish when a non-

[1] http://www.ted.com

speech segment is simply a pause inside a speaker's utterance or a true 'break' between utterances as would be judged by human annotators. Figure 5.1 shows an illustration of this. In this example, the automatic system, as represented by the segmentation S^{hyp}, has hypothesised non-speech 'breaks' for some parts of the signal where indeed there was no audible speech. However, as this was a complete utterance it should not have been broken up in this way. The manually annotated reference, S^{ref}, shows that the human annotator has marked the whole utterance as one long segment. They are able to do this as they can infer from the linguistic content that the acoustic pauses are not in fact to be interpreted as a break in utterance. This is very common behaviour for acoustically motivated automatic speech segmentation systems.

Often these pauses are quite short so, intuitively, a naïve approach might be to simply alter the minimum duration constraint for non-speech regions. However, in practice we find that this simply shifts the balance from Missed Speech to False Alarm errors by removing more potential breaks, quickly resulting in segments that are too long. A significant part of this behaviour is due to the fact that such systems are only able to make local decisions about whether or not to include a non-speech segment.

Another approach is to explicitly make the assumption that acoustic speech segmentation, S^{hyp}, is always correct when there is audible speech i.e. it has a high precision for speech classification. Then, given an ideal reference segmentation S^{ref}, we can then make the assumption that $S^{hyp} \subset S^{ref}$ i.e.:

$$S^{hyp} \approx S^{hyp} \cup S^{ref} \tag{5.1}$$

This is equivalent to thinking about the problem as that of *smoothing* – we have a segmentation that has too many breaks and we need to smooth over some of them.

To remedy this problem, we propose to investigate methods for globally optimising the smoothing of an acoustic segmentation, incorporating prior knowledge about the likelihood that non-speech breaks should be included or excluded given their temporal relationship i.e. the duration between them.

5.2.1 Break Candidates

To begin with, we will introduce the concept of a candidate break sequence B, which can be thought of as the interval set equivalent of the non-speech set implied by a given speech segmentation set S^{hyp}:

$$B = I(\mathbb{T} \setminus S^{hyp}) \tag{5.2}$$

Figure 5.1: An example recording of a complete utterance that has been wrongly over-segmented by an automatic system S^{hyp}. The human annotator's reference S^{ref} shows that they have inferred from the linguistic component that this was a complete utterance and have duly marked it as one long segment despite the acoustic evidence to the contrary.

Here, T is the set of all possible values for time t in a given recording and $I(S)$ is the function that maps from a segmentation set to an ordered interval set (see Section 1.4). By taking the relative complement of $\mathbb{T}$ with S^{hyp}, we essentially get the *non-speech* segmentation.

The globally optimal sequence of utterance breaks, B^*, should be as close as possible to the equivalent break sequence from the ideal reference segmentation and can be defined in a similar way:

$$B^* \approx I(\mathbb{T} \setminus S^{ref}) \tag{5.3}$$

Ideally the candidate sequence should be dense enough that it includes a good optimal sequence as a sub-set so we would therefore require that $|B| \geq |B^*|$, where $|B|$ and $|B^*|$ are the number of breaks in the candidate and optimal sequences respectively. This allows our assumption in Equation 5.1 to hold. The candidates themselves can be determined by any kind of initial segmentation method such as an existing acoustically motivated speech segmentation algorithm.

5.2.2 Utterance-break Prior

In order to make decisions about whether or not to include a candidate break, we need to know the prior probability of a break, which we condition on the time since the last break was observed. This may be derived from a statistical model of segment durations. Figure 5.2 shows a histogram of speech segment durations for a development set of lecture data (the IWSLT 2010/11 dev. sets). We can see from the histogram that the data has a left-skewed log-normal distribution. This is to be expected as spoken utterances have several natural constraints. Physical constraints prevent us from talking endlessly as we need to pause regularly for breath. We also structure our speech in order to communicate better. For example, it is often better to 'break up' chunks of information with pauses to allow the listener to process it. From the histogram we can see that the majority of utterances last between 2 and 4 seconds. Very short utterances of only a few words are also rare as they do not contain enough information. There are of course exceptions as evidenced in this example by the tails of our distribution.

We investigated the use of log-normal and gamma distributions to represent the behaviour of the data. Ultimately, we chose the former as it provided a slightly better fit as shown in Figure 5.2. To derive the break likelihood prior we investigated the use of both the Probability Distribution Function (PDF) and the Cumulative Density Function (CDF) of this distribution as shown in Equations 5.4 and 5.5 where d is the

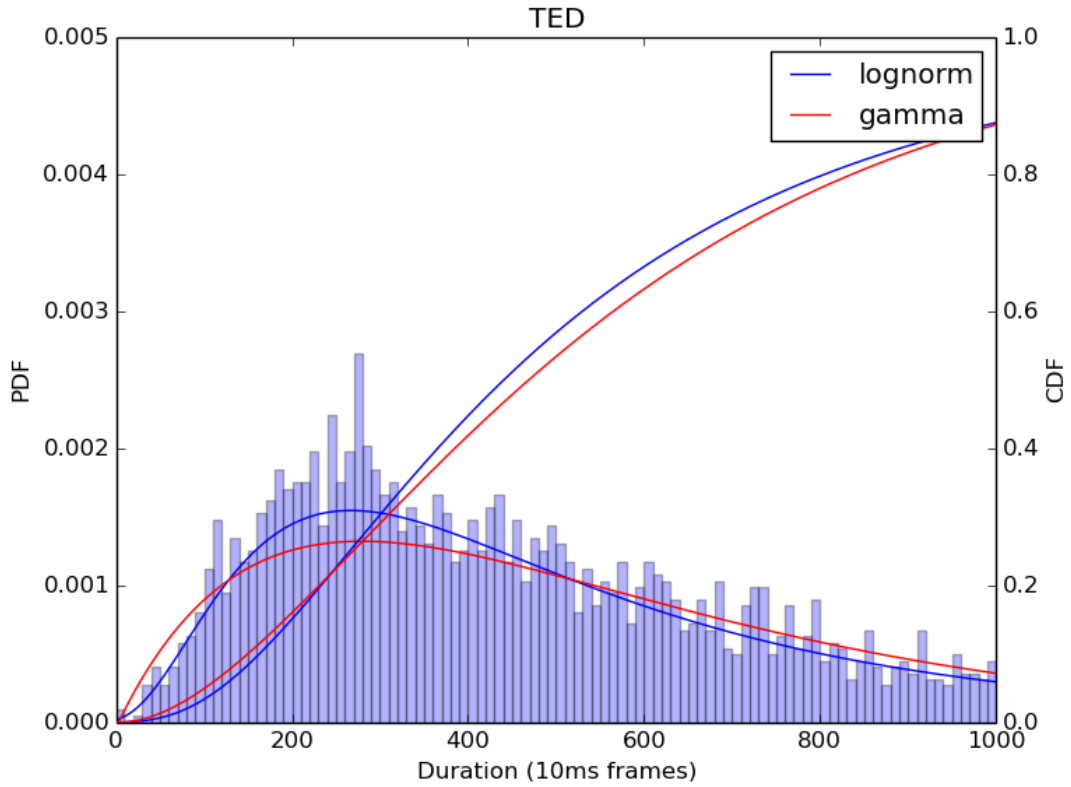


Figure 5.2: *Log-normal vs. Gamma PDFs and CDFs fitted to speech segment durations of TED dev set. These functions can be used to devise the prior likelihood of a new break given the duration since the last break.*

duration since the previous break.

We considered both the PDF and the CDF functions as priors as they can provide slightly different kinds of information for making the decision about whether or not a break should be included. The CDF essentially provides the likelihood of a break given the duration since the previous break in isolation i.e. the longer the duration the more likely another break should be observed. However, the PDF describes the likelihood of such a break in a more global sense i.e. we want to choose breaks that will maximise the distribution we observe over manual segmentation.

$$f_{brk\_utt\_PDF}(d) = \frac{1}{d\sigma\sqrt{2\pi}} e^{-\frac{(\ln d-\mu)^2}{2\sigma^2}} \tag{5.4}$$

$$f_{brk\_utt\_CDF}(d) = \frac{1}{2} + \frac{1}{2}\text{erf}\left[\frac{\ln d-\mu}{\sqrt{2}\sigma}\right] \tag{5.5}$$

We were interested in how such a prior may vary according to the domain. There-

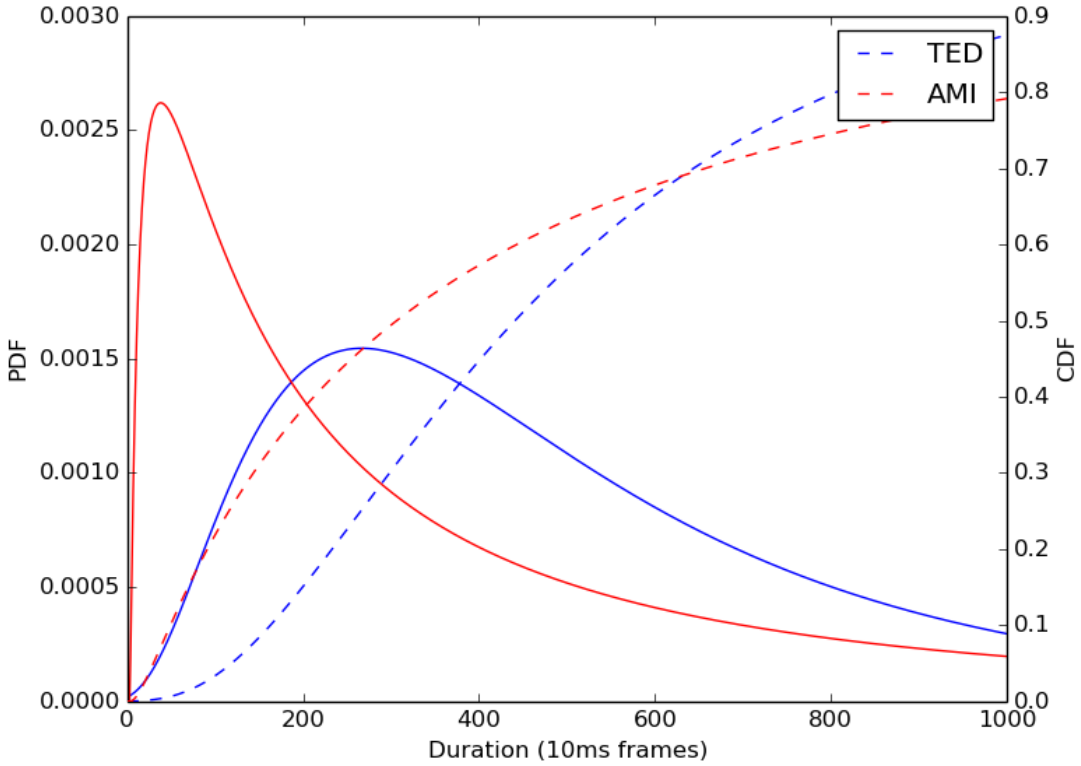


Figure 5.3: *Log-normal PDFs and corresponding CDFs for speech segment durations estimated from examples of manual segmentations from TED and AMI data.*

fore, as a contrast to the prepared, rehearsed, single-speaker speech that is found in TED talks, we also looked at the distribution of speech segments for a set of AMI scenario meetings. These meetings contain multiple speakers discussing a given common task whereby the speech is unprepared and spontaneous. From Figure 5.3 we can observe that the distribution has a much lower mean, illustrating the intuition that speech segments are generally shorter during such dynamic group discourse. This suggests that the break likelihood prior could be adapted for different domains to achieve optimal performance.

5.2.3 Sequence Decoding

We want to find the optimal break sequence B^* that maximises the product of the acoustic likelihood, $P(A|B)$, and the prior likelihood, $P(B)$.

$$B^* = \arg\max_B P(A|B)P(B) \tag{5.6}$$

In order to determine the globally optimal sequence of utterance breaks B^* we consider a sequence of candidate breaks $B = \{b_0, \ldots, b_{|B|}\}$ as a semi-Markov process whereby each candidate is a state.

This is illustrated in Figure 5.4. We can then perform Viterbi decoding over a sparse $|B| \times |B|$ trellis, whereby each position i moves, not through uniform time segments, but through the indices of B. Each position j allows us to consider the transition arriving at break b_i from b_j. As the break candidate states are only forward connected we can only arrive at a given break from a previous break, hence $j < i$. We keep a vector of tuples T that records the start and end frame indices of each break, this allows us to calculate the duration between any pair of break candidates $d_{i,j} = t_{i,start} - t_{j,end}$.

We also use the sums of the frame-level log-likelihoods from the speech/non-speech segmenter for acoustic features X, where x_i is a vector representing all the frames that are in break b_i. We use this to create a posterior probability P_{brk_aco}, that represents the acoustic probability of a given break. From here, we created two different implementations of the algorithm which are described as follows:-

As the duration between the break candidate under consideration and earlier ones increases, it will become very unlikely that such long term transitions would occur. In practice, we can therefore afford to prune the lattice by ignoring transitions from earlier states that are further away than a prescribed maximum segment duration, δ. As such, for each index i we only consider transitions from states where $d_{i,j} \leq \delta$.

Implementation A (ImpA)

The first attempt to implement the break smoothing algorithm created the posterior probability P_{brk_aco} as shown in Equation 5.7. The purpose of the normalisation is that we want the *non-speech* acoustic likelihood of breaks to increase with duration so that long breaks are less likely to be skipped over. This happens as, should our speech/non-speech models function correctly, $\ell_{nonspch}(x)$ and $\ell_{spch}(x)$ will both decrease with the length of x, however $\ell_{spch}(x)$ decreases more rapidly than $\ell_{nonspch}(x)$ and as such the fraction in Equation 5.7 increases.

$$P_{brk\_aco}(i) = \frac{\ell_{nonspch}(x_i)}{\ell_{nonspch}(x_i) + \ell_{spch}(x_i)} \tag{5.7}$$

Therefore, the log probability of the partial sequence that has a break at i is formalised as

$$v_i = \max_{j<i} [v_j + \log(f_{brk\_utt}(d_{i,j}))^{\alpha}] + \log P_{brk\_aco}(i) \tag{5.8}$$

with $v_0 = 0$. For each candidate i we store the identity of the state j which maximises Equation 5.8. This allows to B^* to be recovered by a backtrace procedure.

As there is a difference in the dynamic range between $f_{brk\_utt}(d)$ and $P_{brk\_aco}(i,j)$, we included a scaling parameter α, which we tuned heuristically.

Note that we used $f_{brk\_utt}(d) = f_{brk\_utt\_CDF}(d)$ in this implementation as $f_{brk\_utt\_PDF}(d)$ tended to produce segments that were generally too short. This was likely because legitimate long segments were being penalised by the PDF regardless of how acoustically likely they were.

Ideally, the break prior as modelled by the PDF should be informing the global distribution of breaks and the acoustic model should be informing the local decisions. With this first implementation, we were not properly considering the acoustic cost of *skipping* over candidate breaks that exist between any given b_i and b_j. In order to rectify this we introduced the penalty described in Implementation B.

Implementation B (ImpB)

For the second implementation of the break smoothing algorithm we added the sum of normalised *speech* acoustic likelihood of all breaks between i and j such as to consider also a penalty for skipping these break candidates. This means we are more likely to include long candidate breaks as they are less likely to be skipped, and we will also not create speech segments that are too long as they will have a high penalty for skipping lots of break candidates. This is shown in Equation 5.9

$$P_{brk\_aco}(i,j) = \frac{\ell_{nonspch}(x_i)}{\ell_{nonspch}(x_i) + \ell_{spch}(x_i)} + \sum_{j+1}^{i-1} \frac{\ell_{spch}(x_j)}{\ell_{nonspch}(x_j) + \ell_{spch}(x_j)} \tag{5.9}$$

For this implementation we now found that $f_{brk\_utt}(d) = f_{brk\_utt\_PDF}(d)$ produced the best results and a resultant segment length distribution that was closer to the real PDF. This is because legitimately long speech segments (those that are long but have few actual acoustic breaks) are more likely and are therefore less penalised by the PDF.

We also found that by scaling up the original features at frame-level before calculating $P_{brk\_aco}(i,j)$ we could achieve a much lower difference in the dynamic range as compared with $f_{brk\_utt}(d)$ and as such we could eliminate the scaling parameter α from the decoding algorithm.

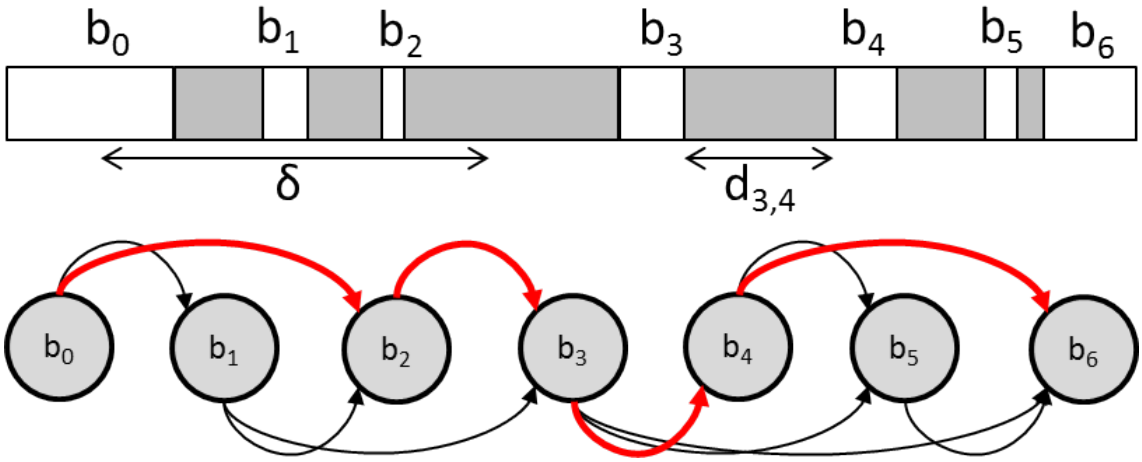


Figure 5.4: *An example of a candidate break sequence and associated state topology. We can see that the states can only feed forward and some long-term transitions have been pruned such as $b_0 \rightarrow b_3$ as $d_{0,3} > \delta$. The transitions highlighted in red show an example optimal break sequence $B^* = \{b_0, b_2, b_3, b_4, b_6\}$*

5.3 ASR and MT - Experiments

Our initial experiments investigated the effect of our utterance-break prior smoothing algorithm on speech recognition and machine translation.

5.3.1 Data

For evaluation, we used the data made available for the IWSLT evaluation campaigns(Cettolo et al., 2012). This comprises a series of TED talks that have been divided into development sets (dev2010 and dev2011) and a test set (tst2010), each containing 8-11 talks. The talks average just under 10mins in length and each contains a single English speaker (either native or non-native). The talks are manually segmented and transcribed at the utterance level. We also had manual English-French translations for evaluating the MT system component.

We chose to begin with this data as there is only one speaker present in each talk. This simplifies our initial investigation as it controls for the effect of speaker changes on speech segmentation behaviour. As we are trying to produce an automatic segmentation that is closer to a manual segmentation, the TED data also allows us to also look at the effect of speech segmentation on machine translation – an NLP task that typically expects 'sentence-like' segmentation.

5.3.2 Speech Segmentation Systems

5.3.2.1 Manual

Here we simply pass the manual segmentation to the ASR and MT systems directly in order to form the oracle standard with which to compare our automatic speech segmentation methods.

5.3.2.2 SHOUT

This is the SAD component of the SHOUT toolkit, the details of which are described in Section 3.1.3.1. We used this in order to compare our results with a competitive state-of-the-art method. The default settings for the toolkit were used.

5.3.2.3 Baseline segmenter

Our baseline system, labelled *Simple* in the tables, is identical to that used for our IWSLT 2013 transcription system (Bell et al., 2013; Driesen et al., 2013) and comprises a GMM-HMM based model which is used to perform a Viterbi decoding of the audio. Speech and non-speech are modelled with diagonal-covariance GMMs with 12 and 5 mixture components respectively. We allow more mixture components for speech to cover its greater variability. Features are calculated every 10ms from a 30ms analysis window and have a dimensionality of 14 (13 PLPs and energy). Models were trained on 70 hours of scenario meetings data from the AMI corpus using the provided manual segmentations as a reference. A heuristically optimised minimum duration constraint of 500ms is enforced by inserting a series of 50 states per class that each have a transition weight of 1.0 to the next, the final state has a self transition weight of 0.9.

5.3.2.4 Break Smooth

Here we introduce our utterance-break prior model. In order to establish the candidate break sequence we use the system in Section 5.3.2.3 to do an initial segmentation pass over the data. The only exception is that the minimum duration constraint is reduced to 100ms. If used directly, this would normally perform very poorly as an SAD but when used as input to the subsequent break smoothing we have three advantages over the original constraint: better guarantee of enough candidates to find an optimal solution, the ability to find shorter speech segments ($\geq$100ms), and more accurate end-points for

segments between 100-500ms. The break likelihood prior was trained on the speech segment durations of the dev2010 and dev2011 sets. The maximum segment duration δ is set to 30 seconds.

For *ImpA*, we have also shown results for 2 different operating points of the scaling factor α, 30 and 80. While this parameter is designed to mitigate for the difference in dynamic range with the acoustic model, we found it subsequently functioned as a form of segment duration tuning whereby a greater α results in more break smoothing and hence longer segments.

5.3.2.5 Uniform

As well as our automatic methods we also considered segmenting each talk into uniform speech segments of length N seconds, which is equivalent to having a break of zero length at every interval. This allowed us to check whether or not the benefit of our utterance-break prior may simply be due to an 'averaging' of the break distribution. As the ASR system is still able to do decoder-based segmentation within each given segment, this is also a way of measuring its influence. Here, longer uniform segments leave more responsibility to the decoder for segmentation and at $N = 300$, the maximum segment length for the ASR system, we effectively allow the decoder to do all the segmentation (with potentially a small error at the initial segment boundaries).

5.3.3 Downstream System Descriptions

5.3.3.1 Automatic Speech Recognition (ASR)

ASR was performed using a system based on that described in (Bell et al., 2013). Briefly, this comprises deep neural network acoustic models used in a tandem configuration, incorporating out-of-domain features. Models were speaker-adapted using CMLLR transforms. An initial decoding pass was performed using a 3-gram language model, with final lattices re-scored with a 4-gram language model.

5.3.3.2 Machine Translation (MT)

We trained an English-French phrase-based machine translation model using the Moses (Koehn et al., 2007) tool-kit. The model is described in detail in our 2013 IWSLT shared task paper (Birch et al., 2013). It is the official spoken language translation system for the English-French track. It uses large parallel corpora (80.1M English words and 103.5M

French words), which have been filtered for the TED talks domain. The tuning and filtering used the IWSLT dev2010 set.

The goal of our machine translation experiments is to test the effect that ASR segmentation has on the performance of a downstream natural language processing task. The difficulty with allowing arbitrary segmentations in MT is that automatic evaluation is performed matching MT output with gold reference sentences which have their own manual segmentation. In order to evaluate translations which have different segmentations, we need to align the MT output segmentation with the reference. We use a tool provided by the Travatar (Neubig, 2013) tool-kit which aligns files with different segmentations. It searches for the optimal alignment according to the BLEU score. We use it to align our MT output with a variety of different segmentation models, to our gold reference with manual segmentations. We align each TED talk in the test set separately to maximize performance.

5.3.4 Gold Transcription Mapping

Any improvement a given segmentation provides to the ASR system could subsequently improve the performance of the MT system. However, this makes it difficult to infer how much of the MT performance gain is simply a consequence of a better source transcript as compared with the direct influence of the segmentation itself. To control for this, we used the ASR system to make a forced alignment of the manual transcription in order to gain word-level timing information. We were then able to map any of our given segmentations with the same gold-standard transcription.

5.4 ASR and MT - Results

We present results for all of our end-to-end automatic systems in Table 5.1. Firstly, we note that our *Simple* segmenter is able to significantly outperform *SHOUT*, confirming that we have a competitive acoustic segmenter with which to form the foundation of our experiments. We can then see that our *Break Smooth* segmenters are further able to improve the performance of both ASR and MT over the *Simple* segmenter.

While the results of *ImpA* managed to improve upon our simple segmenter, we were able to make further gains with *ImpB*. This is likely due to the fact that *ImpB* is much better able to segment according to the original utterance prior. This can be seen in Figure 5.5, whereby the PDF of *ImpB* is much closer to the PDFs of the test and dev

sets. It is also worth noting, that by eliminating the need for the scaling parameter α, *ImpB* is also a more robust system than *ImpA*.

The results from the *Uniform 300s* system showed a strong performance for ASR, falling only slightly short of our best performing *Break Smooth* system. We attribute this to the nature of TED talks whereby there is typically very little non-speech (illustrated by the 6.49% FA of *Uniform 300s*), which itself mostly comprises periods of silence of small duration, which is implicitly segmented by the silence HMMs used by the decoder itself. In contrast, however, when we use a uniform segmentation for MT, we find that it does not perform as well despite the good ASR performance. As the MT system ideally expects sentence-like segments, a uniform segmentation will not be practical for these purposes. This also shows that a segmentation that works well for ASR may not necessarily work well for MT and vice-versa.

We should also note how the metrics for scoring SAD cannot be taken as a reliable estimator of downstream performance. For example, *ImpA* $\alpha = 30$ and *Uniform 300s* have a similar total SAD score (6.58% vs. 6.49%) but quite different WER and MT scores. Similarly, *ImpA* $\alpha = 80$ has a significantly *worse* SAD score than *ImpA* $\alpha = 80$, (9.32a% vs. 6.58%), yet outperforms it in WER (14.6% vs 15.0%). This is yet more evidence to support the assertion that SAD is an ill-defined problem and a reminder that this kind of SAD metric only scores the relative similarity with one specific reference transcription, which itself may not be optimal.

It is also worth noting that, while the overall speech segmentation error is better, the false alarm rate goes up for the smoothed segmentations (compared with shout). For the downstream ASR task this poses a potentially less significant issue than if it were missed speech – missed speech is completely non-recoverable, whereas false alarms will not necessarily produce word insertions. Indeed, it may be beneficial to explicitly tune the smoothing algorithm to favour lowering missed speech error at the expense of false alarms for the ASR task.

In order to fully control for the dependence MT has on the WER of the ASR transcript it receives, we have shown the results for when we map each of our segmentations to the force-aligned gold transcription in Table 5.1. First of all, from the variation in performance we can infer that segmentation does indeed have a direct effect on MT performance. However, in these conditions we find that the MT system favours the break smoothing algorithm with shorter segments. Figure 5.5 shows how the prior and posterior distributions compare. We can see that when $\alpha = 30$ the distribution takes a closer shape to the true distribution with a 'shift' to shorter segments, which could be

	SAD			ASR	MT:ASR	MT:Gold
Segmentation	E_{spch}^{miss}	E_{spch}^{fa}	E_{spch}	WER	BLEU	BLEU
Manual	-	-	-	13.6	0.2472	-
SHOUT	12.71	0.16	12.87	18.3	0.1967	0.2256
Simple	9.91	2.66	12.57	16.7	0.2007	0.2319
Uniform 300s	0.00	6.49	6.49	14.8	0.2014	0.2369
ImpA $\alpha = 30$	4.25	2.33	6.58	15.0	0.2085	0.2409
ImpA $\alpha = 80$	7.86	1.46	9.32	14.6	0.2104	0.2368
ImpB	5.31	1.64	6.95	14.3	0.2127	-

Table 5.1: *Segmentation, ASR and MT results for each segmenter. MT results are shown for both ASR output and gold transcripts segmented with different segmentation models.*

due to the fact that the automated methods have more accurate segment boundaries. As such the MT system in this case could be benefiting from more 'sentence-like' utterances, whereas the ASR system can actually afford to have, and may actually benefit from, slightly longer segments as it is able to further segment in more detail using its own decoder.

5.5 Speaker Diarization - Experiments

We turned our attention back to the NIST RT meetings data in order to experiment with what we learned from our new segmentation techniques. There are two main motivations for experimenting with this data:-

Firstly, as we mentioned in Section 5.2.2, we would like to know how robust our utterance-break prior technique is to different domains. The spontaneous group conversation of the NIST RT data contrasts the prepared and rehearsed monologue of TED talks, but is functionally identical to the AMI data. We therefore decided to look at the effect of applying the utterance-break smoothing technique to the NIST RT data with each of the priors.

Secondly, we wanted to investigate the effect of different segmentations on speaker diarization performance. In Chapter 4 we showed that if we are able to select purer speech segments then we can estimate better speaker models. Our utterance-break

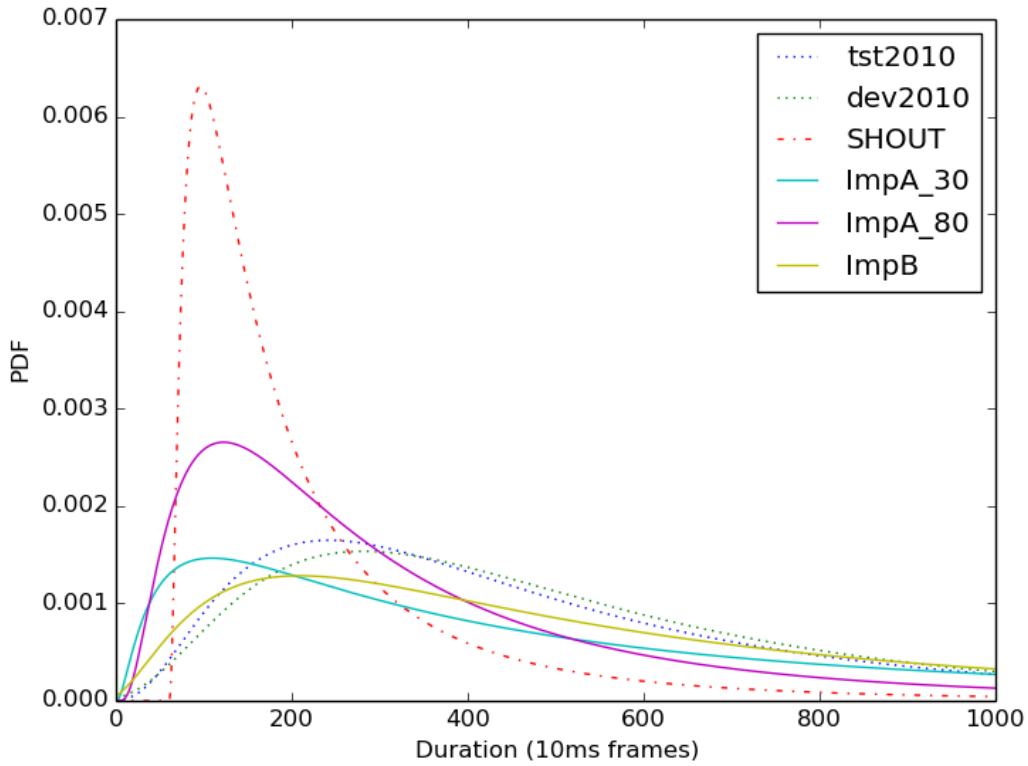


Figure 5.5: *A comparison of the prior and posterior segment length log-norm distributions (fitted to the segmentation data in each case). Note that the best performing system* ImpB *also has a distribution that is closest to the prior.*

prior smoothing was shown to produce segmentations that are closer in nature to manual segmentations. However, by smoothing over non-speech breaks the resultant segments contain information that is not useful for speaker discrimination such as audible non-speech noise and silence. This can have the effect of introducing statistical noise to the clustering process which can detriment performance.

For the purposes of speaker clustering it could therefore be better to use a purely acoustic segmentation that has a high frame-level precision, such as what we use to create the break candidate sequence. This would help to ensure that the frames being passed to speaker diarization have a higher 'speech purity'.

5.5.1 Speech Segmentation Systems

An acoustic model was trained on AMI data using the same setup as described in Section 5.3.2.3. We ran the simple method as before to represent an example of a state-of-the-art acoustic speech segmentation method.

Then we used implementation B (ImpB) of the smoothing algorithm with both the AMI prior and the TED prior. For the smoothed versions, we started with a sensitive acoustic segmentation (SensAco) that allowed a minimum duration of only 100ms for both speech and non-speech segments.

We also considered the SAD results for the QIO-Aurora Toolkit (see Section 3.1.3.2) as this was the segmenter used by our first speaker diarization system for our oracle experiments (see Chapter 4).

5.5.2 Speaker Diarization System

The speaker diarization system is identical to the baseline system shown in Section 3.3.3. We use the Kaldi toolkit for feature extraction – note that this differs slightly from the HTK-based system used in Chapter 4.

5.6 Speaker Diarization - Results

First, we looked at the effect of our uterance-break prior smoothing on speech segmentation alone. The results are shown in Table 5.2.

We find that the smoothed segmentations improve performance significantly over both the *QIO-Aurora* and the *Simple* results. However, while we expected the AMI prior to perform better than the TED prior, we find that their results are very similar.

Segmentation	E_{spch}^{miss}	E_{spch}^{fa}	E_{spch}
Qio-Aurora	5.09	2.11	7.20
Simple	6.02	1.30	7.32
ImpB (AMI Prior)	0.85	3.65	4.49
ImpB (TED Prior)	0.85	3.58	4.43

Table 5.2: *Speech segmentation error (%)results for NIST RT06/07/09.*

This suggests that the overall gain from the smoothing is not particularly sensitive to the specific choice of prior. Indeed, the difference in the respective distributions of each prior may not be significant in terms of deciding which break candidates to smooth – they may be mostly informing the same decisions.

Another possible reason for this is that we used the same acoustic scaling factor but potentially have quite different acoustic conditions and subsequent frame-level likelihoods. In particular, the TED scenarios contain much more varied non-speech noise conditions and events than the AMI/NIST RT scenarios. It could therefore be the case that the ratio of influence of the prior to the acoustic information when decoding the break sequence could be lower in some instances because the disparity between the speech and non-speech classes is much greater.

Next, we looked at how the different segmentation methods affect speaker diarization. For these experiments we used the TED prior for smoothing as it showed slightly better speech segmentation results. We also included speaker diarization results when we use the sensitive acoustic speech segmentation (SensAco).

We use *SensAco* directly but also combine it with *Oracle* and *ImpB (TED prior)* segmentations. The combination is done by first using *SensAco* to perform the speaker clustering, then introducing the other segmentation for a final re-segmentation with the final speaker models after clustering has terminated.

The results are shown in Table 5.3.

The first observation to note is that all of our systems show an improvement in DER when compared to the QIO-aurora system. This shows that our methods are competitive with other state-of-the-art methods.

If we then look at our sensitive acoustic speech segmentation *SensAco* compared with the smoothed segmentation *ImpB*, we see that *SensAco* actually has a better DER (28.19% vs. 29.20%). This is despite the fact that *SensAco*, expectedly, has higher E_{spch} error (7.62% vs. 4.70%). This is due to all the missed 'speech' when compared

to the reference, which in all likelihood is not strictly acoustically speech at a frame-by-frame level. The reason the DER is still better, even with the large hit taken to E_{spch}^{miss}, is that E_{spkr} is much lower (14.72% vs. 18.75%). This suggests that the speaker clustering has worked better and, consequently, the resultant speaker clusters are purer.

The final row in the table *SensAco->ImpB* shows what happens when we use the 'purer' speaker models garnered from *SensAco* to perform a final speaker re-segmentation over the *ImpB* speech segmentation. This allows us to restore some of the lost E_{spch}^{miss} by gaining the better E_{spch} of *ImpB*. Of course, we also introduce some error to E_{spkr} as we cannot guarantee all the previously missed speech will be assigned the correct speaker, however it is still lower than for *ImpB* alone (16.05% vs. 18.75%). Overall this results in a further substantial improvement to DER (26.49% vs. 28.19%) as we are able to get the 'best of both worlds' – purer speaker models and a segmentation that is closer to the reference.

We also tried combining *SensAco* with our oracle speech segmentation *SensAco->Oracle*. This time we see a less significant improvement to DER (21.13% vs. 21.31%). This is likely due to the fact that, although some segments will contain non-speech, E_{spkr}^{fa} is zero so there is less statistical noise from extra 'unwanted' segments. This supports the assertion that there can exist some speech segmentation that is actually better for speaker diarization than the reference segmentation.

5.7 Conclusion

We have shown that speech segmentation can be improved by exploiting non-acoustic prior knowledge – in this case, the use of an utterance-break model. Such improvements have been shown to propagate through larger system chains to further benefit the performance of downstream tasks such as automatic speech recognition (ASR), machine translation (MT) and speaker diarization.

We have further shown that the benefits to MT are not simply a consequence of the benefits to ASR, suggesting that speech translation performance is highly dependent on the quality of the speech segmentation.

For speaker diarization we were able to substantially reduce diarization error rate by combining our new method with purely acoustic speech segmentation techniques in order to improve segmentation and speaker clustering.

This leads us to assert that the optimal speech segmentations for each task are not necessarily the same – furthermore, typical speech segmentation evaluation metrics are

not a reliable indicator of downstream system performance.

Given what we have learned from this investigation we believe there is scope in future work to add linguistic knowledge into the segmentation model, such as language modelling scores and even syntactic bracketing information. This would require running segmentation as an iterative procedure, on the output of an ASR model, before feeding it back in as the input to an ASR system.

Indeed, by treating the problem in the semi-Markov manner which we have presented, it could be possible to *inject* any additional information into the decoding process for scoring break candidate likelihoods. As such we could include scores from a suite of acoustic models (such as those for explicit audible non-speech e.g. music), linguistic information (if available) and any other features that would seem suitable.

Segmentation	E_{spch}^{miss}	E_{spch}^{fa}	E_{spch}	E_{spkr}^{miss}	E_{spkr}^{fa}	E_{spkr}	DER(%)	E_{clst}	$\|E_{clst}\|$
Oracle	0.00	0.00	0.00	5.04	0.00	16.28	21.31	-0.63	1.21
SensAco->Oracle	0.00	0.00	0.00	5.03	0.00	16.10	21.13	-0.71	1.29
QIO-aurora	5.09	2.11	7.20	14.85	2.34	14.26	31.47	1.54	2.21
ImpB (TED prior)	0.86	3.84	4.70	5.97	4.48	18.75	29.20	-0.58	1.33
SensAco	6.65	0.97	7.62	12.35	1.12	14.72	28.19	-0.67	1.25
SensAco->ImpB (TED prior)	0.86	3.84	4.70	5.97	4.50	16.05	26.49	-0.71	1.29

Table 5.3: Comparing the effect of different speech segmentation methods on speaker diarization results for NIST RT(06/07/09). The symbol ->indicates that the first segmentation is used for diarization during clustering and the second segmentation is used for a re-segmentation pass with the final speaker models after clustering has terminated.

Chapter 6

A Punctuation System Combining Segmentation, Acoustic and Machine Translation Models

6.1 Introduction

One of the most significant differences between written and spoken examples of a given language is the adherence to the inherent syntactic rules. Spontaneous speech in particular does not typically follow these rules as we do not structure natural discourse in the same way as we do in written language. This means that concepts such as sentences become much more difficult to define. For languages such as English, punctuation provides the anchor points in the written form. It controls the pace of reading and helps to form the syntactic structure of the text. In speech, we rely on speaking rate, pauses and prosody among other cues to provide this sense of structure.

The work we have presented in previous chapters has shown how we can improve automatic speech segmentation and closer emulate the natural segmentation of human annotators. We can consider this task to be that of exposing the inherent structure in speech mentioned above and that there is a relationship between punctuation and speech segmentation. We hypothesise, therefore, that it may be appropriate to exploit acoustic evidence in order to correlate such information with punctuation, allowing us to effectively *punctuate* speech. This task is known as Punctuation Restoration.

The implications of such a task would allow for a richer transcription of speech by providing another layer of information to be represented by the resultant text i.e. some of the intended expression from the speaker could be captured in the punctuation. This also make the transcription easier to read and search. However, the potential benefits are much wider, as adding punctuation allows transcribed speech to propagate onwards to further Natural Language Processing (NLP) tasks. An example of this is Machine Translation (MT), whereby current systems are typically trained on corpora of parallel sentences between source and target languages. If we are not able to structure speech into sentence-like units we may not be able to use such systems effectively. We may also benefit from punctuation when transcribing speech for subtitling, where it may be better to present text sentence-by-sentence as opposed to simply a time-based window.

Most existing research approaches this problem as a post-processing task whereby punctuation is restored to the raw transcription output from an ASR system by methods which only exploit linguistic information. Many of these systems exploit a language model trained on punctuated text and use the resultant posterior scores when scoring the unpunctuated transcription to insert punctuation where it would have been most likely (Gravano et al., 2009; Hassan et al., 2007). Increasingly however, we are seeing methods which treat the problem as an MT task (Cho et al., 2012; Peitz et al., 2011;

Ueffing et al., 2013). This concept treats unpunctuated and punctuated texts as if they were different languages and applies standard MT methods to *translate* from one to the other.

In this work, we present an acoustic method based on the segmenter introduced in Chapter 5, as well as an MT approach similar to current state-of-the-art methods. We compare their individual performance on a punctuation restoration task and suggest ways in which they could be combined in order to complement each other.

6.2 Punctuation Restoration - Task Definition

We can consider the task of Punctuation Restoration to be that of adding punctuation tokens to an unpunctuated word sequence, W, resulting in a punctuated word sequence, η. This process can be informed by several different factors. One of the main factors is linguistic information and can be represented by a language model trained on punctuated text. However, another significant factor can also be acoustic information, such as pause durations, pitch inflections, energy dynamics, etc. By considering punctuation restoration in this way, it is related to the task of speech recognition and is analogous in many respects.

The task of speech recognition – converting speech to text – is often defined as follows:-

$$\hat{W} = \underset{W \in \mathscr{L}}{\arg\max} P(O|W)P(W) \tag{6.1}$$

Here $\hat{W}$ is an optimal word sequence from the domain $W \in \mathscr{L}$, where $\mathscr{L}$ is some language vocabulary of words and O is acoustic observations – e.g. frames of MFCCs. The two probabilities, $P(O|W)$ and $P(W)$, are informed by acoustic and language models respectively. The optimal word sequence is therefore that which maximises the product of these probabilities. Typically, speech recognition systems do not output punctuation, only sequences of word tokens. This is because, in normal speech, punctuation is not explicitly spoken.

We can consider punctuation marks (periods, commas, etc.) to be word tokens, just like any other word. These tokens can be put into a punctuation vocabulary, C. The task of Punctuation Restoration can then be thought of as inserting punctuation tokens between the existing word tokens in an un-punctuated word sequence from a speech recognition system. This can be thought of as an injective function, f, from the

unpunctuated word sequence, $\hat{W}$, to a new punctuated word sequence η:-

$$f : \hat{W} \in \mathscr{L} \rightarrowtail \eta \in \mathscr{L} \cup C \tag{6.2}$$

In practice, we can impose further constraints:-

- the relative positions of the words in $\hat{W}$ should be preserved in η – i.e. no re-ordering or deletions.
- there should be maximum (typically 1) number of punctuation tokens inserted between words.

If we have punctuated text data, we could train a punctuated language model in order to find the probability of any given punctuation sequence $P(\eta)$. The optimal punctuated word sequence $\hat{\eta}$ could then be defined as follows:-

$$\hat{\eta} = \arg\max_{\eta} P(\eta) \tag{6.3}$$

However, this method only exploits linguistic information. We would like to know if it is possible to use acoustic information to influence the decision about the optimal punctuation. In particular, we hypothesise that the temporal gaps between words could be exploited. For example, certain punctuation marks, such as periods and commas, are correlated with pauses in speech. We could then re-define the optimal punctuated word sequence $\hat{\eta}$ as:-

$$\hat{\eta} = \arg\max_{\eta} P(A|\eta)P(\eta) \tag{6.4}$$

Where A represents any relevant acoustic observation. In the case where the only acoustic information we consider is the duration between words then $A = D = \{d_0, d_1, \dots\}$ and d_i represents the duration between the end of the word at position $i-1$ and the word at position i. We can then consider any punctuated word sequence based on its linguistic and acoustic likelihood. There are many different ways we could actually model $P(D|\eta)$, some of which will be discussed in throughout this chapter.

6.3 MT Method

We trained a phrase-based Statistical Machine Translation (SMT) model using the MOSES toolkit (Koehn et al., 2007). The training corpus was taken from the English side of the French-English IWSLT 2013 shared task. We then created a parallel

Parallel Corpora	MT	ASR
TED(In Domain)	3.7	3.2
Europarl v7	54.1	48.8
News Commentary v7	3.4	3.1
Common Crawl	79.5	69.9

Table 6.1: Word counts (in millions) for corpora used to train the punctuation model (MT) and the ASR language model (ASR). The difference in the counts is mainly due to the extra punctuation tokens for the punctuation model.

corpus by formatting the manually punctuated reference transcription text to remove punctuation, this can essentially be considered an approximation of the ASR output. Table 6.1 describes the size of the different corpora used for training the translation model and the language model.

SMT systems typically expect training and testing examples that presented as sentences. However, ASR output does not inherently have this structure. Speech, particularly when it is spontaneous and unprepared, does not necessarily conform to the syntactic rules of written language making the concept of a *sentence* more vague in this domain. Aside from this, even if we were presented with sentence-structured speech, the ASR system outputs text strings corresponding to what occurred in the segmentation it was given. If there is no correspondence between segments and the underlying sentence structure then the output will not be guaranteed to be *sentences*.

In order to accommodate for the difference between ASR output and written language, we needed to deviate from conventional SMT language model training. We concatenated the punctuated English sentences into very long strings comprising 20 sentences on each line. Each line of the parallel corpus would therefore have 20 matching sentences with and without punctuation. This effectively emulates the ASR output more closely. Tuning was done on the IWSLT 2010 development set, where lines also comprised 20 sentences.

For the translation model we simulated the lack of input segmentation by translating a sliding window of text with segmentations removed. Thus all punctuation is the result of lexical signals, and not the result of any end of line input, except for the final end of line in the test set. The translation model output a 1000-best list. Punctuation in the translation output was divided into two types, end of sentence markers and end of clause markers. The typical end of sentence marker is a full stop, and the typical end

Punctuation Class	Punctuation Token
End sentence (period)	. ! ?
End clause (comma)	, - ” : ; –

Table 6.2: Mapping between reference/training punctuation tokens and our two levels of segmentation class.

of clause marker is a comma. The mapping is shown in Table 6.2.

6.4 Acoustic Method

For the acoustic model we started with the same segmentation system as presented in Chapter 5 and worked with the same IWSLT dataset(Cettolo et al., 2012). We used the best performing implementation of the system (ImpB – see Section 5.2.3) to produce a segmentation that was then passed to the ASR system, which again, was the same as presented in Chapter 5. We were then able to do a phone-alignment with the resultant output which allowed us to discover if the ASR system hypothesised any gaps between words – i.e. if the recogniser passed through the silence state between words.

We then aligned the ASR output from the development set with the manually punctuated reference using the same two-level segmentation mapping as shown in Table 6.2. While there would clearly be some mis-match due to word errors, this allows us to have a reasonable approximation of where punctuation *should* have been in our ASR hypothesis.

At this point, we now have the development ASR transcript where we know: the words; the duration of the gap (if any) between adjacent words; where punctuation should have occurred; and where the SAD system hypothesised a break. This allowed us to find the distribution of each punctuation mark with respect to the duration of the gap between words $f_{period}(d)$ and $f_{comma}(d)$, as well as that of no punctuation occurring $f_{none}(d)$. We decided to model this differently for 2 different cases: a discrete distribution when there is no gap between words and a continuous distribution for when there is.

The probability P for each punctuation case $c \in C = \{period, comma, nopunc\}$ if

c	$P_{nosil}(c)$	$prior_{sil}(c)$
period	0.0175	0.1839
comma	0.0435	0.1514
nopunc	0.9389	0.6646

Table 6.3: Discrete probabilities for each punctuation case when there is no gap between words $P_{nosil}(c)$ compared with the prior distribution of punctuation when there is a gap $prior_{sil}(c)$.

there is no gap between words is therefore:-

$$\begin{aligned} N_{nosil} &= \#period_{nosil} + \#comma_{nosil} + \#nopunc_{nosil} \\ P_{nosil}(c) &= \frac{\#c_{nosil}}{N_{nosil}} \end{aligned} \tag{6.5}$$

And for the case where there is a gap between words:-

$$\begin{aligned} N_{sil} &= \#period_{sil} + \#comma_{sil} + \#nopunc_{sil} \\ prior_{sil}(c) &= \frac{\#c_{sil}}{N_{sil}} \\ f(c,w_i) &= prior_{sil}(c) \cdot \mathcal{N}(d(w_i), \theta_c) \\ P_{sil}(c,w_i) &= \frac{f(c,w_i)}{\sum_{u \in C} f(u,w_i)} \end{aligned} \tag{6.6}$$

Whereby $d(w_i) = w_{(i+1,start)} - w_{(i,end)}$ is the duration in frames between the current word w_i and the next w_{i+1}, and *theta* is the parameter set that describes the normal distribution for each punctuation case. We get the probability $P_{sil}(c)$ by normalising $f(c)$ over all punctuation cases.

For the IWSLT dev set we found the discrete probabilities to be as shown in Table 6.3. Here we see that, as is to be expected, it is significantly more probable that there will be no punctuation after any given word, with the probability of commas slightly outweighing that of periods.

For the case when there is a gap between words, we found the punctuation cases were distributed as shown in Figure 6.1 with the prior probabilities as shown in Table 6.3. The priors tell us that if there is a gap between words then it becomes much more likely that we should insert punctuation as compared with when there is no gap. From the distributions, we can see that as the duration between words increases we move from no punctuation being most likely to commas, then periods. We can also see that

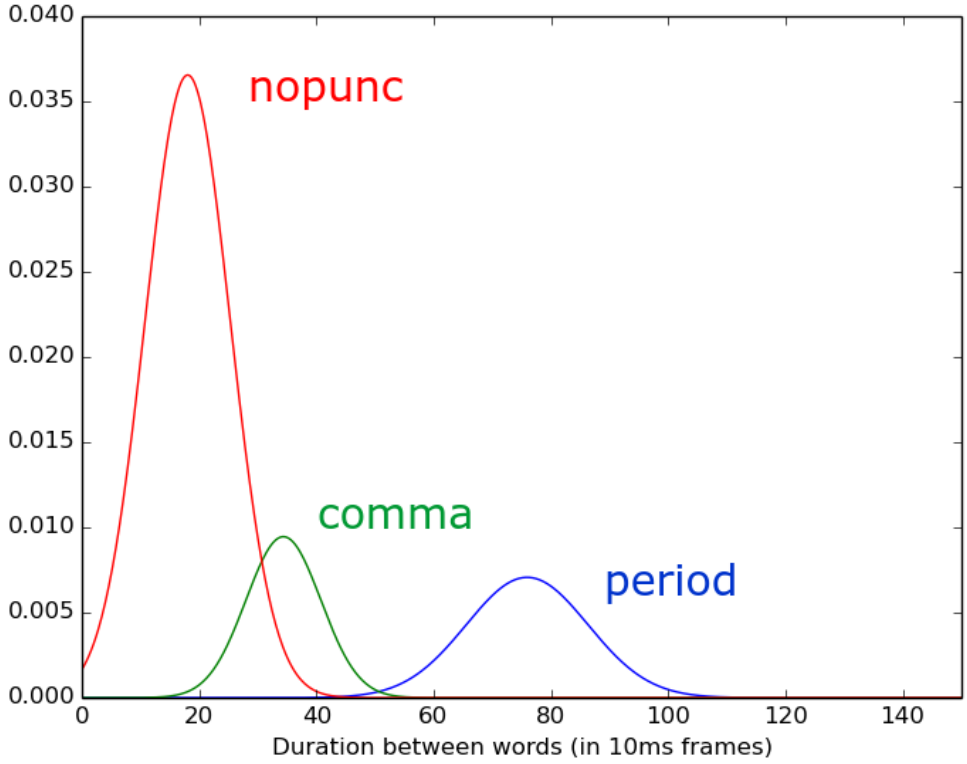


Figure 6.1: *Normal distribution of punctuation cases as a function of the duration between words. Each distribution has also been scaled by its respective prior probability.*

as the duration tends to infinity, P_{period} will tend to one. This is the behaviour we want, as given a very long pause, we should always insert a period.

For any given word we can then decide whether or not to insert punctuation after it, based on Equation 6.7.

$$punc(w_i) = \begin{cases} \arg\max_{c \in C}(P_{nosil}(c)) & \forall d(w_i) = 0 \\ \arg\max_{c \in C}(P_{sil}(c)) & \forall d(w_i) > 0 \end{cases} \tag{6.7}$$

It is, of course, a naïve solution to use only acoustic information when linguistic information is available. However, these methods allow us to investigate how much information, if any, we can garner from acoustics alone with regards to punctuation restoration.

6.5 Experiments

We prepared 3 versions of reference transcription for the IWSLT tst2010 dataset: without punctuation; with commas and periods mapped according to Table 6.2; and with all punctuation merged to 1 class *PUNC*. We were then able to score our systems based on Word Error Rate (WER) as well as the percentage of punctuation tokens that were correct.

6.5.1 Systems

Acoustic Method A (AM-A): SAD

Our simplest method for restoring punctuation using acoustics involves taking our original SAD segmentation and simply adding a period at the end of each segment. This method, of course, is unable to insert commas as it is unable to discriminate between periods and commas.

Acoustic Method B (AM-B): Gap Distributions

Our next method involves inserting punctuation according to the method introduced in 6.4. This will essentially make local decisions on whether or not to introduce punctuation after each word according to the learned distributions for each punctuation class given the gap between the given word and the following one.

MT Method (MT)

For the MT method we take the 1-best output from the translation system. This method is completely agnostic to all acoustic information and segmentation, and uses only the linguistic information that can be garnered from the ASR transcript.

Combined Methods

We also explored a naive system combination by taking the union of the MT method with each of the acoustic methods in turn. Here we would insert punctuation if it featured in either system hypothesis. In the case of a disagreement between systems over whether to insert a comma or a period, we considered what would happen if favouring either one over the other.

6.6 Results

The results for precision, recall and F-score for each system are shown in Table 6.4 and the corresponding WER scores are shown in Table 6.5. Here we can see that by introducing punctuation to the reference our WER jumps from 14.3% to 24.2%, meaning that the inclusion of punctuation tokens can have a significant impact if a system is unable to hypothesise anything for it.

Overall, we see that the MT method outperforms both of the acoustic methods. This is hardly surprising as we would expect there to be a limitation as to how much the acoustic evidence alone will correlate with punctuation. However, it is worth noting that the precision is quite high – greater than 70% – for both acoustic methods when punctuation is merged (rows **AM-A** and **AM-B**. This suggests that an acoustic gap between words is in fact a strong indicator that punctuation should be inserted. However, it remains to be seen with further experimentation whether or not this can complement a linguistic-based approach such as our MT method.

For the two systems that are able to insert commas, *AM-B* and *MT*, there is a significant gain when punctuation tokens are merged. This suggests that one of the biggest challenges is actually to discriminate between commas and periods. Indeed, the very subjective nature of comma usage may be harder to evaluate without multiple human transcriptions.

The combined systems were unable to outperform the MT system overall. However, in some examples they were able to produce a better F-score for one of the punctuation cases, such as periods for *MT+AM-A** and commas for *MT*+AM-A*. This is, of course, a simple naive system combination but it does offer some insight into the potential for combining the systems in other ways.

6.7 Conclusion and Future Work

We have established a baseline from which to continue experimentation and research on the task of punctuation restoration. We have shown that acoustic information can be used to predict punctuation, but we need further analysis to determine how significant this can be when contrasted with state-of-the-art linguistic approaches such as the MT method we have presented. We would therefore need to investigate methods for combining both the MT method and acoustic information. This may include combining acoustic scores into the MT model during alignment. An alternative may be

System	Period			Comma			Merge		
	Prec.	Recall.	F-score	Prec.	Recall	F-score	Prec.	Recall	F-score
Raw ASR	0	0	0	0	0	0	0	0	0
AM-A	0.5466	0.3816	0.4492	0	0	0	0.7954	0.2407	0.3694
AM-B	0.5058	0.4307	0.4650	0.2970	0.1209	0.1718	0.7088	0.4202	0.5274
MT	0.6032	0.4695	0.5280	0.4651	0.4132	0.4374	0.7532	0.6320	0.6872
MT*+AM-A	0.4776	0.5129	0.4944	0.4651	0.4155	0.4386	0.6895	0.6593	0.6738
MT*+AM-B	0.4497	0.5256	0.4844	0.3926	0.4456	0.4172	0.6162	0.6963	0.6536
MT+AM-A*	0.4807	0.6072	0.5364	0.4789	0.3563	0.4084	0.6895	0.6593	0.6738
MT+AM-B*	0.4333	0.5576	0.4876	0.3767	0.3925	0.3842	0.6162	0.6963	0.6536

Table 6.4: *Precision, recall and F-score measures for each system considering periods, commas and when all punctuation is merged in both hypothesis and reference. For the combined systems, the asterisk denotes the system hypothesis that was favoured if there was a disagreement between punctuation tokens at the same position.*

System	None	All	Merge
Raw ASR	14.3	24.2	-
AM-A	-	23.1	23.1
AM-B	-	23.9	22.7
MT	-	22.9	20.4
MT*+AM-A	-	23.1	21.4
MT*+AM-B	-	24.9	23.1
MT+AM-A*	-	23	21.4
MT+AM-B*	-	25.1	23.1

Table 6.5: *WER for each system considering the reference with no punctuation, all punctuation (period and comma), and when all punctuation is merged in both hypothesis and reference. For the combined systems, the asterisk denotes the system hypothesis that was favoured if there was a disagreement between punctuation tokens at the same position.*

simply to re-score the MT systems N-best list based on the acoustic likelihood of a given sequence.

Chapter 7

Deep Neural Net Speech Segmentation

7.1 Introduction

In Chapter 5 we presented a novel method for exploiting non-acoustic prior knowledge to improve speaker segmentation. One of the key assumptions that allow this method to work is that we have a reliable frame-level acoustic segmentation to begin with.

We found that conventional GMMHMM-based acoustic speech segmentation was adequate for scenarios, such as in a meeting room or a well recorded presentation, where there are not a lot of challenging noise conditions. However, when we tried working with more difficult scenarios such as television broadcasts we found that the acoustic models were less reliable and we could no longer justify the assumption that they have a high frame-level precision. This is likely due to the demanding noise conditions that are present across this kind of data: music, background noise, variable channel quality, applause, laughter, etc. In order to try and improve our acoustic models we decided to look at applying Artificial Neural Networks (ANNs) and Deep Neural Network (DNN) techniques. We hypothesise that such techniques may allow us to improve the robustness of the acoustic modelling aspect of our speech segmentation system but may also help to improve or incorporate the smoothing that is also required.

Artificial Neural Networks (ANNs) are machine learning computational models that take influence from complex neurological systems such as the human brain. The fundamental building-block of all ANNs is the *neuron*. A neuron takes multiple inputs and outputs some function of them. Such functions are normally very simple parametrised mathematical operations, the parameters (weights) of which can be learned from data. In isolation, a neuron has limited computational potential. However, when numerous neurons are inter-connected together as nodes in a larger network, more complex and powerful systems can be realised.

It is the number of neurons as well as the number of connections between them in a given network that governs how complex a problem it can potentially solve. The difficulty with increasing the number of neurons, and in particular the number of connections, is training – the data and computational time required can soon become impractical on conventional computer hardware.

With the advent of affordable commodity Graphics Processing Units (GPUs) in recent years, this difficulty has been somewhat relaxed. GPUs are explicitly designed to process data in a highly parallel fashion, with modern system examples comprising several thousand specialised processing ‘cores’. This makes them ideal for many scientific applications involving large complex datasets with parallelisable processing

tasks (Luebke, 2008).

These recent hardware advancements have meant that many ANN inspired concepts, that may have previously only been possible to discuss in theory, can now feasibly be introduced as practical solutions to many real-world problems. Perhaps the most prolific and substantive example in the speech recognition community is the use of Deep Neural Networks (DNNs) (Hinton et al., 2012). DNNs are feed-forward ANNs, with more than one hidden layer between the input and output layers. The number of hidden layers determines the 'depth' of the network – the longest path from input to output. In the conventional setup, each layer strictly only feeds-forward to the next, with no other connections possible.

This architecture allows for rich networks to be realised while still allowing for tractable and efficient layer-wise training algorithms (Larochelle et al., 2009; Glorot and Bengio, 2010). In this way, DNNs can be considered as a sequence of non-linear transforms of the input. This can be exploited to produce posterior likelihoods for classification or even enhanced feature transformations through the use of bottle-necks (Yu and Seltzer, 2011; Gehring et al., 2013).

7.2 Motivation

The Multi-Genre Broadcast (MGB) challenge is a new evaluation campaign involving tasks in speech recognition, speaker diarization and lightly supervised alignment. The data comprises audio from television shows broadcast by the BBC over a diverse range of genres and scenarios. Such diversity inherently presents a large variety of noise conditions which places increased demand for the robustness of any assosciated acoustic model.

It is known that DNNs can be highly noise robust compared with conventional GMM-based systems for speech recognition tasks (Seltzer et al., 2013). We wanted to see if this same behaviour could be exploited for the speech segmentation task, i.e. robustly detecting speech segments in noise.

Another advantage of DNNs is that they are inherently very good at modelling a wide temporal context. We can do this by 'splicing' many frames of context to the left and right of any given central frame. For a conventional GMM, this would greatly increase the number of required parameters in the model, and subsequently the amount of data required to train such a model. With DNNs this is less of an issue as any under-represented or noisy feature dimensions will simply be weighted down during training,

effectively 'ignoring' them.

By modelling a greater context like this, the DNN may be able to learn some of the local temporal structure of speech.

7.3 Feature Choice

In order to be consistent with previous experiments we chose to use PLP features for both the GMM and the DNN based systems. For the GMM system we used 12 PLP coefficients plus energy, plus delta and delta-delta features, totalling a feature vector of dimensionality 39. For the DNN system we used the same 12 PLP coefficients plus energy, but instead of using dynamic features we spliced 40 frames of context either side of each frame, totalling an input vector of dimensionality $81 \times 13 = 1053$.

Such a large vector would be nearly impossible to model adequately with a GMM for this task, however this is relatively trivial for a DNN. The advantage of the DNN is that it is able to *explore* this larger feature space in order to find a more speaker discriminative sub-space that would be difficult to find with conventional expectation maximisation learning of GMMs.

We could also have considered using filter-bank coefficients directly. This may allow the DNN to discover any speaker discrimination information that may be lost during PLP parametrization. However this would make the comparison between DNN and GMM more difficult as diagonal covariance matrices are not able to account for the high correlation across the filter-bank.

7.4 Network Architecture

Choosing the right network architecture can be a difficult first step when applying DNNs to a given problem. Often the optimal number of layers, as well as the number of nodes per layer, can only be determined heuristically as a trade-off between performance and computational tractability.

There are many examples of different DNN architectures for ASR (Dahl et al., 2012; Graves et al., 2013) which usually prescribe 5 or more layers with around 2000 nodes per layer. However, there are relatively few published examples for speech segmentation (Ryant et al., 2013). This means there is a lack of an established standard for the speech segmentation task among the research community.

We could use the ASR architecture as a template, but the tasks are also quite different in topology. For ASR we typically start with a relatively small input vector (tens of features) which is ultimately transformed into a much larger output vector (thousands of HMM states). For speech segmentation we have a larger input vector (hundreds of dimensions, due to the greater contextual frame-splicing) and require only a few dimensions in the output vector, nominally one for speech and one for non-speech. It may not be pertinent therefore to assume that there is any relationship between the optimal DNN architectures for each task.

We decided to try a few combinations of different numbers of hidden layers and nodes per layer. We used the frame-level cross-validation error on a held-out 10% subset of the training data as a performance metric. The results are shown in Table 7.1.

The first thing to notice is that we do not gain anything by increasing the number of hidden layers from 3 to 4 as there is no change in cross validation error for the same number of nodes per layer in each case. This suggests that the fourth layer is largely redundant. The number of nodes per layer does seem to matter as we get a small but significant gain by increasing from 1024 to 2048 nodes per layer (23.9% CV error to 23.5% in both cases).

Table 7.1: Comparing different DNN architectures with the MGB training data.

# Hidden Layers	# Nodes p/Layer	CV Error(%)
3	1024	23.9
3	2048	23.5
4	1024	23.9
4	2048	23.5

7.5 Experiments

In order to compare our GMM-based speech segmentation models with DNN-based models we trained several of each on the MGB challenge training data.

The training data set comprises several thousand hours of audio from television programmes. These recordings are lightly supervised by automatically force aligning manually produced subtitles with the audio. This can be unreliable for several reasons:-

- The subtitles do not represent the speech in a strictly verbatim manner.

- There are many examples of speech in the audio that do not have associated subtitles (such as during the announcements before and after a show).
- While care was taken to parse the subtitles, there are examples of words that are not actually spoken (such as credits and onomatopoeia).
- The automatic forced-alignment is not perfect and introduces its own errors.

All of these reasons mean that the supervision contains a significant amount of corruption. However, as this is common to both types of model it is in some sense controlled for the purposes of experimentation.

From the full training set, we selected 100 hours. In order to maximise the diversity of the set, each programme was represented by a maximum of one episode per parent series.

We selected the most promising models for each type, GMM and DNN, based on a cross-validation set of roughly 10% of the training set. The GMM model had 100 Gaussian mixture components for both speech and non-speech classes. The DNN model was identical to that which appears in Table 7.1 with 3 hidden layers, each comprising 2048 nodes.

We applied our utterance-break prior smoothing to both methods (see Chapter 5) using the TED prior – we found this to work well generically across many domains. We produced the initial acoustic-only segmentation in slightly different ways. For the GMM system we used the standard GMM-HMM decoding technique with a minimum duration of 100ms. For the DNN system we simply used the output from a forward-pass decode. As the DNN has a wide context input vector, it is able to do a kind of localized smoothing so does not require any technique to introduce a minimum duration constraint.

7.6 Results

We compare the frame-level speech segmentation error, $E_{spch}^{miss} + E_{spch}^{fa} = E_{spch}$, as well as the resultant Word Error Rate (WER) from the ASR system across the MGB development dataset. The results are shown in Table 7.2 along with the WER when the manual segmentation is used.

Here, we see that the DNN system performs better than the GMM-based system on all metrics. We observe a reduction in E_{spch} (11.8% vs. 15.6%) which comprises

Table 7.2: Comparing the effect of different segmentations on the ASR results of the MGB dev. set.

Segmentation	$E_{spch}^{miss}(\%)$	$E_{spch}^{fa}(\%)$	$E_{spch}(\%)$	WER(%)
manual	0.0	0.0	0.0	41.8
GMM	5.0	10.6	15.6	44.3
DNN	2.6	9.2	11.8	43.4

a reduction in both of its components E_{spch}^{fa} and E_{spch}^{miss}. The speech segmentation improvement propagates onwards to a reduction in WER from 44.3% to 43.4% for the downstream ASR system component. However, this is not in proportion with the improvement to E_{spch}^{miss}. This suggests that not all recovered speech is necessarily transcribed correctly and at some point improving the speech segmentation will encounter the limitations of the ASR system.

7.7 Improving Training Data

In order to close the gap between the DNN system and the manual segmentation we would likely need to improve the quality of the training data. One possible way to do this would be to use only the segments of audio that presented a high confidence score from the automatic aligner. Here, we would be able to make a stronger assumption that the speech segments contain mostly speech. However, we would still not have any guarantee that non-speech segments do not contain speech.

If we instead look at the phone-level alignments and consider the ASR model's 'silence' and 'noise' phones to be non-speech, we could produce a relatively reliable speech/non-speech segmentation *within* a speech segment.

There are two main issues relating this method:-

Firstly, the amount of reliable non-speech we can harvest in this way may not be enough. Additionally it will only be representative of non-speech in close proximity to speech, which may not fully capture the variability of noise – particularly as many programmes will have been produced to increase the signal-to-noise ratio of speech when it is present.

Secondly, there is a correlation between the WER of the aligner and noise – when the WER is low there is likely to also be low noise. This means it may be difficult to get reliable alignments for estimating speech models *and* diversity of noise for estimating

non-speech models.

Table 7.3 shows what happens to the training and cross-validation errors when we train DNNs with segments from different bands of alignment WER. We trained models with segments of WER up to 10, 20 and 30%. We see that as we increase the threshold, the corresponding training and validation errors increase. This shows that the training data is becoming simultaneously more corrupted and more challenging to represent.

Table 7.3: Training and cross-validation errors for DNN models trained on forced-aligned data with different bands of WER.

Max. WER(%) of train segs	Train Err.	Valid. Err.
10	6.38	8.14
20	8.65	10.45
30	9.43	11.03

Excluding manual segmentation, the best method to improve the training data may be to iteratively train models then re-segment the full dataset. However, with thousands of hours of data, this would be very costly in terms of computational time.

7.8 Conclusion

In this chapter we explored the use of Deep Neural Networks (DNNs) to improve the robustness of speech segmentation for the MGB challenge dataset which comprises audio from television broadcasts. We showed that DNN-based models can outperform conventional GMM-based models for both speech segmentation and speech recognition metrics.

On of the main challenges prohibiting further improvement is the availability of large quantities of reliably supervised data. This is a problem that is becoming increasingly more difficult and costly to solve through human intervention, i.e. manual transcription. Therefore, a better approach would be to devise methods to automatically improve the quality of supervision or to better select reliable data.

Chapter 8

Future Work and Conclusion

This chapter serves as a conclusion for the findings of this thesis and offers some ideas for future work that could follow on from it. We present some examples for speech segmentation, speaker diarization and then for potential downstream tasks that could be investigated.

8.1 Speech Segmentation

Multi-task DNNs

One of the major problems we encountered when training DNNs for speech segmentation is that the quality of the training labels is often unreliable. What is deemed a 'speech segment' by manual annotators often contains a high proportion of non-speech at the frame level. With such coarse labelling the DNN is unable to learn anything inherent about speech that should help it to classify robustly in challenging scenarios such as when it is mixed with music and noise.

We propose instead that speech segmentation DNNs should be trained in a multi-task manner (Seltzer and Droppo, 2013). Here, we would design a training scheme for a network that jointly learns to predict speech segments along with a smaller unit of speech such as mono-phones. This would force the network to learn something about what actually constitutes speech so that it can still be classified in the presence of noise.

Richer Speech/Non-speech Segmentation

Until now we have only considered segmenting audio into speech and non-speech categories. However, for many scenarios it would be useful to provide a richer segmentation that includes, for example, explicit labels for music, applause, laughter, speech plus noise, etc. This could be useful for adapting downstream systems or simply to provide additional auditory analysis of a recording.

The challenge would initially be gathering data as there are few data sets that contain this level of supervised labels and it would likely require at least some manual annotation.

We did some preliminary experiments whereby we used our speaker diarization system on the *non*-speech segments of some broadcast data. We found that there was some homogeneity within the clusters – a cluster of applause, a cluster of music, etc. This suggests that we could use a similar method to perform an initial unsupervised clustering of a dataset that could simply be corrected and labelled

by human annotators. The benefit could be a significant reduction in the time required to annotate a corpus.

8.2 Speaker Diarization

DNN-based Features

While we have shown that conventional feature extraction techniques for speech processing such as MFCC and PLP parametrization can work for Speaker Diarization (see Section 2.2), they are still not inherently designed for the task. We would like to know if Deep Neural Networks (DNNs) can be used to find a feature space that is more speaker discriminative. The motivation for using DNNs is their inherent ability to explore a wide potential feature space with a large temporal context.

We have completed some initial work whereby we created a DNN with an input layer comprising 19 MFCC coefficients with 40 frames of context either side of the central frame. The output layer was a set of 100 speaker label targets from the AMI corpus. We tried several hidden layers (3-5) with a differing number of neurons per layer (512-2048). After these hidden layers and before the output layer we impose a bottleneck layer with a small number of neurons (20-40). The idea behind this is that the network will be forced to classify the speakers through the bottleneck layer, so we can assume that the activations of this layer will be equivalent to highly speaker-discriminative features. These features could then be used in a conventional GMMHMM system in place of the standard MFCCs.

We found that outright speaker diarization performance did not improve significantly. However, the features did seem to be highly channel and gender discriminative. As these factors represent the greatest variance across the speakers in the training set, it may be have been the case that the network was exhausting its resources to make these partitions in the feature space and did not have enough discriminative power to actually split speakers. This could perhaps be resolved by exploiting some kind of feature stream combination, whereby conventional features are combined with the DNN bottleneck features.

Extension to the Break-prior Model

The speech segmentation method introduced in Chapter 5 considered the task to be a search for the optimal break sequence over an initial sequence of break can-

didates. This method lends itself to being extended for the purposes of speaker diarization by jointly considering speaker change points along with breaks. Standard cluster merging scores, such as BIC, could be used in the decision to include a candidate break or speaker change point in order to find an optimal path over the whole session. If speaker segments are successfully found, the same technique could be applied to each speaker on an individual level, allowing the breaks between segments of the same speaker to be merged. This would allow for speaker segments to ultimately be overlapping, which is currently not a feature of the system.

The SAD system could be adapted with little effort for this task. The challenge would mostly be related to finding suitable ways to evaluate the performance and control the rate of break smoothing.

We have already completed some preliminary work exploiting a segment duration prior which is currently ready to be submitted to an academic conference in short paper form (see Appendix C).

Effect on ASR for Meetings

Another critical metric for measuring diarization performance should be how it propagates on to benefit ASR output. It may, for example, transpire that the optimal diarization for ASR may not be the same as that which correctly identifies each speaker. Consider that if 2 speakers are very similar then it may be better to group them together for the sake of creating 1 adaptation transform for both.

Currently, we only had fully working ASR systems designed for TED talks or broadcast media. The results for the TED system on RT data are shown in Table 8.1. Clearly it is not an appropriate system for this task and would need to be replaced with a dedicated single distant microphone meetings recogniser to achieve state-of-the-art performance suitable for further experimentation. Some promise at least comes from the fact that diarization does seem to improve the WER of even the TED system.

8.3 Downstream Tasks

We would also like to investigate further the behaviour of speech segmentation and speaker diarization on a wider range of downstream tasks. Ultimately, speech segmentation and speaker diarization systems are rarely used in isolation so it is important to

Segmentation	E_{spch}^{miss}	E_{spch}^{fa}	WER
Manual SAD	-	-	64.0
Break Smooth SAD	2.58	2.04	64.4
+Spkdia	2.58	2.04	63.2

Table 8.1: WER for RT data with TED ASR system.

know how well they perform for a variety of different tasks that they may precede.

Machine Translation

We have had some success exploring the effect of speech segmentation on MT-based tasks (see Chapter 5). However, there are many more opportunities to take this work further. We could, for example, look at coupling the segmentation with the MT system directly in order to find segmentation parameters that maximise the MT model performance. We did not, for example, investigate the effect of including more linguistic information into the segmentation process during a second pass after an initial ASR run.

This is likely to benefit from the punctuation restoration work introduced in Chapter 6 as MT systems are generally designed to work with sentence-like units. Punctuation restoration could therefore help to format speech transcripts into the kind of input expected by MT systems.

Diarization is less likely to have an influence here, however one could imagine a speaker dependant MT model which caters to the nuances of individuals. This could be of benefit in scenarios where certain speakers have strong idiosyncratic behaviour or perhaps if there is disparity between the language proficiency of speakers – e.g. we may change the MT model for a speaker who regularly makes the same grammatical errors.

Subtitling

The recent work on speech segmentation and punctuation restoration lends itself well to the task of Automatic Subtitling. With acoustic and linguistic knowledge we can derive strategies for the presentation of subtitles that allow for a balance between showing word sequences that are of temporal relevance as well as those which syntactically make sense.

Speaker Diarization would also allow us to colour or annotate subtitles according

to the discourse that is taking place between speakers. Indeed, we can consider the task of colourising speaker changes as being a special case of speaker diarization where we may not care if we get the overall number of speakers correct so long as we get all speaker change points.

The main challenge here would be in evaluating subtitles beyond simply WER. The most appropriate subtitling sequence is subjective and would need to be evaluated as such.

Speaker Identification

This task is closely related with Speaker Diarization, whereby we would also be able to identify a speaker in a new recording from a pool of known speakers, or indeed, hypothesise that it is an unknown speaker.

There are many experiments that could take place within the context of this task. This may include scenarios when part or all of a group of speakers are known – a supervised diarization in a sense. There is currently no significant existing work in the community combining diarization with identification and many of the state-of-the-art identification methods would not be directly applicable to such a task as they often require extensive training data or complex models.

Punctuation Restoration

In Chapter 6 we were able to show how speech segmentation information can be exploited to aid the task of punctuation restoration. While this purely acoustic method was unable to compete with our machine translation-based method, we were able to show that there is potential to combine this information. Future work would investigate the use of acoustic break information in combination with linguistic methods such as the MT-based system. In theory, the acoustic information would be able to override the linguistic information, and vice-versa, in certain conditions e.g. there is a long acoustic pause but the linguistic model did not hypothesise a period.

The implication of such a system could mean improved text segmentation and richer transcription of speech. This could ultimately improve the interface with subsequent text-based NLP systems such as machine translation and text summarization.

8.4 Conclusion

This thesis presented findings on the tasks of speech segmentation and speaker diarization. We began in Chapter 1 by describing each task in terms of its objectives and challenges, showing that they are both ill-defined problems in nature. This lead us to outline a formal theoretical framework in order to establish a 'language' for discussing and representing the problems, the relationship between them and how to evaluate potential solutions for them.

In Chapters 2 and 3 we looked at some of the state-of-the-art feature extraction and system implementation methods. We showed how speech segmentation and speaker diarization compare and contrast with the task of automatic speech recognition. In particular, we drew attention to the converse relationship between the objectives of speaker diarization and speech recognition – the former is required to be highly speaker discriminative and phone-independent, while the latter is the opposite.

The shortfalls and persistent challenges still faced by state-of-the-art speaker diarization methods were investigated in Chapter 4. Here, we were able to isolate the influence of individual system components on the overall performance through a series of oracle experiments. This allowed us to motivate the direction of our research in order to most effectively improve system performance.

From the oracle experiments we found that speech activity detection remains one of the largest contributors to overall speaker diarization error. We observed that many speech segmentation systems do not exploit the inherent structure of speech. This serves as inspiration for the novel speech segmentation method presented in Chapter 5, whereby we investigate the use of non-acoustic prior information regarding the temporal distribution of segments. We show that this information can be exploited to improve existing acoustic speech segmentation methods. We also show how this method can produce segments that are closer to the manual segmentation resulting in better performance for tasks that expect sentence-like units such as machine translation. Additionally, we show that the ill-defined nature of the speech segmentation problem is also dependent on the downstream task – an optimal segmentation for speech recognition may not be optimal for other NLP tasks such as machine translation.

By highlighting the importance of speech segmentation and how its effects can propagate to downstream NLP tasks, we also looked for other ways information from this stage can be exploited. In Chapter 6 we provide an example of this by showing the relationship between acoustic speech segmentation and the task of punctuation restora-

tion. We show how the timings of breaks between speech segments are correlated with punctuation marks and can be used to actually restore them without any linguistic knowledge. We compare this with a state-of-the-art machine translation-based punctuation restoration method that used linguistic information and offered potential ideas for how acoustic and linguistic methods could be combined.

In Chapter 7 we look at improving the robustness of our speech segmentation system by replacing the GMM acoustic model with a Deep Neural Network (DNN) based model. We show how DNNs are capable of including a much wider temporal context than GMM-based methods as well as incorporating a much larger feature space. Experimental results indicated that we were able to improve the robustness of our speech segmentation system for a challenging broadcast media based dataset. This improvement propagated onward to improve the resultant word error rate for the speech recognition output.

Overall, we believe that this thesis can serve as a basis for future speech segmentation and speaker diarization work. Often these tasks are overlooked and their challenge underestimated in the wider speech processing community. By showing how much effect they can have on down-stream speech recognition and NLP tasks we have highlighted the potential that can be realised by improving such components in the initial stages of larger end-to-end systems. We hope that the formal description of the problems that we have provided along with an in-depth analysis of their objectives as well as potential solutions, can be used to motivate subsequent research on similar topics.

Appendix A

Conference Papers - Research

WHERE ARE THE CHALLENGES IN SPEAKER DIARIZATION?

Mark Sinclair[*]*, Simon King*[†]

The Centre for Speech Technology Research, The University of Edinburgh, UK

M.Sinclair-7@sms.ed.ac.uk, Simon.King@ed.ac.uk

ABSTRACT

We present a study on the contributions to Diarization Error Rate by the various components of a typical speaker diarization system. Following on from an earlier study by Huijbregts and Wooters, we extend into more areas and draw somewhat different conclusions. From a series of experiments combining real, oracle and ideal system components, we are able to conclude that the primary cause of error in diarization is the training of speaker models on impure data, something that is in fact done in every current system. We conclude by suggesting ways to improve future systems, including a focus on training the speaker models from smaller quantities of pure data instead of all the data, as is currently done.

Index Terms— speaker diarization, diarization error rate

1. INTRODUCTION

Speaker Diarization involves segmenting audio into speaker-homogenous regions and labelling regions from wach individual speaker with a single label. Knowing both *who* spoke and *when* has useful applications and can form part of a rich transcription of speech. The task is challenging because it is generally performed without any *a priori* knowledge about the speakers present, not even how many speakers there are.

The NIST Rich Transcription (RT) evaluation campaign [1] ran annually between 2002 and 2009, focusing on promoting Metadata Extraction (MDE) for speech. For some of the years, the campaign included a dedicated speaker diarization task and the use of the associated datasets and evaluation tools have come to form the standard for developing and comparing most current systems. The NIST RT challenges have probably been the most significant driving force for community interest and support for speaker diarization.

The more recent campaigns (RT05/06/07/09) focused on diarization of meetings data. However, system performance on this task has been notoriously meeting-dependent and hyper-sensitive to system parameters [2]. Diarization systems based on agglomerative clustering generally involve an initialisation step, followed by interleaved iterations of re-segmenting the speech, re-estimating the speaker models, and merging models, to gradually converge on the correct number of speakers and the best segmentation and speaker assignments. This architecture means that the final system performance is a complex function of the performance of the individual parts, making it very difficult to identify the causes of error. The work we present here was motivated by the need for a better understanding of the system component factors that contribute to diarization error. Our ultimate goal is to identify where improvements are needed and, conversely, which parts of the system already work well.

Huijbregts and Wooters [3] conducted an investigation along similar lines in 2007. Our investigation is complementary to that work: we investigate several aspects of the system that they did not consider in detail, and we also reach different conclusions about where efforts should be focussed in order to reduce diarization error rate. Our methodology is broadly similar to theirs: we start with a diarization system that is capable of good performance in the standard fully-unsupervised mode, and then conduct various 'oracle' experiments to isolate the effects of various components.

First, we describe the system in Section 2 and then introduce the methodology and experiments in Section 3, summarising our findings and making conclusions about where to focus effort in Sections 4 and 5.

2. SYSTEM DESCRIPTION

There are several diarization systems with competetive state-of-the-art performance such as ICSI [4], IDIAP [5], LIA-EURECOM [6] and I^{2}R [7]. We used our own modular speaker diarization system and chose parameters and methods that would closely emulate that of the ICSI system. The performance of our system is therefore comparable e.g. for single distant microphone (sdm) RT09 data, ICSI has an average of 31.3%DER [4] vs. our 31.8%.

Unlike many other systems (e.g., [8]) we choose not to use a beamformed signal from multiple channels of a microphone array and instead opt for single distant microphone data. A beamformed signal typically improves DER results for systems that ignore overlap [9], but could be a poor choice if we wish to detect a number of simultaneous speakers.

[*]Funded by an EPSRC studentship.

[†]Partially funded by EPSRC grant EP/I031022/1 (Natural Speech Technology) and from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287678 (Simple4All).

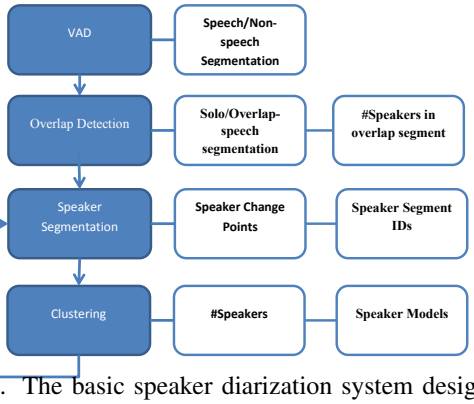


Fig. 1. The basic speaker diarization system design, showing information at each stage that can be replaced with oracle knowledge.

Speech Activity Detection (SAD) is performed by the QIO-Aurora toolkit [10] as this has proven to work well with the RT datasets [11]. For the regions labelled as speech, feature extraction is performed using HTK [12]: we use the first 19 MFCCs computed from a bank of 26 Mel-scaled triangular filters with a pre-emphasis coefficient of 0.97 and cepstral lifting coefficient of 22. We used an analysis window of 30ms and a timeshift of 10ms.

The system uses a GMM-HMM framework whereby 16 clusters (states) are initialised with speech data by dividing the speech frames uniformly into 32 parts and using 2 parts (from different points in the data) to initialise each of the 16 GMMs. Given these models, the system then segments all speech using the Viterbi algorithm with a forced minimum duration constraint of 250ms. After segmentation, the models are retrained, and this is followed by a clustering step in which the most similar clusters are merged – the choice of which clusters to merge is based on the Bayesian Information Criterion. The putative merged model has a complexity (i.e., number of model parameters) equal to the sum of the complexity of the models being merged, which means that a penalty factor parameter is not required.

The process of segmentation and clustering is then iterated until a termination criteria is met: for example, all BIC scores for putative cluster merges are negative. Fig.1 shows an outline of the system design and also illustrates information at each stage that can be replaced by oracle knowledge.

3. EXPERIMENTS

3.1. Data

We used the data from RT06, RT07 and RT09 in a series of experiments designed to control for the influence of separate system components by replacing them with oracle or *ideal* equivalents. Often, in the literature, we see that results on the RT corpora are presented by campaign year. However there are no inherent differences in terms of task or conditions and, while inter-meeting variations are observed in results, no inter-campaign variations are. Therefore, results for all meetings are presented here together as a single set.

3.2. Diarization Error Rate

The main metric for system evaluation is the Diarization Error Rate (DER) which is a sum of three contributing factors as shown in Eq.1: speaker missclassification *SpkErr*, false alarm *FA* (speaker attributed when no speech exists) and missed speech *Miss* (speaker not attributed when speech exists).

$$DER = E_{Spkr} + E_{FA} + E_{Miss} \quad (1)$$

However there is some contention between how overlap should be considered. Some authors [3] choose to take the *FA_speech* and *Miss_speech* errors from the SAD which essentially ignores overlap and results in a lower overall DER. This error is referred to as *speech* time error in the results computed by NIST tools[1].

Others [13] choose to report the *FA_speaker* and *Miss_speaker* errors inclusive of overlap, e.g. a segment which contains two speakers that has been completely missed by the system will have twice the error. This error is referred to as *speaker* time error by the results of the NIST tools and is in fact the default formulation of the *overall speaker diarization error* of the output. This form is used for all results shown in this paper.

The difference between *Miss_speech* and *Miss_speaker* is attributed to overlap. For systems which do not consider overlap, there is no difference between *FA_speech* and *FA_speaker*.

3.3. System configurations

The system was configured to use various combinations of real, oracle and 'ideal' components.

3.3.1. End-to-End

This is the fully automatic unsupervised system. The system is not provided with any oracle knowledge. Apart from a few heuristically-selected parameters (as is the case for all diarization systems), it is completely unsupervised. SAD is done automatically using the QIO-Aurora toolkit. These are the standard conditions for speaker diarization.

3.3.2. Oracle Number of Speakers

One key problem in the clustering stages of diarization is knowing when to stop. Over-clustering will lead to the speech being labelled with too few speakers and typically this results in a sudden increase in DER. In this condition the clustering stops at precisely the known number of speakers per meeting.

[1] http://www.itl.nist.gov/iad/mig/tools

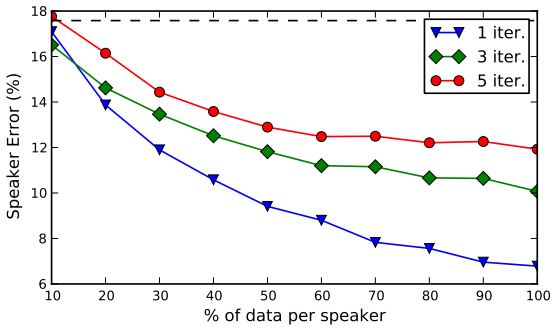


Fig. 2. The *SpkErr* obtained when creating ideal speaker models using varying amounts of data, up to and including all the data. The three lines show the results for segmentation using these models directly ('1 iter.') and when re-segmenting and re-training the models on all the data in the usual iterative fashion ('3 iter.' and '5 iter.'). The dashed line is the DER of the Oracle SAD system.

3.3.3. Oracle SAD

This condition is the same as End-to-End, except only that the initial SAD segmentation is provided by reference transcriptions. This is intended to give insight into how SAD-related errors made at the beginning of the process propagate to other parts of the system. All speaker IDs are relabelled as *speech* and are then collapsed into a standard speech/non-speech segmentation (i.e., overlap is not represented).

3.3.4. Ideal Cluster Initilization

Normally the initial seed clusters to the algorithm are derived by uniformly dividing the data and attributing a portion of it to each cluster. In this condition the clusters are instead each initialized with homogenous data belonging to only one speaker. In order to maintain a similar amount of data in each cluster as in the End-to-End condition, each speaker's data is split across a number of clusters based on the proportion of his or her speaking time. The number of initial clusters is the same as in the End-to-End condition. Ideally, at each iteration, the algorithm should choose to merge clusters in such a way as to maximise cluster purity – that is, belonging to the same speaker. This condition allows us to check how sensitive the clustering process is to initialization.

3.3.5. Ideal Models

The speaker models are rather simple: Gaussian mixture models with simple duration modelling. It is reasonable to ask whether these are adequate for the task. The reference transcription is used to create optimal speaker models by creating a number of clusters equal to the known number of speakers and training each with data from one speaker only. This way, we can discover whether the models themselves and the associated acoustic feature set have sufficient speaker discrimination power for the task.

Table 1. Experimental Results for RT06/07/09

System Stage	*Miss_Spkr*	*FA_Spkr*	*SpkrErr*	*DER*
end2end	14.86	2.34	16.38	33.58
numspks	14.86	2.34	16.68	33.88
SAD	10.50	0.00	17.58	28.09
SAD_idealclust	10.50	0.00	15.27	25.78
SAD_numspks_idealclust	10.50	0.00	15.46	25.96
SAD_idealmodels	10.50	0.00	6.78	17.29
0OL	19.04	0.0	0.07	19.11
1OL	10.2	0.0	0.99	11.19
2OL	1.35	0.0	5.21	6.56
allOL	0.0	0.0	5.87	5.87

We also vary the amount of data used to train these ideal models, from 10% of the available data per speaker up to 100%. We examine the effect of further iterations of re-segmentation + re-training (no merging) too, from 1 iteration (i.e., segmentation with ideal models) up to 5 iterations of re-segmenting + re-training. These iterations *should* improve the models (or, in the 100% data case, do nothing).

3.3.6. Overlap Segmentation

SAD is used prior to diarization to classify the signal into speech and non-speech (e.g., silence, music, noise, etc.). We could also benefit from knowing if each speech segment contains one speaker (solo speech) or multiple (overlap speech). This condition employs such a three-class segmentation derived from reference transcriptions. We first use the ideal models to select the most likely speaker for each solo speech segment. Then, at the end, we revisit overlapping segments and attribute more speakers to them, based on the top few most likely models. Thus, overlap speech is ignored during model training, but is still labelled with speaker ids.

4. RESULTS

Oracle Number of Speakers: As Table 1 shows, knowing the number of speakers has little effect on performance and in sometimes degrades it. Slightly too many clusters can actually be better, if each speaker is well represented – i.e., speaker-attributed clusters have high purity and the *extra* clusters are small. Continuing until the oracle number of speakers is reached may result in incorrect cluster merges.

Oracle SAD: One of the more substantial contributing factors to overall DER was found to be the intial SAD. The automatic method was subject to Missed Speech error in particular. Adding an oracle segmentation, of course, completely eliminates all *Miss_speech* and *FA_speech* error. However, as observed in Fig.3, it is worth noting that this does not propagate on to a substantial reduction in *SpkErr*. This suggests that, while still important, the performance of the diarization algorithm itself is not highly dependent on SAD. Importantly, this also indicates that it is safe to use oracle SAD when investigating other components of the system.

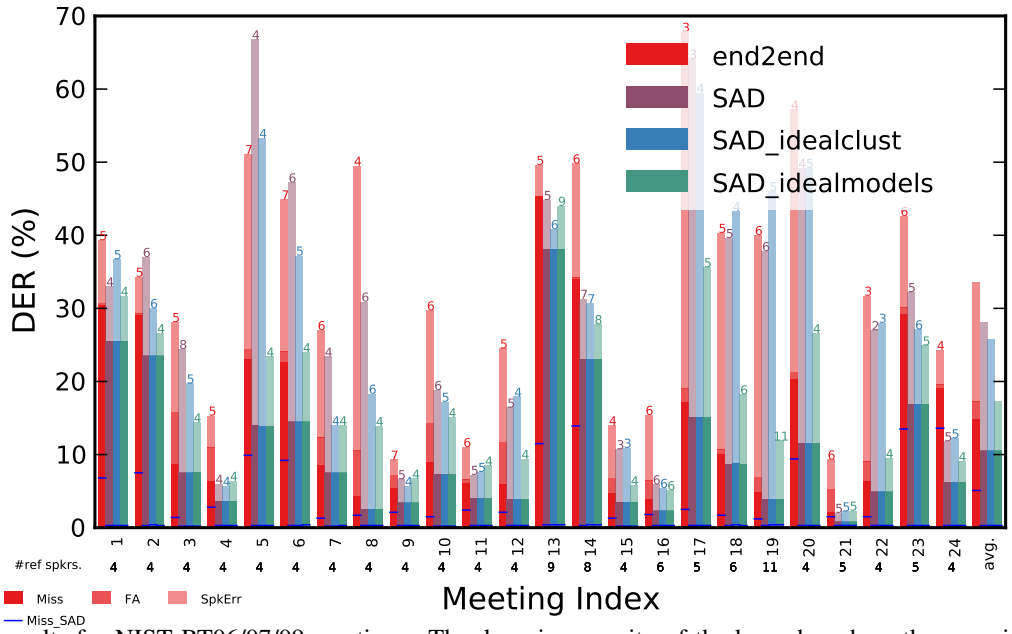


Fig. 3. DER results for NIST RT06/07/09 meetings. The decaying opacity of the bars show how the error is composed of Missed and False Alarm speaker time (inclusive of overlap) as well as speaker error (due to speaker id misclassification). The blue horizontal bar indicates the amount of missed speech error contributed by SAD. Above each bar the number of hypothesised speakers is shown and the reference is provided parallel to the x-axis.

Ideal Cluster Initilization: While the average *SpkErr* shows a reasonable reduction Fig.3, the inclusion of ideal cluster initialisations has greatest effect for meetings where the End-to-End system gave a high *SpkErr*. This suggests that poor cluster initiliastion, whereby initial clusters all have a low purity, may be non-recoverable.

Ideal Models: Providing the system with ideal models trained on each speaker's data substantially reduces *SpkErr*, confirming the models do work. Fig.2 shows the effect of varying the amount of data used to train the models. As little as 10% improves over baseline. Worryingly, more iterations *degrades* performance. This suggests cluster purity is critical to the clustering process: impurities introduced at each iteration cannot be accommodated, and the models do not recover.

Overlap Segmentation: As we see in Table 1, ignoring overlap (0OL) is costly (19.11% in this case), especially when using *Miss_Spkr* to calculate DER. By attempting to get at least 1 speaker right per overlap region, we halve that error. An average minimum 10.50% DER is always incurred if only 1 speaker at a time is possible, but by getting at least the 2nd speaker correct, we halve the error again.

5. CONCLUSIONS

5.1. Relation to Huijbregts and Wooters

As Huijbregts and Wooters [3] also found, results are highly dependent on the evaluation data (i.e., high variation in DER between meetings) and some system components can be sensitive to the performance of preceeding ones. Like them, we found that SAD can be a major contributor to DER by directly contributing *Miss_speech*, but we would add that subsequent components actually have little dependence on its performance.

5.2. New Findings and Future Work

One of the key findings from our experiments is the importance of estimating the speaker models on pure data: speech from just one speaker. If this could be achieved, dramatic reductions in DER would result (Fig.2). Even if only a fraction of the data for each speaker could be reliably identified, free from the polluting effects of data from other speakers, than large improvements would still be expected. Methods for estimating some form of *confidence* in speaker homogeneity when seeding clusters with data should therefore work well, even if that entails rejecting a large proportion of the data.

Our ideal models are strong enough to allocate multiple speakers to overlap regions. So another focus of future research should be in overlap-speech detection. Systems which do not consider overlap will always concede substantial error.

The take-home message, given that further iterations degrade models that were initially pure (Fig.2), is that the final set of speaker models should not necessarily be trained on all data to be diarized, but only on reliably-identified pure data.

6. REFERENCES

[1] NIST, *Spring 2006 (RT-06S) Rich Transcription Meeting Recognition Evaluation Plan*, 2006.

[2] N. Mirghafori and C. Wooters, "Nuts and Flakes: a Study of Data Characteristics in Speaker Diarization," in *Proc. IEEE Int Acoustics, Speech and Signal Processing Conf. ICASSP 2006*, 2006, vol. 1.

[3] Marijn Huijbregts and Chuck Wooters, "The blame game: performance analysis of speaker diarization system components.," in *INTERSPEECH*. 2007, pp. 1857–1860, ISCA.

[4] G. Friedland, A. Janin, D. Imseng, X. Anguera, L. Gottlieb, M. Huijbregts, M. Knox, and O. Vinyals, "The ICSI RT-09 Speaker Diarization System," *IEEE Transactions on Audio, Speech, and Language Processing*, , no. 99, 2011, Early Access.

[5] Deepu Vijayasenan, Fabio Valente, and Petr Motlícek, "Multistream speaker diarization through Information Bottleneck system outputs combination.," in *ICASSP*. 2011, pp. 4420–4423, IEEE.

[6] Simon Bozonnet, Nicholas W. D. Evans, and Corinne Fredouille, "THE LIA-EURECOM RT'09 SPEAKER DIARIZATION SYSTEM: ENHANCEMENTS IN SPEAKER MODELLING AND CLUSTER PURIFICATION," *ICAASP 2010*, 2010.

[7] Hanwu Sun, Bin Ma, Swe Zin Kalayar Khine, and Haizhou Li, "Speaker diarization system for RT07 and RT09 meeting room audio.," in *ICASSP*. 2010, pp. 4982–4985, IEEE.

[8] Xavier Anguera, Chuck Wooters, and Jose M. Pardo, "Robust Speaker Diarization for Meetings: ICSI RT06s evaluation system," *Interspeech 2006*, 2006.

[9] X. Anguera, C. Woofers, and J. Hernando, "Speaker diarization for multi-party meetings using acoustic fusion," in *Proc. IEEE Workshop Automatic Speech Recognition and Understanding*, 2005, pp. 426–431.

[10] André Gustavo Adami, Lukás Burget, Stéphane Dupont, Harinath Garudadri, Frantisek Grézl, Hynek Hermansky, Pratibha Jain, Sachin S. Kajarekar, Nelson Morgan, and Sunil Sivadas, "Qualcomm-ICSI-OGI features for ASR.," in *INTERSPEECH*, John H. L. Hansen and Bryan L. Pellom, Eds. 2002, ISCA.

[11] Erich Zwyssig, Steve Renals, and Mike Lincoln, "Determining the number of speakers in a meeting using microphone array features.," in *ICASSP*. 2012, pp. 4765–4768, IEEE.

[12] Steve J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book Version 3.4*, Cambridge University Press, 2006.

[13] Xavier Anguera Miró, Simon Bozonnet, Nicholas W. D. Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals, "Speaker Diarization: A Review of Recent Research.," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 20, no. 2, pp. 356–370, 2012.

A semi-Markov model for speech segmentation with an utterance-break prior

Mark Sinclair[1]*, Peter Bell*[1]*, Alexandra Birch*[2]*, Fergus McInnes*[1]

[1]Centre for Speech Technology Research, [2]Statistical Machine Translation Group
School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, UK
{mark.sinclair, a.birch, peter.bell, fergus.mcinnes}@ed.ac.uk

Abstract

Speech segmentation is the problem of finding the end points of a speech utterance for passing to an automatic speech recognition (ASR) system. The quality of this segmentation can have a large impact on the accuracy of the ASR system; in this paper we demonstrate that it can have an even larger impact on downstream natural language processing tasks – in this case, machine translation. We develop a novel semi-Markov model which allows the segmentation of audio streams into speech utterances which are optimised for the desired distribution of sentence lengths for the target domain. We compare this with existing state-of-the-art methods and show that it is able to achieve not only improved ASR performance, but also to yield significant benefits to a speech translation task.

Index Terms: speech activity detection, speech segmentation, machine translation, speech recognition

1. Introduction

We define speech segmentation as the problem of finding the end points of a speech utterance in time. While this may at first seem like a relatively simple goal it is in fact a non-trivial problem to define. As speech does not strictly follow the same rules we find in written language, such as sentence breaks, it can often be highly subjective as to what constitutes an appropriate segmentation of speech for a given task. The vagueness and high-order decision processes that surround these concepts make it challenging to design an effective automatic speech segmentation system.

Most automatic speech segmentation methods work by identifying speech and non-speech regions based on acoustic evidence alone e.g. contrasting energy levels or spectral behaviour[1] [2] [3]. Some more recent research has improved upon this foundation by using richer feature sets that are more suited to the task or include long-term dependances [4] [5] [6]. Others have begun to apply deep learning techniques which can garner more discriminative features and improve robustness [7] [8] [9]. However, all of these methods still only consider the acoustics and are not necessarily exploiting the underlying structure of the spoken language.

Human transcribers, on the other hand, are capable of segmenting speech by exploiting a greater wealth of prior information such as syntax, semantics and prosody in addition to such acoustic evidence. As a result, human transcribers may opt to ignore acoustically motivated 'breaks' in speech in favour of maintaining longer segments based on semantic knowledge. Such an informed segmentation can greatly influence subsequent system components that have been explicitly designed to exploit the patterns and structure of natural language, e.g. the language models used in automatic speech recognition (ASR) or machine translation (MT).

Previous work on detecting segmentation of sentence like units has looked at using features such as prosody [10], language model scores [11] [12], translation model scores [13] and syntactic constituents [14]. [15] presents a review of some of this work and also motivates tuning the segmentation of speech to the task at hand as we do in this paper. Our approach of modelling global sentence length distribution is orthogonal to much of this previous work, and combining these information sources would be beneficial. There has been some previous work which attempts to exploit some of these cues [16]. However, in the present paper, we have limited our focus to the use of statistics of utterance durations and present a novel way of exploiting this to select a globally optimal sequence from acoustically motivated'break candidates'. We find that this yields an advantage over the use of local acoustic information alone at putative pauses in the speech. We also find indications that the optimal setting of the segmentation parameters varies with the ultimate task (e.g. transcription or translation) that is to be achieved using the segmented speech. We present results on segmentation, recognition and translation of TED talks[1].

2. Utterance-break Modelling

While the automatic speech segmenters that we initially used are only able to segment on an acoustic basis, they would actually perform this task very well. When compared to the manual segmentation we found the False Alarm rate to be very low (2-3%) while the more dominant error is Missed Speech.

Empirical evidence suggests that the automatic segmenters work very well at framewise classification but are not able to distinguish when a non-speech segment is simply a pause inside a speaker's utterance or a true 'break' between utterances as judged by human annotators. Often such pauses are quite short and as such a naïve approach might be to simply alter the minimum duration constraint for non-speech regions. However, in practice we find that this simply shifts the balance from Missed Speech to False Alarm errors by removing more potential breaks, quickly resulting in over-long segments. A significant part of this behaviour is due to the fact that such systems are only able to make local decisions about whether or not to include a non-speech segment. To remedy this problem, we propose to investigate methods for globally optimising the sequence of utterance breaks, incorporating prior knowledge about the likelihood that non-speech breaks should be included given their temporal relationship i.e. the duration between them.

2.1. Break Candidates

As a precursory step to find the globally optimal sequence of utterance breaks B^*, we first need to derive a sequence of can-

[1]http://www.ted.com

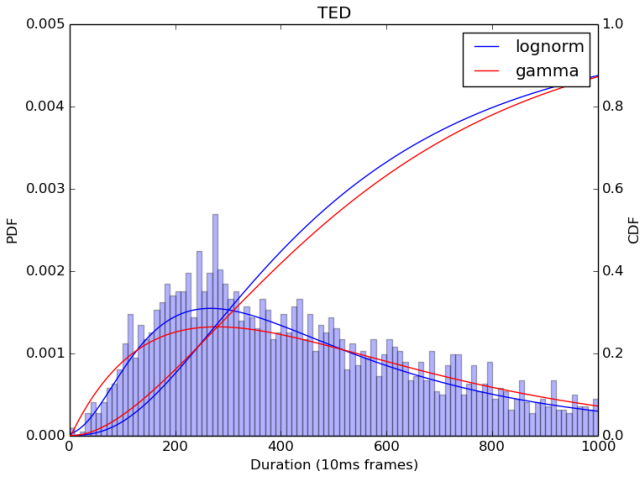


Figure 1: *Log-normal vs. Gamma PDFs fitted to speech segment durations of TED dev set. The CDFs provide the prior likelihood of a new break given the duration after the last break.*

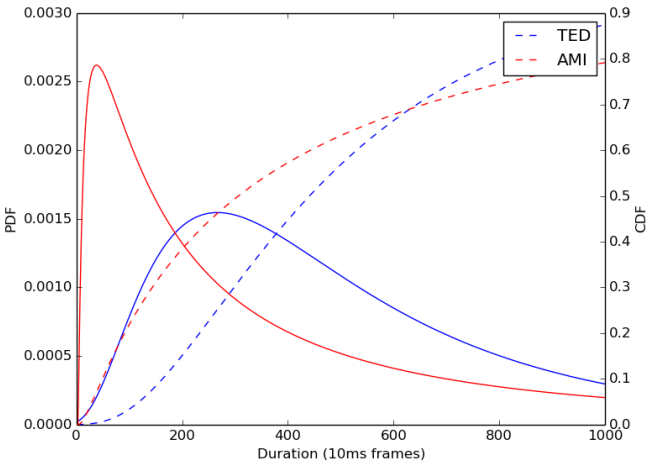


Figure 2: *Log-normal PDFs and corresponding CDFs for speech segment durations of TED vs AMI data.*

didate break points B. Ideally the candidate sequence should be broad enough that it includes a good optimal sequence as a sub-set so we would therefore require that $|B| \geq |B^*|$, where $|B|$ and $|B^*|$ are the cardinalities of the candidate and optimal sequences respectively. The candidates themselves can be determined by any kind of initial segmentation method such as an existing acoustically motivated speech segmentation algorithm.

2.2. Utterance-break Prior

In order to make decisions about whether or not to include a candidate break, we need to know the prior probability of a break, which we condition on the time since the last break was observed. This may be derived from a statistical model of segment durations. Figure 1 shows a histogram of speech segment durations for a development set of lecture data. We investigated the use log-normal and gamma distributions to represent the behaviour of the data and ultimately chose the former as it provided a slightly better fit as shown in Figure 1. To derive the break likelihood prior we simply use the cumulative density function (CDF) of this distribution as shown in Equation 1 where d is the duration since the previous break and α is a scaling factor to account for the difference in dynamic range compared to the acoustic likelihood.

$$f_{brk\_utt}(d) = (\frac{1}{2} + \frac{1}{2}\text{erf}\left[\frac{\ln d - \mu}{\sqrt{2}\sigma}\right])^{\alpha} \quad (1)$$

We were interested in how such a prior may vary according to the domain. Therefore, as a contrast to the prepared, rehearsed, single-speaker speech that is found in TED talks, we also looked at the distribution of speech segments for a set of AMI scenario meetings. These meetings contain multiple speakers discussing a given common task whereby the speech is unprepared and spontaneous. From Figure 2 we can observe that the distribution has a much lower mean, illustrating the intuition that speech segments are generally shorter during such dynamic group discourse. This suggests that the break likelihood prior could be adapted for different domains to achieve optimal performance.

2.3. Viterbi Decoding

In order to determine the globally optimal sequence of utterance breaks B^* we consider a sequence of candidate breaks $B = \{b_0, \ldots, b_{|B|}\}$ as a semi-Markov process whereby each candidate is a state. We can then perform Viterbi decoding over a sparse $|B| \times |B|$ trellis, whereby each position i moves, not through uniform time segments, but through the indices of B. Each position j allows us to consider the transition arriving at break b_i from b_j. As the break candidate states are only forward connected we can only arrive at a given break from a previous break, hence $j < i$. We keep a vector of tuples T that records the start and end frame indices of each break, this allows us to calculate the duration between any pair of break candidates $d_{i,j} = t_{i,start} - t_{j,end}$.

We also use the sums of the frame-level log-likelihoods from the speech/non-speech segmenter for acoustic features X, where x_i is a vector representing all the frames that are in break b_i. We use this to create a posterior probability P_{brk_aco}, as shown in Equation 2, that represents the acoustic probability of a given break. The purpose of the normalisation is that we want the acoustic likelihood of breaks to increase with duration so that long breaks are favoured.

$$P_{brk\_aco}(i) = \frac{\ell_{nonspch}(x_i)}{\ell_{nonspch}(x_i) + \ell_{spch}(x_i)} \quad (2)$$

Therefore, the probability of the partial sequence that has a break at i is formalised as

$$v_i = \max_{j<i} [v_j + \log f_{brk\_utt}(d_{i,j})] + \log P_{brk\_aco}(i) \quad (3)$$

with $v_0 = 0$. For each candidate i we store the identity of the state j which maximises Equation 3. This allows to B^* to be recovered by a backtrace procedure.

As the duration between the break candidate under consideration and earlier ones increases, it will become very unlikely that such long term transitions would occur. In practice, we can therefore afford to prune the lattice by ignoring transitions from earlier states that are further away than a prescribed maximum segment duration, δ. As such, for each index i we only consider transitions from states where $d_{i,j} \leq \delta$. This is illustrated in Figure 3.

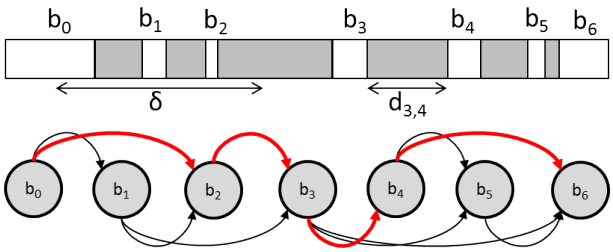


Figure 3: *An example of a candidate break sequence and associated state topology. We can see that the states can only feed forward and some long-term transitions have been pruned such as $b_0 \rightarrow b_3$ as $d_{0,3} > \delta$. The transitions highlighted in red show an example optimal break sequence $B^* = \{b_0, b_2, b_3, b_4, b_6\}$*

3. Experiments

3.1. Data

For evaluation, we used the data made available for the IWSLT evaluation campaigns[17]. This comprises a series of TED talks that have been divided into development sets (dev2010 and dev2011) and a test set (tst2010), each containing 8-11 talks. The talks average just under 10mins in length and each contains a single English speaker (either native or non-native). The talks are manually segmented and transcribed at the utterance level. We also had manual English-French translations for evaluating the MT system component.

3.2. Speech Segmentation Systems

3.2.1. Manual

Here we simply pass the manual segmentation to the ASR and MT systems directly in order to form the oracle standard with which to compare our automatic speech segmenters.

3.2.2. SHOUT

This system makes use of the SHOUT Toolkit (v0.3)[2][18] which is a widely-used off-the-shelf speech segmentation system. The tool uses a GMM-HMM-based Viterbi decoder, with an iterative sequence of parameter re-estimation and re-segmenting. Minimum speech and silence duration constraints are enforced by the number of emitting states in the respective HMMs.

3.2.3. Baseline segmenter

Our baseline system, labelled "simple" in the tables, is identical to that used for our recent lecture transcription system [19] and comprises a GMM-HMM based model which is used to perform a Viterbi decoding of the audio. Speech and non-speech are modelled with diagonal-covariance GMMs with 12 and 5 mixture components respectively. We allow more mixture components for speech to cover its greater variability. Features are calculated every 10ms from a 30ms analysis window and have a dimensionality of 14 (13 PLPs and energy). Models were trained on 70 hours of scenario meetings data from the AMI corpus using the provided manual segmentations as a reference. A heuristically optimised minimum duration constraint of 500ms is enforced by inserting a series of 50 states per class that each have a transition weight of 1.0 to the next, the final state has a self transition weight of 0.9.

[2]http://shout-toolkit.sourceforge.net/download.html

3.2.4. Break Smooth

Here we introduce our utterance-break prior model. In order to establish the candidate break sequence we use the system in Section 3.2.3 to do an initial segmentation pass over the data. The only exception is that the minimum duration constraint is reduced to 100ms. If used directly, this would normally perform very poorly as a VAD but when used as input to the subsequent break smoothing we have three advantages over the original constraint: better guarantee of enough candidates to find an optimal solution, the ability to find shorter speech segments ($\geq$100ms), and more accurate end-points for segments between 100-500ms. The break likelihood prior was trained on the speech segment durations of the dev2010 and dev2011 sets. The maximum segment duration δ is set to 30 seconds.

We have also shown results for 2 different operating points of the scaling factor α, 30 and 80. While this parameter is designed to mitigate for the difference in dynamic range with the acoustic model, we found it subsequently functioned as a form of segment duration tuning whereby a greater α results in more break smoothing and hence longer segments.

3.2.5. Uniform

As well as our automatic methods we also considered segmenting each talk into uniform speech segments of length N seconds, which is equivalent to having a break of zero length at every interval. This allowed us to check whether or not the benefit of our utterance-break prior may simply be due to an 'averaging' of the break distribution. As the ASR system is still able to do decoder-based segmentation within each given segment, this is also a way of measuring its influence. Here, longer uniform segments leave more responsibility to the decoder for segmentation and at $N = 300$, the maximum segment length for the ASR system, we effectively allow the decoder to do all the segmentation (with potentially a small error at the initial segment boundaries).

3.3. Downstream System Descriptions

3.3.1. Automatic Speech Recognition (ASR)

ASR was performed using a system based on that described in [19]. Briefly, this comprises deep neural network acoustic models used in a tandem configuration, incorporating out-of-domain features. Models were speaker-adapted using CMLLR transforms. An initial decoding pass was performed using a 3-gram language model, with final lattices rescored with a 4-gram language model.

3.3.2. Machine Translation (MT)

We trained an English-French phrase-based machine translation model using the Moses [20] toolkit. The model is described in detail in our 2013 IWSLT shared task paper [21]. It is the official spoken language translation system for the English-French track. It uses large parallel corpora (80.1M English words and 103.5M French words), which have been filtered for the TED talks domain. The tuning and filtering used the IWSLT dev2010 set.

The goal of our machine translation experiments is to test the effect that ASR segmentation has on the performance of a downstream natural language processing task. The difficulty

with allowing arbitrary segmentations in MT is that automatic evaluation is performed matching MT output with gold reference sentences which have their own manual segmentation. In order to evaluate translations which have different segmentations, we need to align the MT output segmentation with the reference. We use a tool provided by the Travatar [22] toolkit which aligns files with different segmentations. It searches for the optimal alignment according to the BLEU score. We use it to align our MT output with a variety of different segmentation models, to our gold reference with manual segmentations. We align each TED talk in the test set separately to maximize performance.

3.4. Gold Transcription Mapping

Any improvement a given segmentation provides to the ASR system could subsequently improve the performance of the MT system. However, this makes it difficult to infer how much of the MT performance gain is simply a consequence of a better source transcript as compared with the direct influence of the segmentation itself. To control for this, we used the ASR system to make a forced alignment of the manual transcription in order to gain word-level timing information. We were then able to map any of our given segmentations with the same gold-standard transcription.

4. Results

We present results for all of our end-to-end automatic systems in Table 1. Firstly, we note that our *Simple* segmenter is able to significantly outperform *SHOUT*, confirming that we have competitive acoustic segmenter with which to form the foundation of our experiments. We can then see that our *Break Smooth* segmenter is further able to improve the performance of both ASR and MT over the *Simple* segmenter.

The performance of the *Uniform 300s* system showed a strong performance for ASR, falling only slightly short of our best performing *Break Smooth* system. We attribute this to the nature of TED talks whereby there is typically very little non-speech (illustrated by the 6.49% FA of *Uniform 300s*), which itself mostly comprises periods of silence of small duration, which is implicitly segmented by the silence HMMs used by the decoder itself. In contrast, however, when we use a uniform segmentation for MT, we find that it does not perform as well despite the good ASR performance. As the MT system ideally expects sentence-like segments, a uniform segmentation will not be practical for these purposes. This also shows that a segmentation that works well for ASR may not necessarily work well for MT and vice-versa.

In order to fully control for the dependence MT has for the WER of the ASR transcript it receives, we have shown the results for when we map each of our segmentations to the force-aligned gold transcription in Table 1. First of all, from the variation in performance we can infer that segmentation does indeed have a direct effect on MT performance. However, in these conditions we find that the MT system favours the break smoothing algorithm with shorter segments than MT. Figure 4 shows how the prior and posterior distributions compare. We can see that when $\alpha = 30$ the distribution takes a closer shape to the true distribution with a 'shift' to shorter segments, which could be due to the fact that the automated methods have more accurate segment boundaries. As such the MT system in this case could be benefitting from more 'sentence-like' utterances, whereas the ASR system can actually afford to have, and may actually benefit from, slightly longer segments as it is able to further segment in more detail using its own decoder.

	SAD		ASR	MT:ASR	MT:Gold
Segmentation	Miss	FA	WER	BLEU	BLEU
Manual	-	-	13.6	0.2472	0.2472
SHOUT	12.71	0.16	18.3	0.1967	0.2256
Simple	9.91	2.66	16.7	0.2007	0.2319
Break Smooth 30	4.25	2.33	15.0	0.2085	0.2409
Break Smooth 80	7.86	1.46	14.6	0.2104	0.2368
Uniform 300s	0.00	6.49	14.8	0.2014	0.2369

Table 1: Segmentation, ASR and MT results for each segmenter. MT results are shown for both ASR output and gold transcripts segmented with different segmentation models.

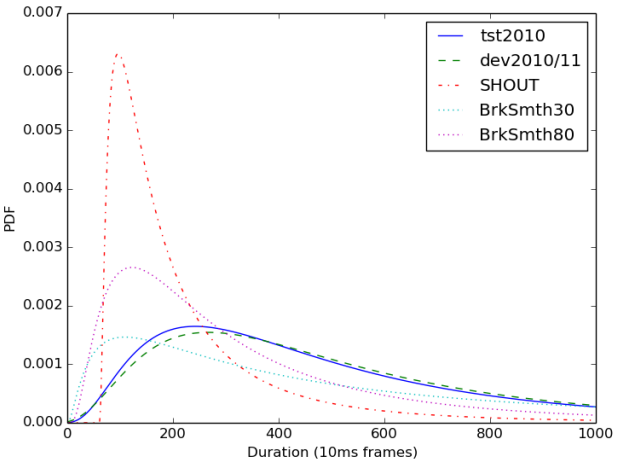


Figure 4: *A comparison of the prior and posterior segment length distributions.*

5. Conclusions and Future Work

We have shown that speech segmentation can be improved by exploiting non-acoustic prior knowledge – in this case, the use of an utterance-break model. Such improvements can be shown to propagate to further benefit the performance of downstream tasks such as ASR and MT. We have also shown that the benefits to MT are not simply a consequence of the benefits to ASR suggesting that speech translation performance is highly dependent on the quality of the speech segmentation. However, we observed that the optimal segmentations for each task are not necessarily the same – furthermore, typical speech segmentation evaluation metrics are not a reliable indicator of downstream system performance.

Given what we have learned from this investigation we believe there is scope in future work to add linguistic knowledge into the segmentation model, such as language modelling scores and even syntactic bracketing information. This would require running segmentation as an iterative procedure, on the output of an ASR model, before feeding it back in as the input to an ASR system.

6. References

[1] I. Boyd and D. K. Freeman, "Voice activity detection," Jan. 4 1994, uS Patent 5,276,765.

[2] J. Ramırez, J. C. Segura, C. Benıtez, A. De La Torre, and A. Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech communication*, vol. 42, no. 3, pp. 271–287, 2004.

[3] R. Tucker, "Voice activity detection using a periodicity measure," *IEE Proceedings I (Communications, Speech and Vision)*, vol. 139, no. 4, pp. 377–380, 1992.

[4] J. Ye, T. Kobayashi, M. Murakawa, and T. Higuchi, "Incremental acoustic subspace learning for voice activity detection using harmonicity-based features." in *INTERSPEECH*, 2013, pp. 695–699.

[5] M. Graciarena, A. Alwan, D. Ellis, H. Franco, L. Ferrer, J. H. Hansen, A. Janin, B. S. Lee, Y. Lei, V. Mitra *et al.*, "All for one: feature combination for highly channel-degraded speech activity detection." in *INTERSPEECH*, 2013, pp. 709–713.

[6] A. Tsiartas, T. Chaspari, N. Katsamanis, P. K. Ghosh, M. Li, M. Van Segbroeck, A. Potamianos, and S. Narayanan, "Multi-band long-term signal variability features for robust voice activity detection." in *INTERSPEECH*, 2013, pp. 718–722.

[7] N. Ryant, M. Liberman, and J. Yuan, "Speech activity detection on youtube using deep neural networks." in *INTERSPEECH*, 2013, pp. 728–731.

[8] X.-L. Zhang and J. Wu, "Deep belief networks based voice activity detection," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 4, pp. 697–710, 2013. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6362186

[9] F. Eyben, F. Weninger, S. Squartini, and B. Schuller, "Real-life voice activity detection with lstm recurrent neural networks and an application to hollywood movies," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 483–487. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6637694

[10] A. Stolcke, E. Shriberg, R. A. Bates, M. Ostendorf, D. Hakkani, M. Plauche, G. Tür, and Y. Lu, "Automatic detection of sentence boundaries and disfluencies based on recognized words." in *ICSLP*, 1998.

[11] A. Stolcke and E. Shriberg, "Automatic linguistic segmentation of conversational speech," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 2. IEEE, 1996, pp. 1005–1008. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=607773

[12] E. Matusov, A. Mauser, and H. Ney, "Automatic sentence segmentation and punctuation prediction for spoken language translation." in *IWSLT*, 2006, pp. 158–165.

[13] E. Matusov, D. Hillard, M. Magimai-Doss, D. Z. Hakkani-Tür, M. Ostendorf, and H. Ney, "Improving speech translation with automatic boundary prediction." in *INTERSPEECH*, vol. 7, 2007, pp. 2449–2452.

[14] B. Roark, Y. Liu, M. Harper, R. Stewart, M. Lease, M. Snover, I. Shafran, B. Dorr, J. Hale, A. Krasnyanskaya *et al.*, "Reranking for sentence boundary detection in conversational speech," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1. IEEE, 2006, pp. I–I. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1660078

[15] M. Ostendorf, B. Favre, R. Grishman, D. Hakkani-Tur, M. Harper, D. Hillard, J. Hirschberg, H. Ji, J. G. Kahn, Y. Liu *et al.*, "Speech segmentation and spoken document processing," *Signal Processing Magazine, IEEE*, vol. 25, no. 3, pp. 59–69, 2008. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4490202

[16] D. Rybach, C. Gollan, R. Schluter, and H. Ney, "Audio segmentation for speech recognition using segment features," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 4197–4200.

[17] M. F. M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 evaluation campaign," *Proceedings IWSLT 2012*, 2012.

[18] M. A. H. Huijbregts, "Segmentation, diarization and speech transcription: surprise data unraveled," 2008.

[19] P. Bell, F. McInnes, S. R. Gangireddy, M. Sinclair, A. Birch, and S. Renals, "The UEDIN english ASR system for the IWSLT 2013 evaluation," *IWSLT, Heidelberg, Germany*, 2013.

[20] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of the Association for Computational Linguistics Companion Demo and Poster Sessions*. Prague, Czech Republic: Association for Computational Linguistics, 2007, pp. 177–180. [Online]. Available: http://www.aclweb.org/anthology/P/P07/P07-2045

[21] A. Birch, N. Durrani, and P. Koehn, "Edinburgh SLT and MT System Description for the IWSLT 2013 Evaluation," in *Proceedings of the 10th International Workshop on Spoken Language Translation*, Heidelberg, Germany, December 2013, pp. 40–48.

[22] G. Neubig, "Travatar: A Forest-to-String Machine Translation Engine based on Tree Transducers," in *Proceedings of the ACL Demonstration Track*, Sofia, Bulgaria, August 2013.

Appendix B

Conference Papers - System

The UEDIN English ASR System for the IWSLT 2013 Evaluation

Peter Bell, Fergus McInnes, Siva Reddy Gangireddy,
Mark Sinclair, Alexandra Birch, Steve Renals

School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, UK
{peter.bell,fergus.mcinnes,a.birch,s.renals}@ed.ac.uk,
{s.gangireddy,m.sinclair-7}@sms.ed.ac.uk

Abstract

This paper describes the University of Edinburgh (UEDIN) English ASR system for the IWSLT 2013 Evaluation. Notable features of the system include deep neural network acoustic models in both tandem and hybrid configuration, cross-domain adaptation with multi-level adaptive networks, and the use of a recurrent neural network language model. Improvements to our system since the 2012 evaluation – which include the use of a significantly improved n-gram language model – result in a 19% relative WER reduction on the `tst2012` set.

1. Introduction

We report on experiments carried out for the development of automatic speech recognition (ASR) systems on the English datasets of the International Workshop on Spoken Language Translation (IWSLT) 2013. We report our work on the new TED German task in an accompanying paper [1] since the development of the two systems was largely independent. Work on our machine translation system may be found in [2]. Significant changes to the English ASR system since 2012 include improvements to our baseline language models, described in Section 2.1, and the use of recurrent neural network language models, described in Section 2.2. The acoustic models are described in Section 3 – the main addition is that we now use deep neural networks in a hybrid configuration, and apply automatic voice activity detection to the `tst2013` test set.

2. Language modelling

The ASR system used Kneser-Ney smoothed N-gram language models for decoding and lattice rescoring, and a recurrent neural network (RNN) language model for a final rescoring stage based on N-best lists. These models are described in the subsections below.

This work was supported by the European Union under the FP7 projects inEvent (grant agreement 287872) and EU-Bridge (grant agreement 287658), and by EPSRC Programme Grant grant EP/I031022/1, *Natural Speech Technology*.

2.1. N-gram models

The N-gram language models were obtained by interpolating individual modified Kneser-Ney discounted LMs trained on the small in-domain corpus of TED transcripts and the larger out-of-domain (OOD) sources. The OOD sources were Europarl (v7), News Commentary (v7), News Crawl (2007 to 2011) and Gigaword (Fifth Edition).

The News Crawl and Gigaword sources in particular contained a wide variety of phenomena such as money amounts and other numerical expressions, abbreviations, and listed and tabulated information, which required normalisation to create data resembling spoken word sequences. Considerable effort was put into developing appropriate text normalisation scripts. Starting from the scripts used in LM training for the IWSLT 2012 evaluation, over 1000 lines of Perl code and 1400 abbreviation entries were added (expanding the original files by more than 50%). The processing applied to the data can be summarised as follows.

1. Remove documents that are not of type *story*, strip out markup and split text into sentences (required for Gigaword only).
2. Eliminate duplicate lines (common in some newswire sources, where multiple copies or variants of the same story may occur).
3. Convert Unicode characters and encodings for fractions, symbols etc into standard ASCII forms such as "1/4" (for subsequent conversion to words).
4. Filter out newswire datelines, e.g. "LONDON, Nov 2", and other extraneous material.
5. Normalise punctuation, abbreviations, units of measurement etc.
6. Convert numerical expressions to words.
7. Remove punctuation and convert to lower-case without diacritics.
8. Convert British to American English spellings and correct some common spelling errors.

The vocabulary for the ASR system was defined so as to include all words occurring in the in-domain training corpus (other than words which occurred only once and were not in a standard dictionary) and all words exceeding specified occurrence count thresholds in the OOD corpora, while remaining below the maximum of 64K words imposed by the version of HDecode in use here. The vocabulary size was 62,522.

Initialisms included in the vocabulary were treated as single words for LM purposes, e.g. "u.s." (with the dots retained to distinguish them from words such as "us"). Once the vocabulary had been defined, out-of-vocabulary initialisms were broken into single letters, e.g. "m. f. n.", so as to be modelled as sequences of in-vocabulary words (letter names) rather than treated as OOV.

In view of the mismatch in content and style between the target domain (TED talks) and the OOD data, a data selection process [3, 4] was applied to the OOD corpora to obtain an appropriate subset of data for LM training. The set of out-of-domain data D_S was chosen by computing a cross-entropy difference (CED) score for each sentence s:

$$D_S = \{s | H_I(s) - H_O(s) < \tau\} \quad (1)$$

where $H_I(s)$ is a cross-entropy of a sentence with a LM trained on in-domain data, $H_O(s)$ is a cross-entropy of a sentence with a LM trained on a random subset of the OOD data of similar size to the TED corpus, and τ is a threshold to control the size of D_S

Language models were trained on the in-domain and OOD data using the SRILM toolkit [5], and were interpolated with weights optimised on the TED development set (dev2010 and tst2010: total 44,456 words).

Perplexities on the development set with 3-gram and 4-gram models trained on the TED corpus and selected OOD data are shown in Table 1. Selecting 25% of the OOD sentences yielded an OOD training set of 751M words; setting the CED threshold to 0 gave a smaller but more targeted set of 312M words, which gave a lower perplexity on the TED data than the 751M word set when used alone to train the LM, but a slightly higher perplexity after interpolation with the TED LM. The perplexities obtained here are substantially lower than the values of 160 (3-gram) and 159 (4-gram) with the LMs used in our IWSLT 2012 system [6], which were trained using a much smaller set of OOD data with no CED filtering.

The LMs finally used in the ASR system were the TED+312MW trigram model (for decoding) and the TED+312MW 4-gram model (for lattice rescoring). The amounts of data from the respective sources used in these LMs are shown in the "Selected" column of Table 2. Comparison with the total sizes of the source corpora (after text normalisation) given in the preceding column shows that the proportion of data selected by the CED criterion ranged from 8% for the Gigaword corpus to 15% for News Commentary.

Language model	Perplexity
TED 3-gram	183.2
OOD (312MW / 751MW) 3-gram	133.5 / 138.3
TED+OOD (312MW / 751MW) 3-gram	125.1 / 124.9
TED 4-gram	179.9
OOD (312MW / 751MW) 4-gram	123.9 / 126.4
TED+OOD (312MW / 751MW) 4-gram	114.9 / 113.4

Table 1: Perplexities of N-gram language models on TED development set.

Corpus	Total	Selected
TED	2.4M	2.4M
Europarl	53.1M	6.3M
News Commentary	4.4M	0.7M
News Crawl	693.5M	72.9M
Gigaword	2915.6M	232.9M
OOD total	3666.6M	312.8M

Table 2: Numbers of words in LM training sets.

2.2. RNN models

Neural network language models have shown to consistently improve the word error rates (WER) of LVSCR tasks [7, 8, 9]. For this year's evaluation, we investigated the effectiveness of RNN LMs for TED lecture transcription. To study the effectiveness of RNNs we rescored the n-best hypothesis using RNNs trained on in-domain and different subsets of out-of-domain (OOD) data, shown in Table 3, selecting according to the CED score as in Section 2.1 In-domain data consists of 2.4M tokens. Since it is very difficult to train the RNNs on large amounts of OOD data, we restrict the maximum size of OOD data to 30M.

The number of hidden neurons ranged from 300 to 500 and number of classes in the output layer was 300. Models are trained using RNN training tool of [10]. Table 4 shows the perplexity (PPL) and WER on on development data provided by IWSLT evaluation campaign. We can observe that rescoring the n-best hypothesis with the RNNs reduce the WER by 0.8%. We choose the best model from this experiments to rescore the n-best hypothesis from `tst2011`, `tst2012` and the `tst2013` test sets. The interpolation weight between n-gram and RNNLM is optimised on devel-

Table 3: Subsets of OOD data

#Words	#Sentences	Threshold(τ)
5M	664.2K	-1.14
10M	1156.7K	-0.963
15M	1596.7K	-0.862
20M	2011.3K	-0.79
25M	2412.6K	-0.733
30M	2792.4K	-0.687

Table 4: Perplexity and WER on development data

Tokens	Vocabulary	PPL	WER(%)
n-gram	-	-	15.6
7.4M	47.7K	171.56	15.2
12.4M	54.8K	161.66	15.2
17.4M	61.7K	147.17	15.0
22.4M	68K	142.22	14.9
27.4M	74.3K	133.5	14.8
32.4M	80K	**126.0**	**14.8**

opment data, to minimise WER.

3. Acoustic modelling

For the acoustic modelling components of the system, we used a setup identical to that described in [11], where more details may be found. Briefly, we used a combination of tandem and hybrid deep neural network (DNN) systems trained on a corpus of in-domain TED talks, incorporating out-of-domain data of multi-party meetings from the AMI corpus using the multi-level adaptive networks (MLAN) technique [12]. Compared to our 2012 system, the main addition is the use of DNNs with MLAN features in the hybrid framework. We describe this further below. Additionally, unlike earlier test sets from the IWSLT evaluation, the 2013 test set was not provided with a manually derived segmentation; we therefore employed an automatic segmentation system, described in Section 3.3.

3.1. Training data

For in-domain training data, we used 813 TED talks recorded prior to the end of 2010. The talks were segmented and aligned to the crowd-sourced transcriptions available online using a lightly-supervised technique described in [13]. This produced 143 hours of labelled speech segments for use in acoustic model training. Additionally, we used 127 hours of out-of-domain data from the AMI Corpus of multi-party meetings[1] using a setup based on [14]. This data is not in general well-matched to the TED-domain. The OOD data was not used directly in acoustic model training, but used to generate out-of-domain neural network features for the in-domain data.

3.2. Deep neural network systems

For our 2012 system, we used neural networks within the tandem framework [15, 16], using DNNs to generate log probabilities over monophones. The monophone probabilities are decorrelated and projected to 30 dimensions, then augmented with the original acoustic features to give a total feature vector of 69 dimensions. These vectors are used for standard HMM-GMM training. Additionally in this year's system, we

[1] http://www.amiproject.org/

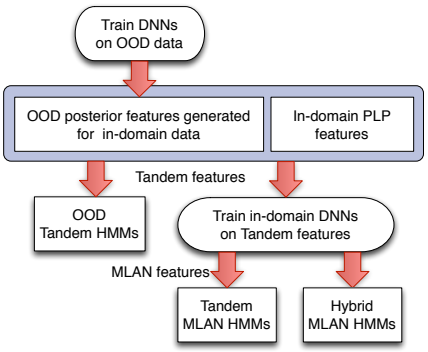


Figure 1: Tandem and hybrid MLAN training

used DNNs in a hybrid configuration, generating posterior probabilities over tied-state triphones, as proposed in [17]. These are converted to pseudo-likelihoods for use in the decoder.

Both tandem and hybrid nets used PLP input features with 9 frames of temporal context. For the tandem systems, the final nets used had four hidden layers with 1024 hidden units per layer; the hybrid systems used six hidden layers with 2048 hidden units per layer. The tandem nets had an output layer of size 46; the size of the output layer of the hybrid nets varies according to the number of tied states, which resulting from clustering with a GMM; it was typically around 6,000. The nets were trained with a tool based on the Theano library [18] on NVIDIA GeForce GTX 690 GPUs. For the tandem systems, we applied speaker adaptive training of the GMMs using CMLLR [19] regression class trees with 32 classes. For the hybrid systems, we performed adaptation of the input feature space at training and test time using a global CMLLR transform for each speaker. Tandem systems were discriminatively trained with MPE.

As in the 2012 system, we incorporated out-of-domain data using the MLAN technique. Neural networks were trained on the AMI corpus and the resulting nets used to generate posterior features for each utterance in the TED corpus. Th2ese neural net features are known to provide a degree of domain-independence [20]. In the MLAN scheme, the OOD features are augmented with the original acoustic features and a further DNN is trained on these features, allowing further adaptation to the target domain. This second adaptive network may be used to generate tandem features, or used in a hybrid system. The possible configurations are illustrated in Figure 1.

3.3. Voice activity detection

The voice activity detection component of the system comprises a GMM-HMM based model which is used to perform a Viterbi decoding of the audio. The HMM has 2 classes: speech and non-speech. These are modelled with diagonal-covariance GMMs with 12 and 5 mixtures respectively. We allow more mixture components for speech to

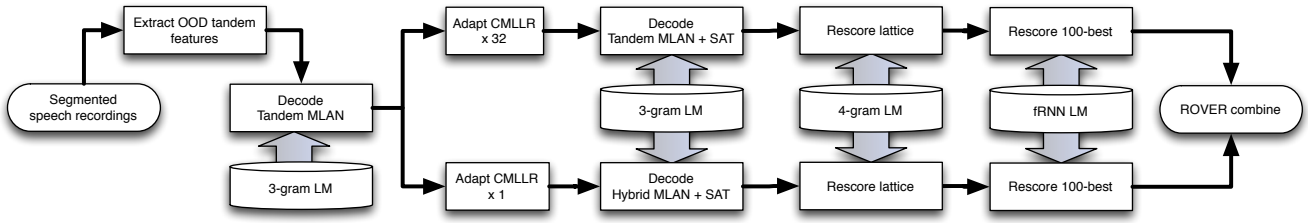


Figure 2: The full decoder architecture

cover its greater variability. Features are calculated every 10ms from a 30ms analysis window and have a dimensionality of 14 (13 PLPs and energy). Models were trained on 70 hours of scenario meetings data from the AMI corpus using the provided manual segmentations as a reference. To avoid over segmentation a minimum duration constraint of 50ms is enforced by inserting a series of 50 states per class that each have a transition weight of 1.0 to the next, the final state has a self transition weight of 0.9.

4. Decoder architecture

Figure 2 shows the complete decoding architecture. After an initial pass, used to generate transcripts to estimate speaker transforms, we operate two parallel decoding sequences for the tandem and hybrid acoustic models. For each model, the complete process consists of a decoding with the trigram LM using HTK's HDecode[2]. Lattices output from the this pass were rescored using the 4-gram LM, generating 100-best lists, which were rescored with the final interpolated RNN LM. Finally, the one-best outputs from tandem and hybrid systems are combined at the hypothesis level using ROVER.

5. Results

In this section we first present development results from individual components of the complete system pipeline. Table 5 shows results using the manual segmentations provided for earlier evaluations. The results may differ slightly from official results due to variations in scoring procedure. It may be observed that there is no clear winner out of the tandem and hybrid systems; however, they are clearly complementary as system combination consistently yields improved performance.

The trends are similar when the automatic segmentation is used, shown in Table 6. When the automatic segmentation is used there is a deterioration in performance of up to 3% WER. Some of this may be attributed to an increase in insertion and deletion errors of the result of segmentation errors; however, an additional source of error, particularly affecting the RNN LM, is that the automatic segmenter typically results in shorter segments, not divided along semantic lines as the manual version is, resulting in reduced language modelling power, since we do not propagate LM probabilities across segment boundaries. Note that the results with the RNN model are available only for a subset of experiments as this component of the system was not fully automatic at the time of system development.

[2] `http://htk.eng.cam.ac.uk`

System	dev2010	tst2010	tst2011
Tandem MLAN	15.9	14.1	11.2
+ 4gram	15.6	13.6	10.8
+ RNN	-	-	10.4
Hybrid MLAN	15.6	13.9	11.5
+ 4gram	15.2	13.5	11.3
+ RNN	-	-	10.5
ROVER combination			
4gram	**14.7**	**12.6**	10.3
+ RNN	-	-	**9.9**

Table 5: *Development system results with manual segmentation (WER%)*

System	dev2010	tst2010	tst2011
Tandem MLAN	18.8	17.6	14.9
+ 4gram	18.4	17.2	14.5
+ RNN	17.6	16.6	-
Hybrid MLAN	18.6	17.4	14.6
+ 4gram	18.4	17.2	14.3
+ RNN	17.6	16.7	-
ROVER combination			
4gram	17.6	16.2	**13.2**
+ RNN	**17.0**	**16.1**	-

Table 6: *Development system results with automatic segmentation (WER%)*

Finally, we provide the official results from the 2013 evaluation in Table 7. Automatic segmentation is used only for `tst2013` set. It is notable that the WER is substantially higher on this set than on the other development and evaluation sets. A preliminary analysis suggests that this is probably not due to problems with the segmentation, as insertion and deletion errors do not make up a noticeably higher proportion of the total errors than for the other test sets. Over the talks, the WER ranges from 9% to 48%, suggesting that

	tst2011	tst2012	tst2013
Primary system	10.2	11.6	22.1

Table 7: *Official system results from the 2013 evaluation (WER%)*

perhaps this year's test set contains a more diverse range of acoustic conditions.

6. Machine translation

We applied machine translation to the ASR output. Details may be found in the accompanying paper [2]. Table 8 compares MT performance for various inputs from the ASR system. Note that performing translation from a confusion network containing multiple ASR hypotheses resulted in worse results that using the one-best output. We are investigating the reasons for this – one theory is that, due to the generally low WER of the systems, the alternative hypotheses are rarely correct, often simply indicating OOV errors when they have high acoustic scores. Table 9 presents, for reference, the official 2013 BLEU results comparing, as inputs, the use of our best system, and the transcription by the IWSLT organisers.

ASR input	en-fr
1-best	22.9
1-best punctuated	24.1
Confusion net	18.4

Table 8: Cased BLEU results for models when tuned and tested on ASR output in different formats.

	en-fr
Edinburgh ASR system	22.45
IWSLT ASR system	23.00

Table 9: Official test 2013 cased BLEU results for 1Best SLT input. The Edinburgh ASR system input was our primary system.

7. Conclusions

We have described our ASR system for the English 2013 IWSLT evaluation Improvements to our system since the 2012 evaluation result in relative WER reductions of 17% 19% on the `tst2011` and `tst2012` sets respectively. The use of RNN LMs does not give improved performance on the `tst2013` set, a result that is probably due to the shorter utterances derived from the automatic segmentation.

Improvements planned for future systems include the use of neural network based voice activity detection, and the pooling of German and English audio data in multi-condition DNN training, whereby both systems are trained simultaneously, sharing lower layers of the network. We also plan to apply talk-level language model adaptation.

8. References

[1] J. Driesen, P. Bell, and S. Renals, "Description of the UEDIN system for german ASR," in *Proc. IWSLT*, 2013.

[2] A. Birch, N. Durrani, and P. Koehn, "Edinburgh SLT and MT system description for the IWSLT 2013 evaluation," in *Proc. IWSLT*, Heidelberg, Germany, 2013.

[3] R. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proc. ACL Conference Short Papers*, Uppsala, 2010, pp. 220–224.

[4] H. Yamamoto, Y. Wu, C. Huang, X. Lu, P. Dixon, S. Matsuda, C. Hori, and H. Kashioka, "The NICT ASR system for IWSLT 2012," in *Proc. International Workshop on Spoken Language Translation*, Hong Kong, Dec. 2012.

[5] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit," in *Proc. ICSLP*, vol. 2, 2002, pp. 901–904.

[6] E. Hasler, P. Bell, A. Ghoshal, B. Haddow, P. Koehn, F. McInnes, S. Renals, and P. Swietojanski, "The UEDIN systems for the IWSLT 2012 evaluation," in *Proc. International Workshop on Spoken Language Translation*, Hong Kong, Dec. 2012.

[7] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, pp. 1137–1155, 2003.

[8] H. Schwenk, "Continuous space language models," *Computer Speech & Language*, vol. 21, no. 3, pp. 492–518, 2007.

[9] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH*. ISCA, 2010, pp. 1045–1048.

[10] T. Mikolov, M. Karafiát, L. Burget, J. Černokcý, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. Interspeech*, 2010.

[11] P. Bell, H. Yamamoto, P. Swietojanski, Y. Wu, F. McInnes, C. Hori, and S. Renals, "A lecture transcription system combining neural network acoustic and language models," in *Proc. Interspeech*, Lyon, France, Aug. 2013.

[12] P. Bell, M. Gales, P. Lanchantin, X. Liu, Y. Long, S. Renals, P. Swietojanski, and P. Woodland, "Transcription of multi-genre media archives using out-of-domain

data," in *Proc. IEEE Workshop on Spoken Language Technology*, Miama, Florida, USA, Dec. 2012.

[13] A. Stan, P. Bell, and S. King, "A grapheme-based method for automatic alignment of speech and text data," in *Proc. IEEE Workshop on Spoken Language Technology*, Miama, Florida, USA, Dec. 2012.

[14] T. Hain, L. Burget, J. Dines, P. Garner, F. Grezl, A. Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan, "Transcribing meetings with the AMIDA systems," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 486–498, 2012.

[15] H. Hermanksy, D. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, 2000, pp. 1635–1630.

[16] Q. Zhu, B. Chen, N. Morgan, and A. Stolcke, "On using MLP features in LVCSR," in *Proc. Interspeech*, 2004.

[17] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[18] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proc. SciPy*, June 2010.

[19] M. Gales, "Maximum likelihood linear transforms for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 75-98, 1998.

[20] A. Stolcke, F. Grézl, M.-Y. Hwang, X. Lei, N. Morgan, and D. Vergyri, "Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons," in *Proc. ICASSP*, 2006.

Description of the UEDIN System for German ASR

Joris Driesen, Peter Bell, Mark Sinclair, Steve Renals

Center for Speech Technology Research, University of Edinburgh, UK

{jdriesen,peter.bell,s.renals}@inf.ed.ac.uk, M.Sinclair-7@sms.ed.ac.uk

Abstract

In this paper we describe the ASR system for German built at the University of Edinburgh (UEDIN) for the 2013 IWSLT evaluation campaign. For ASR, the major challenge to overcome, was to find suitable acoustic training data. Due to the lack of expertly transcribed acoustic speech data for German, acoustic model training had to be performed on publicly available data crawled from the internet. For evaluation, lack of a manual segmentation into utterances was handled in two different ways: by generating an automatic segmentation, and by treating entire input files as a single segment. Demonstrating the latter method is superior in the current task, we obtained a WER of 28.16% on the dev set and 36.21% on the test set.

Index Terms: Light supervision, Segmentation, Acoustic Model Training

1. Introduction

In ASR, good acoustic models are an important prerequisite for high recognition accuracies. The quality of these models is determined by both the quality and the quantity of the data on which they were trained. Such data consists of speech as well as accurate orthographic transcriptions. Since the latter must be manually created by human transcribers, which is a slow and expensive process, it can be difficult to obtain training data in sufficiently large quantities. In languages or domains where resources are scarce, i.e., where no large amounts of dedicated transcribed training is available, acoustic models can still be obtained from untranscribed or poorly transcribed data, using unsupervised or lightly supervised training methods [1, 2, 3, 4, 5]. Since German ASR has historically received little attention at UEDIN, there are very few resources available for it on site. Therefore, even though German is by no means an under-resourced language, we have been compelled to treat it as such, collecting large amounts of publicly available data and processing it with the lightly supervised training methods mentioned above. Although this methodology is not strictly necessary for German, it can in theory be applied to unlock other, truly under-resourced languages, for which no alternative training methods exist. The available resources used for acoustic model training are discussed below in section 2. The lightly supervised training is explained fully in section 4. Acoustic model training is finalised by training a Deep Neural Network (DNN) in a hybryd setup with a traditional context-dependent tri-phone based Hidden Markov Model (HMM), as explained below, in section 6.

Aside from acoustic modelling, the proposed system has state-of-the-art language modelling. In a first phase, text corpora are collected, containing in total almost 10^9 words. Based on the cross-entropy with the evaluation domain, as proposed in [6], the top 30 percentile of this data is selected and 4-gram language models, as well as Recurrent Neural Network Language Models (RNNLM) are trained on it [7]. Details of this setup can be found below, in section 5.

Since no manual segmentation for the evaluation set is provided, it is necessary to produce a segmentation automatically. Alternatively, ASR can be performed on entire talks, treating them as a single segment. There is an inherent trade-off between these approaches, since each has its own advantages and disadvantages. A segmentation that is generated automatically may contain erroneous segment boundaries, which can easily lead to recognition errors. When segmentation is avoided, on the other hand, recognition could be performed on non-speech segments, generating unpredictable erroneous outputs. In section 6, evaluation is performed comparing both approaches.

2. Available Resources for Acoustic Modelling

The data on which an ASR system is trained determines to a large extent its eventual performance. Several properties of the training data are important. Firstly, its domain must be matched as closely as possible to the domain of the evaluation set. Even when using techniques like fMLLR [8] to adapt acoustic models to the test domain, any mismatch will significantly reduce recognition accuracies. Also accurate orthographic transcriptions of the training data are necessary. Even small amounts of transcription errors can significantly reduce recognition performance, e.g. [9]. Lastly, the size of the training set plays an important role. Although there is no such thing as a direct linear relation between training set size and recognition performance, having more training data does usually lead to better results. Several tens of hours is believed to be a minimum for acoustic model training, depending on

This work has been funded by the European Union as part of the Seventh Framework Programme, under grant agreement no. 287658 (EU-BRIDGE), and by EPSRC Programme Grant grant EP/I031022/1, *Natural Speech Technology*.

the size and complexity of the models being trained.

2.1. Globalphone

One of the suitable speech corpora accessible to us is GlobalPhone [10]. It is a multi-lingual corpus, covering a selection of the world's most widely spoken languages, one of which is German. For each language, it contains speech from about 100 adult native speakers, reading a number of articles taken from a local newspaper. For German, this adds up to about 18 hours of speech. Only 14 hours of this can be used as training data, since the rest is divided over a dev set and a test set. In the context of this paper, the GlobalPhone corpus is less suitable for acoustic model training, due to its small size and its large domain mismatch with the IWSLT evaluation data. However, the German lexicon that is included in the corpus is invaluable to us, since it is the only lexicon we have at our disposal. It contains 36994 unique words, with 39520 pronunciations, indicating that a relatively large number of words is listed with more than a single pronunciation variant. Furthermore, a 3-gram language model for this data is available to us. It is the same language model that was used in [11], and is specifically tuned to the domain of news articles. Using this LM is not our only option though, since we have the option to train our own, more tuned to the domain of TED-talks, see section 5.

2.2. Europarl

The second set of data was obtained by crawling the website of the European Parliament [12], which has committed itself to making its plenary sessions publicly available online, along with their transcripts. These sessions contain speech in a wide variety of languages, German among them. Although, generally speaking, the transcriptions do not match the spoken content of the speech perfectly, techniques for lightly supervised acoustic model training may be employed to circumvent this. We will elaborate on this below in section 4. In this work, we downloaded all parliamentary sessions of the years 2008, 2009, and 2010. This is about 990 hours of audio data. This data contains 23 audio streams in parallel: one stream with the raw unaltered recordings, and one additional stream for each of the 22 languages of the European Union. In these audio streams, speech in any other language than the target language is replaced with its on-the-fly translation, done in real-time by professional interpreters. For each parliamentary speech, there is only a single start and end time given, shared over all 22 parallel versions of that speech. Since translations may take longer than the original speech, or may be shifted in time, the audio segments delineated by these boundaries are usually 10–20 seconds longer than the speech they contain, and tend to overlap each other. Adding the lengths of all these segments together therefore leads to an overestimate of the available data, but can nonetheless be a useful indication. The total amount of speech data we counted like this, is 733 hours. One must be cautious in using all this data directly, however, since it contains directly recorded speech from German-speaking MEP's, as well as interpreters' speech. There are very distinct differences between these types of speech: e.g. whereas MEP's speak more spontaneously, often with an accent, interpreters tend to speak clearly, with long pauses, and very few corrections and repetitions. Since these types of speech may not be equally well matched to the target domain, we have treated them separately. We identified the speeches that were originally spoken in German, by comparing the German audio stream with the raw unaltered audio. Based on the same rough count as before, this adds up to about 95 hours of speech. Since there is no lexicon available with this data, we reuse the GlobalPhone lexicon, to which the out-of-vocabulary words are added using Sequitor Grapheme-to-Phoneme conversion [13].

3. Text Tokenisation

Although the GlobalPhone lexicon does contain 373 numbers, this list is far from exhaustive. Numbers in the evaluation data are therefore very likely to be OOV. To prevent this from happening, we defined rewrite rules to convert any number that is OOV into its constituent parts, most of which do occur in the lexicon, or are easily added to it. For instance, if "1,234" is encountered, it is rewritten as "1,000 2 100 4 und 30". This way, with no more than 33 lexical entries, we are able to handle any number between 1 and $9,999 \cdot 10^6$. Special exception rules are provided to deal with such things as times, dates, years, and IP-addresses. Measures of distance, length, and volume are fully expanded, as well as currencies, e.g. 'km' is written as 'kilometer', '$' is written as 'dollar', etc. Because of time constraints, handling of abbreviations in our system is rudimentary. Basically, any word that either consists of two or more capitalized letters, or of letters separated by full stops is recognized as an abbreviation. They are then written in a consistent form, namely as uncapitalized letters separated by full stops, and then added to the lexicon using grapheme-to-phoneme conversion. There are several ways in which this methodology is suboptimal. For one, it disregards the possibility of abbreviations being pronounced as words, rather than sequences of separate letters, e.g. the pronunciation of "NATO" as /nato/ rather than /ɛnɑteo/. More importantly, the GlobalPhone lexicon, on which we trained the grapheme-to-phoneme conversion, contains far too few examples to enable accurate pronunciation predictions. As a result, abbreviations in training and evaluation data are expected to reduce the performance of our system.

4. Lightly Supervised Acoustic Model Training

To perform acoustic model training and evaluation, the acoustic data is preprocessed as follows. First, it is converted towards mono-channel 16kHz WAVE-files. MFCC-

coefficients are determined within 25 ms frames which are shifted in increments of 10 ms. Cepstral Mean Normalisation is then applied to the resulting 13-dimensional feature vectors. For each frame, the features within a context window of 9 frames, 4 to the left, 4 to the right, are stacked and projected down to 39 dimensions using LDA-MLLT.

4.1. Training an Initial Model on GlobalPhone

We train an initial GMM-HMM acoustic model from scratch on the GlobalPhone corpus. This model contains 3000 context-dependent states and 48000 Gaussians. It was evaluated on three different evaluation sets: the GlobalPhone dev set, where it resulted in a WER of 12.68%, the GlobalPhone eval set, on which it gave a WER of 19.92%, and the IWSLT dev set, on which it yielded a WER of 56.18%. The language model used in each of these evaluations was the GlobalPhone-specific one, introduced in section 2.1.

4.2. Further Training on Europarl

Acoustic model training on Europarl data cannot be done straightforwardly, since the transcriptions we have of it do not match the acoustics perfectly. There is a variety of light supervision techniques, however, with which this problem may be circumvented, e.g. [14, 1]. Here, we used the greedy matching approach described in [5]. We first bias the GlobalPhone LM towards the Europarl domain by interpolating it with a small LM trained on the imperfect transcriptions. This LM, in combination with the acoustic model trained above in section 4.1, is then used to make a recognition of the Europarl training data. By comparing the recognition result with the imperfect transcription, and greedily collecting the longest sequences that occur in both, a new in-domain training set is constructed. From this, a new acoustic model with the same number of states and Gaussians is trained and the whole process is repeated. This iterative process is illustrated in figures 1 and 2. With each iteration, the accuracy of the ASR transcription is expected to rise, and hence more training data is collected for the iteration after that. Also, with each iteration, the models are expected to get more tuned towards the Europarl domain. In this work, we first apply this technique for 10 iterations on the subset with 95 hours of direct MEP recordings, discussed in section 2.2, and evaluated on the IWSLT dev set in each iteration. The result is shown in the leftmost columns of table 1. The initial WER of 46.36% is obtained with the GlobalPhone acoustic model. The reason why this result is different from the 56.18% reported in section 4.1 is that another LM was used in these evaluations, namely the one that is biased towards Europarl data. Looking at the WER's, we can see that the quality of the acoustic models doesn't improve with each new iteration. If anything, the opposite is true, although the statistical significance of these differences may be questionable. This lack of improvement is probably caused by a slight domain mismatch between Europarl and the TED talks in the IWSLT

	MEP		All	
iter	hours	WER(%)	hours	WER(%)
init	NA	46.36	46.98	41.12
1	45.91	41.13	67.15	40.22
2	46.64	41.20	70.28	40.09
3	46.69	41.36	70.80	**39.95**
4	46.80	41.25	70.83	40.01
5	46.89	41.10	70.92	40.27
6	47.00	41.36	70.93	40.28
7	47.07	41.55	70.99	40.26
8	47.01	41.49	70.95	40.12
9	47.00	41.28	70.89	40.50
10	46.98	41.12	70.94	40.35

Table 1: *The data set sizes and WER rates obtained on the IWSLT dev set in each iteration of lightly supervised training.*

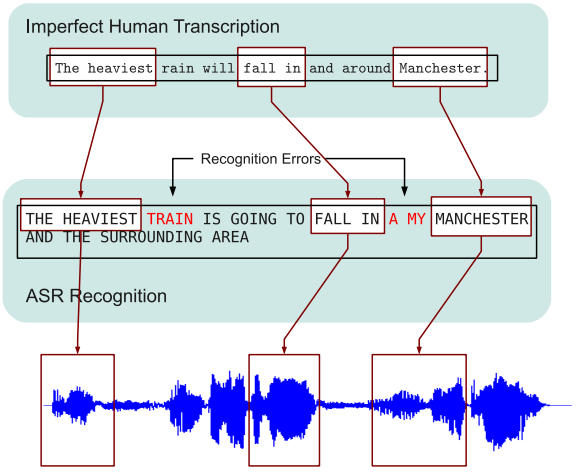


Figure 1: *The longest word sequences occurring both in the approximate transcription and in the ASR output are identified.*

dev set. An interesting experiment would be to evaluate the models in each iteration on an evaluation set in the Europarl domain. Unfortunately, no such evaluation set is available to us. When doing the same experiment on the entire Europarl corpus, MEP speech and interpreters' speech put together, the results become as shown in the rightmost columns of table 1. The acoustic model obtained in iteration 10 of the previous experiment is used here as the initial acoustic model. Although the WER drops about 1% absolute with the inclusion of the interpreters' speech, the results are otherwise comparable to those of the previous experiment. The drop in WER is very likely due to the increase of the training set from 46.98 hours to 67.18 hours. The best performance, a WER of 39.95%, is achieved in the third iteration. Therefore, the training set obtained in that iteration is used for all acoustic model training in further experiments, see section 6.

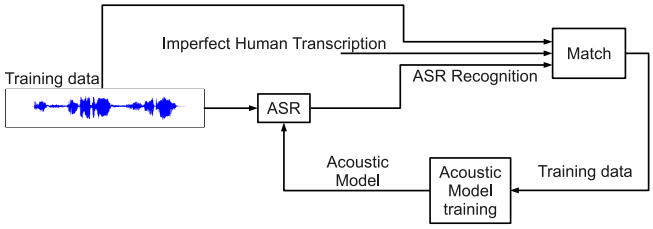


Figure 2: *Illustration of the iterative process, in which training data is collected to obtain acoustic models, which are in turn used to collect a better set of training data.*

name	# words ($\cdot 10^6$)
europarl_v7	47.37
europarl_crawl	2.86
news_crawl_2007	31.47
news_crawl_2008	107.86
news_crawl_2009	101.56
news_crawl_2010	45.89
news_crawl_2011	252.85
news_crawl_2012	319.73
news_comment	4.45
total	914.05

Table 2: *The text resources used for LM training.*

5. Language Modelling

For language model training, we used the resources listed in table 2. All of these were obtained through links on the IWSLT website, except 'europarl_crawl', which consists of the imperfect transcriptions of the Europarl data from section 2.2. All text was first depunctuated and tokenised as described in section 3. From each of these texts, 30% is selected that best matches the domain of the IWSLT dev set, according to the cross-entropy criterion proposed in [6]. Language models are trained on this subset only, disregarding the remaining 70%.

5.1. N-gram Language Models

After winnowing them down to 30%, each of the text corpora is used to train a 3-gram LM, using the MITLM language modelling toolkit [15]. In this training, modified Kneser-Ney smoothing [16] is used with parameters optimised on the IWSLT dev set. These language models are then linearly interpolated with interpolation weights optimised in the same way. The 1-grams in the resulting interpolated model are then written out in decreasing order, according to their smoothed 1-gram probability. Choosing the top-N words from this list allows us to optimally define a dictionary of size N for further LM training. We then repeated the previous procedure, training 3-gram LM's on the whittled down text corpora, with a limited vocabulary of N words, and linearly interpolating them. Finally the same was done with

N	OOV rate(%)	3-gram ppl.	4-gram ppl.
100000	4.18	252.63	246.36
150000	3.32	278.24	263.37
200000	2.78	283.25	275.97
250000	2.52	289.73	282.43
300000	2.37	294.24	**286.86**
350000	2.29	297.03	289.74
400000	2.17	300.30	292.97

Table 3: *The perplexities and OOV rates of the 3-gram and 4-gram LM's on the IWSLT dev set*

4-grams. The OOV rate and perplexity on the dev set for a range of values for N is shown in table 3. As expected, the 4-gram models achieve lower perplexities than 3-gram models. Based on these results, we choose the 4-gram LM with vocabulary size 300000 for the evaluations in section 6, since this yields a good trade-off between word coverage and perplexity. Any of these 300000 words that do not occur either in GlobalPhone or in the crawled Europarl data is added to the lexicon. Using a LM of such size for LVCSR (Large Vocabulary Continuous Speech Recognition) is very demanding in terms of memory and processing power. Therefore, we make a reduced version of this LM, pruning it with a probability threshold of 10^{-7}. The pruned LM is much smaller in size than the original, but this comes at the price of a higher perplexity, which rises from 286.86 to 413.62. Due to its smaller size, it can easily be used to generate word lattices on the evaluation data, which are rescored afterwards using the full unpruned LM. To demonstrate the extent to which they may affect the WER in practice, we perform an ASR evaluation on the IWSLT dev set using the pruned LM, before and after rescoring with the unpruned LM. The acoustic model in this experiment is the optimal model as established in section 4.2. The pruned LM yields in this evaluation a WER of 37.02%, a slight improvement over 39.95%, obtained in section 4.2, with a different LM. Rescoring with the full LM brings the WER further down to 33.69%.

5.2. Recurrent Neural Net Language Models

From a concatenation of all the whittled down text corpora of section 5.1, we train a Recurrent Neural Net Language Model, using the RNNLM toolkit [7]. Due to computational limitations, the vocabulary size for this model is reduced to 50000. The number of nodes in the hidden layer is set to 30. From the final rescored word lattices in section 5.1, N-best lists are generated, with N=100. For each of these 100 recognition hypotheses, the RNNLM is used to calculate a LM score S_{RNNLM}, which is interpolated with the original 4-gram LM score, resulting in the modified score S'.

$$S' = (1-\alpha) \cdot S_{ngram} + \alpha \cdot S_{RNNLM} \qquad (1)$$

This modified score is used to re-rank the N-best list, often changing which hypothesis is considered as the 'best'. The

interpolation factor α was optimised on the dev set, yielding a value of 0.25. Applying this RNNLM rescoring on the word lattices of section 5.1, yields an improvement in WER from 33.69% to 33.17%.

6. ASR System Setup

At this point, we have all the resources to build a finalised system: a large set of transcribed speech for acoustic model training, determined in section 2.2, and a large LM, optimised as described in section 5. The lay-out of our system is depicted in figure 3. All experiments performed with this

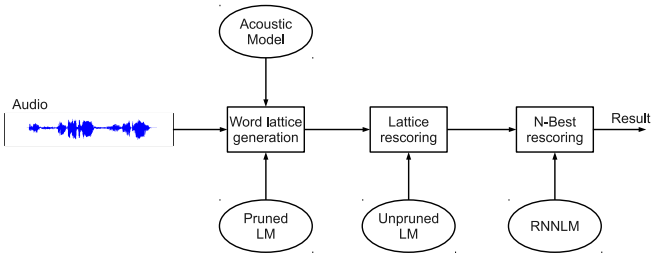


Figure 3: *A schematic overview of the adopted system.*

system, including the evaluations above and those that follow, have been performed using the KALDI Speech Recognition Toolkit [17]. For acoustic modelling, we first train up a GMM-HMM with 3000 context dependent states and 48000 Gaussians, using Speaker Adaptive Training (SAT), where fMLLR is used as the adaptation technique. In principle, it would be possible to assign multiple speeches to a single speaker, since the speaker's identity is given on the Europarl website. This only applies, however, to directly recorded speeches, i.e. untranslated ones. When the speaker is an interpreter, there is no trivial way to ascertain his/her identity. Therefore, we have made the simplifying assumption that each speech in the training data comes from a unique speaker. A feed-forward deep neural network is then trained in a DNN-HMM hybryd configuration, similar to the one used in [18]. This DNN has 6 hidden layers, each containing 2048 nodes. The softmax output layer of this network produces posterior probabilities over the 3000 context-dependent states of the HMM. The input at each time t consists of a stacking of the features in the context window $[t-5, t-4, \ldots t, \ldots t+4, t+5]$. Except for the addition of speaker adaptation, the features in each frame are produced as explained in section 4. Since the IWSLT test set is provided without segmentation into utterances, one can either generate a segmentation automatically, or perform recognition on entire TED-talks without segmentation. For the automatic segmentation, we use a voice activity detection system trained on 70 hours of English conversational speech from the AMI Meetings Corpus [19]. Speech and silence frames are modelled with diagonal covariance GMMs. A minimum duration constraint of 50ms is applied to each segment. For the segmentationless recognition, we use the same technique

	dev2012	tst2013	tst2013\E06
manual segment	27.02	35.27	29.18
auto segment	✗	39.28	33.58
no segment	28.16	36.21	30.24

Table 4: *The resulting WER's in % for several different evaluation sets, both when they are manually segmented, automatically segmented, or recognised in full (not segmented).*

as in [5], where we split an entire talk into overlapping segments, perform ASR on them, and dynamically merge the results into a single long recognition. In this case, segments are 40 seconds long and have an overlap of 20 seconds with each other. The results are listed in table 4. For the development set, no automatic segmentation was performed, since the manual segmentation was available for the official evaluation. There is one talk in the IWSLT test set, namely "E06_Nach-und-doch-so-Fern-Thomas-Mo", that is of very low quality. It has been recorded with a far-range microphone across a reverberant room, and contains quite a bit of non-speaker noise, e.g. coughing, rustling of paper and clothing, etc. Our system has not been designed to deal with such conditions, nor has it been tuned to them in any way, since the development set does not contain similar recordings. We therefore argue that this file unfairly skews the average test results. In tabel 4, the column "tst2013\E06" lists the results when this file is excluded from the evaluation. These error rates are more in line with those obtained on the dev set. The results in this table suggest that for TED talks, in the absence of a manual segmentation, a recognition performed on the whole talk is preferrable to using an automatically generated segmentation. We suspect, however, that this conclusion is fairly domain-specific. An automatic segmentation is essential for files with more music, jingles, applause, laughter, and other non-speaker noise.

7. Conclusion

We have presented the various components in the German ASR system, how they were set up, trained, and combined, to obtain accurate recognitions on the various data sets of the IWSLT evaluation task. Worthy of note is the acoustic model training, which was done almost entirely on publicly available data, without expert human transcriptions, using a lightly supervised training technique. Final evaluation on the unsegmented test set was performed in two different ways. Once with an automatically generated segmentation, and once without segmentation at all. It was found that, even though an oracle segmentation leads to optimal recognition results, avoiding segmentation altogether is preferrable to using an automatically generated one, when an oracle segmentation is not available.

8. References

[1] N. Braunschweiler, M. Gales, and S. Buchholz, "Lightly supervised recognition for automatic alignment of large coherent speech recordings," in *Proc. Interspeech*, September 2010, pp. 2222–2225.

[2] L. Lamel, J.-L. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech and Language*, vol. 16, no. 1, pp. 115–129, 2002.

[3] P. Placeway and J. Lafferty, "Cheating with imperfect transcripts," in *Proc. ICSLP*, vol. 4, 1996, pp. 2115–2118.

[4] P. Moreno and C. Alberti, "A factor automaton approach for the forced alignment of long speech recordings," in *Proc. ICASSP 2009.*, 2009, pp. 4869–4872.

[5] J. Driesen and S. Renals, "Lightly supervised automatic subtitling of weather forecasts," in *Proc. Automatic Speech Recognition and Understanding Workshop*, Olomouc, Czech Republic, December 2013.

[6] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proc. ACL*, Uppsala, Sweden, July 2010.

[7] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. Interspeech*, Makuhari, Japan, September 2010.

[8] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, 1998.

[9] Y. Huang, D. Yu, Y. Gong, and C. Liu, "Semi-supervised GMM and DNN acoustic model training with multi-system combination and confidence re-calibration," in *Proc. Interspeech*, Lyon, France, 2013.

[10] T. Schultz, "GlobalPhone: A multilingual speech and text database developed at karlsruhe university," in *Proc. Interspeech*, Denver, Colorado, USA, 2002.

[11] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *Proc. ICASSP*, Vancouver, Canada, May 2013.

[12] "The website of the european parliament." [Online]. Available: http://europarl.europa.eu

[13] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, May 2008.

[14] A. Katsamanis, M. Black, P. Georgiou, L. Goldstein, and S. Narayanan, "Sailalign: Robust long speech-text alignment," in *Proc. of Workshop on New Tools and Methods for Very-Large Scale Phonetics Research*, January 2011.

[15] "Iterative language model estimation: Efficient data structure & algorithms," in *Proc. Interspeech*, Brisbane, Australia, September 2008.

[16] S. F. Chen, , and J. Goodman, "An empirical study of smoothing techniques for language modeling." in *Proc. ACL*, Santa Cruz, USA, June 1996.

[17] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *Proc. ASRU*, Big Island, Hawaii, US, December 2011.

[18] P. Bell, P. Swietojanski, and S. Renals, "Multi-level adaptive networks in tandem and hybrid ASR systems," in *Proc. ICASSP*, Vancouver, Canada, May 2013.

[19] T. Hain, L. Burget, J. Dines, P. Garner, F. Grezl, A. Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan, "Transcribing meetings with the AMIDA systems," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 486–498, 2012.

A system for automatic broadcast news summarisation, geolocation and translation

Peter Bell, Catherine Lai, Clare Llewellyn, Alexandra Birch, Mark Sinclair

Centre for Speech Technology Research, University of Edinburgh, Edinburgh EH8 9AB, UK
{peter.bell, c.lai, a.birch, mark.sinclair}@ed.ac.uk, C.A.Llewellyn@sms.ed.ac.uk

Abstract

An increasing amount of news content is produced in audio-video form every day. To effectively analyse and monitoring this multilingual data stream, we require methods to extract and present audio content in accessible ways. In this paper, we describe an end-to-end system for processing and browsing audio news data. This fully automated system brings together our recent research on audio scene analysis, speech recognition, summarisation, named entity detection, geolocation, and machine translation. The graphical interface allows users to visualise the distribution of news content by entity names and story location. Browsing of news events is facilitated through extractive summaries and the ability to view transcripts in multiple languages.
Index Terms: multimedia archives, ASR, summarisation, named entity detection, geolocation, machine translation.

1. Introduction

The global media industry produces many thousands of hours of audio and video news content on a daily basis. A challenge for the industry is the need to balance the desire of news consumers for relevant localised content, against the objective of selecting global news items of importance to people located far from the source of the story. This task is highly labour-intensive. The former requires a high volume of news content to be collected and precisely targeted, a particular challenge where the consumers speak a minority language or dialect. The latter demands expensive multilingual media monitoring operations, which all but the largest media organisations struggle to afford.

This "Show & Tell" proposal presents a proof-of-concept automatic system for analysing news content, targeting it to potentially interested audiences on a geographic basis, and making it available in appropriate languages. Our hope is that a fully-developed version of this system could help news organisations operate more efficiently on a global scale. The system integrates our recent research outputs from a range of speech and natural language processing disciplines: audio scene analysis, automatic speech recognition, extractive summarisation, entity detection and spoken language translation. We describe these elements of the processing pipeline in more detail in the following sections.

In its current implementation, the system processes incoming broadcast media in an offline manner. Transcription, summarisation, entity detection and translation are performed for each story and imported into a web-based interactive user interface shown in Figure 1. We do not currently realign the translated content with the original media.

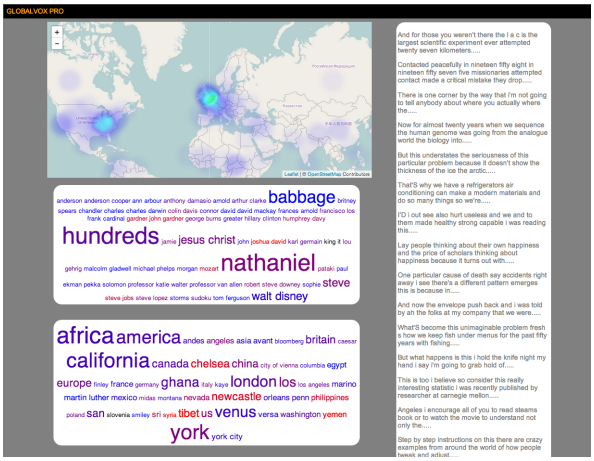


Figure 1: User interface

2. Speech recognition

Incoming video files are processed to extract the audio stream in wave format using FFmpeg[1]. After audio scene analysis using an unsupervised clustering technique, automatic speech recognition (ASR) is applied per item in an offline manner, allowing us to perform adaptation to each news story in a 2-pass configuration. The system uses sequence-trained deep neural networks in a hybrid configuration, following [1]. The models are trained on around 600 hours of multi-genre broadcast data from the British Broadcasting Corporation (BBC) taken from the training set defined for the 2015 MGB Challenge[2]. For ease of demonstration, the original news stories are all in English; we later translate from English to a range of target languages. In a final deployed system we would expect to include multiple source language options. The BBC news videos we use for the demonstration are pre-segmented by hand into discrete stories as part of the transmission process, so in this case we do not need to perform automatic topic segmentation, although clearly this could be required in a future version.

3. Summarisation

We use extractive summarisation techniques to select representative quotes from news stories. In addition to lexical features based on ASR output, the summariser uses prosodic features to

This work was supported by the European Union under the FP7 projects inEvent (grant agreement 287872) and EU-Bridge (grant agreement 287658), and by EPSRC Programme Grant grant EP/I031022/1, *Natural Speech Technology*.

[1] http://www.ffmpeg.org
[2] http://www.mgb-challenge.org

rank utterances, estimating the probability of their appearing in an extractive summary via logistic regression. Previous work has suggested that the use of non-lexical features, such as word and utterance level prosody, can help ameliorate problems with ASR. The models were trained on manually annotated AMI meeting data. We found that using prosodically augmented lexical features provided the best performance on held out meeting data [2]. Even though the summariser was designed for multiparty dialogue, further experiments have shown it extends well to other spoken genres. In a pairwise preference test, we found that quotes ranked higher by the summariser were also selected as more representative by human subjects a significant majority of the time.

4. Named entity detection and geolocation

Our system employs methods to allow for searchable aggregation of summarised speech. The summariser provides the top 10 ranked utterances for further processing. However, text extracted from the spoken news reports lacks punctuation and capitalisation. To allow the use of richer punctuation and capitalisation features, we add a machine translation based punctuation module to the pipeline, described below. In addition, common names and places are capitalised after lookup in various people/place lexicons.

To identify and visualise items of interest, we use the Edinburgh Informatics information extraction tools [3, 4]. These are well established tools that process text to identify entity names and provide geographic coordinates for locations. The named entity recognition tool identifies word sequences as people or place name entities via a rule-based method that takes into account information about part-of-speech, capitalisation, local context and lexicon look-up. Places are then georesolved – the names are looked up in a geographic gazetteer and possible interpretations are returned. These are ranked in order to assign a specific latitude and longitude value. Entities are represented in the interface using wordles with the size of each entity reflecting the frequency; places are represented as a density map (Open Street Map and Leaflet are used[3]).

In order to gauge the general opinion towards each entity, sentiment analysis was performed on the sentences containing those entities using the rule based sentiment analyser Vader [5]. This tool is adapted for use with social media data and is therefore ideal for use with speech segments which can be presented using less formal language. This gave positive negative and neutral scores for each entity which were then represented using colour. Once processed the data was stored in a Mongo noSQL database[4]. The flexible schema of the database allows us to assign the entities counts and sentiment scores to each entity, enabling the recalculation of scores for various document sets.

5. Spoken language translation

Statistical machine translation of transcripts of BBC News reports requires special handling. The ASR output contains errors both in the words recognised and in sentence segmentations. It also lacks punctuation and capitalisation. We therefore used phrase-based machine translation which is more robust than structured syntax-based translation models. We trained two translation models: one which translates from unpunctuated ASR English output to punctuated English output, and one which translates from standard English text to the target language text. Casing was handled by a re-caser, which applies case to words according to their most common case in the training corpus. The training corpora used were Europarl [6], News Commentary, TED [7], and Commoncrawl [8]. We used the Moses SMT toolkit [9] with standard settings, including the use of 5-gram language models.

[3]http://www.leafletjs.com

[4]http://mongodb.org

6. Conclusion and Future Work

Feedback from initial demonstrations of our system to journalists was extremely positive, indicating that this system would be valuable in analysing large volumes of multilingual news content. While our system demonstrates the potential of existing speech and language technologies, it also highlights areas that need attention when building speech based end-to-end systems. For example, we found that segmentation of the speech stream can have a substantial effect on the usability and readability of transcribed speech through the pipeline. Improving segmentation can also improve the quality of extracted summaries and automatic translation. Thus, optimising initial audio segmentation is vital for overall system robustness. Downstream language processing trained on written texts often assume more information than is available from raw ASR output, e.g. punctuation. The problem is exacerbated by the frequency of non-sentential utterances in speech. Besides improving the links between our current modules, we intend to extend our system by including more higher level analysis such as topic and dialogue act detection. We also hope to make more use of audio event detection techniques for determining structure in longer broadcasts, for example detecting topic change indicators such as music.

7. Acknowledgements

This work was developed as part of a BBC News Labs *newsHACK* event. We are grateful to the BBC for their support.

8. References

[1] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc Interspeech*, 2013.

[2] C. Lai and S. Renals, "Incorporating lexical and prosodic information at different levels for meeting summarization," in *Proc. Interspeech 2014*, 2014.

[3] C. Grover, S. Givon, R. Tobin, and J. Ball, "Named entity recognition for digitised historical texts," in *Proc. LREC*, 2008.

[4] C. Grover and R. Tobin, "Rule-based chunking and reusability," in *Proc. LREC*, 2006.

[5] C. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in *Proc. ICWSM*, 2014.

[6] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *MT Summit*, vol. 5, 2005.

[7] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proc EAMT*, 2012.

[8] J. Smith, H. Saint-Amand, M. Plamada, P. Koehn, C. Callison-Burch, and A. Lopez, "Dirt cheap web-scale parallel text from the common crawl," in *Proc ACL*, 2013.

[9] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source

Appendix C

Conference Papers - Pending Submission

EXPLOITING ATYPICAL SEGMENT DURATION DISTRIBUTIONS FOR REMOVING SURPLUS CLUSTERS IN AGGLOMERATIVE CLUSTERING BASED SPEAKER DIARIZATION

Mark Sinclair

The Centre for Speech Technology Research, The University of Edinburgh, UK

ABSTRACT

Often agglomerative speaker diarization algorithms terminate early resulting in an over-estimation of the number of clusters (speakers). These extra 'surplus' clusters typically have a low representation comprising only short, fragmented segments. Such segment duration distribution is atypical in comparison to that of an actual speaker. We propose a method which augments a typical state-of-the-art system with a post-processing step for removing such clusters and redistributing their data among other clusters with a more typical, speaker-like behaviour. We show that this method improves the estimation of the true number of speakers, halving the speaker number error as compared with our standard system, and also provides a significant improvement to the speaker attribution component of the Diarization Error Rate (DER).

Index Terms— Speaker Diarization, Agglomerative Clustering, Termination Criteria

1. INTRODUCTION

Speaker Diarization involves segmenting audio into speaker homogeneous regions and labelling regions from each individual speaker with a single label. Knowing both who spoke and when has useful applications and can form part of a rich transcription of speech. The task is challenging because it is generally performed without any a priori knowledge about the speakers present, not even how many speakers there are.

It is therefore important that speaker diarization systems both find the correct number of speakers and attribute the correct segments of speech to those speakers. However, the most common metric for describing speaker diarization system performance, the diarization error rate (DER), does not consider the number of speakers and instead focuses on the total time correctly attributed by the system. This means that many systems are able to attain good DER scores without necessarily getting the right number of speakers [1]. In fact, we often observe that over-estimating the number of speakers can actually result in higher DER performance [2], however this is only optimising for one part of the speaker diarization task and does not represent a well-rounded solution.

Funded by an EPSRC studentship with tare his work partially funded by Quorate Technology and SICSA

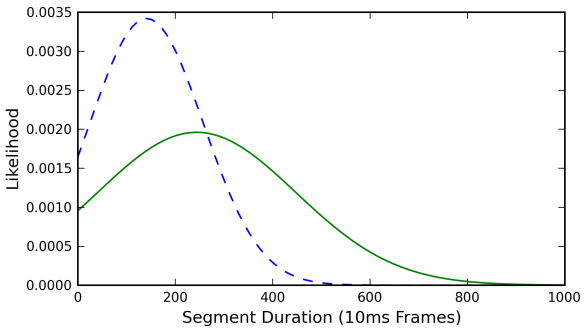


Fig. 1. Segment length distribution: average for oracle speaker segmentation (solid); average of least speaker-pure clusters after standard system termination (dashed).

2. MOTIVATION

We found that our standard system was indeed regularly over-estimating the number of speakers (in 19 out of 24 cases for our test set presented in Section 5). When we analysed this behaviour we found that there were typically clusters which for the most part represented the target speakers and that the additional clusters often represented short, fragmented segments that are atypical of a normal speaker.

We confirmed this by comparing the segment duration distribution of all oracle speech segments with those of the clusters with the lowest speaker-purity after termination of the standard system, using the test data set from Section 5. Speaker purity shows how well a cluster describes one speaker exclusively, whereby an perfect solution would have a number of clusters equal to the number of speakers each with 100% speaker purity. As such, clusters with a low speaker purity are not adhering to the task and are ideal candidates for removal. Figure 1 compares these two distributions and it is quite clear to see that the clusters with low speaker purity have a significantly different distribution which has a lower, sharper peak. As such we devised the method described in Section 4 to exploit this behaviour by removing clusters that have this distinctive distribution.

3. SYSTEM DESCRIPTION

The system uses a GMM-HMM framework whereby 16 clusters (states) are initialised with speech data by dividing the speech frames uniformly into 32 parts and using 2 parts (from different points in the data) to initialise each of the 16 GMMs. Given these models, the system then segments all speech using the Viterbi algorithm. Segments which are shorter than a given minimum duration of 250ms forego the Viterbi segmentation and are simply attributed to the cluster which awards the highest likelihood. After segmentation, the models are retrained, and this is followed by a clustering step in which the most similar clusters are merged. In order to choose which clusters to merge we calculate a normalised cross-likelihood ratio (NCLR) for every pair and the those with the lowest distance become the candidate pair. This is not enough information to decide when to terminate (without perhaps introducing a hyper-parameter to threshold the distance), so the Bayesian Information Criterion (BIC) score is then calculated for the candidate pair only. The putative merged model in this case has a complexity (i.e. number of model parameters) equal to the sum of the complexity of the models being merged, which means that a penalty factor parameter is not required. By calculating the merged model for only the top candidate pair we save considerable computational cost as we are required to train only one merged model for each iteration instead of a merged model for every exclusive pair of clusters as in conventional methods. A similar concept was proposed in [3] but this was applied to an older progressive change point detection method, whereby here we apply it to the GMM-HMM agglomerative clustering method that is common among state-of-the-art systems.

We also compare our results with the IDIAP Speaker Diarization Toolkit[1] [4, 5, 6] which utilises the information bottleneck method [7] to achieve state-of-the-art speaker diarization performance. However, unlike much of the focus of this work, we do not exploit multiple channels or microphone arrays and instead only consider strictly Single Distant Microphone (SDM) scenarios where only one channel is available.

4. PROPOSED METHOD

We first allow the standard system to function as normal. This involves the initialisation step which uniformly distributes the data among the initial clusters, we then do 3 iterations of segmenting and retraining these clusters in order that they move away from a uniform distribution. This is followed by the agglomerative clustering loop which terminates according the BIC criteria. When the standard system has reached termination point we will then enter into the post-processing loop to check for clusters that have an atypically short, fragmented segment duration distribution and remove them. We do this by comparing between clusters using a distance metric to decide how 'typical' each cluster is with respect to the others.

The distance metric we used is shown in Equation 1, which has been inspired from work shown in [8]. The metric was originally devised to compare a pair of GMMs with a different number of mixtures. Here, we have adapted it to compute the distance between a weighted univariate Gaussian distribution representing one cluster and a grouping of the remaining weighted univariate Gaussian distributions representing the remaining clusters. This is analogous to comparing a GMM with one mixture against a GMM with a number of mixtures ≥ 1.

$$D(\mathcal{N}, c) = -\log\left[\frac{2w_c \sum_{i=0, i\neq c}^{C} w_i \sqrt{\frac{(\sigma_c^{-2}+\sigma_i^{-2})^{-1}}{e^{k_{ci}}\sigma_c^2\sigma_i^2}}}{w_c^2\sqrt{\frac{2}{\sigma_c^2}} + \sum_{\substack{i,j=0 \\ i,j\neq c}}^{C} w_i w_j \sqrt{\frac{(\sigma_i^{-2}+\sigma_j^{-2})^{-1}}{e^{k_{ij}}\sigma_i^2\sigma_j^2}}}\right]$$

$$k_{ij} = \frac{\mu_i(\mu_i-\mu_j)}{\sigma_i^2} + \frac{\mu_j(\mu_j-\mu_i)}{\sigma_j^2} \quad (1)$$

The parameter set $\mathcal{N}$ describes a GMM with C mixture components, each of which represents the segment duration distribution for each cluster with mean μ, variance σ and weight w. The mixture weights of these GMMs are derived from the number of frames of data that each cluster represents, normalised by the total amount of frames for all clusters. This ensures that the most dominant clusters influence the shape of the 'typical' segment duration distribution we are trying to optimise i.e. we may yet have many clusters with lots of short segments and we do not want them to have equal weight. The index c represents the cluster we want to measure.

We calculated such a distance for all clusters then we check if the distance is above a prescribed threshold $\mathcal{T}$ in order to consider the cluster as a candidate for deletion. This threshold constitutes a hyper-parameter that must be set by the user. As the distance metric is symmetric and positive, we also check if the mean of the candidate is less than the global mean of all clusters. If it is greater or equal we do not consider it as a candidate as we do not want to remove atypically large clusters – often such cases can represent speakers who have made long monologues and their removal would impact DER significantly. If several candidates still remain, we use the one which has the greatest distance.

If an atypical cluster is found, it will be deleted and the the data is re-segmented with the remaining clusters. Those clusters are then retrained on this new segmentation, thus redistributing the data which belonged to the removed cluster. This process repeats until either only one cluster remains or there are no longer any more candidates for removal according the the criteria described.

Algorithm1 outlines this process in full.

[1] https://www.idiap.ch/scientific-research/resources/speaker-diarization-toolkit

Initialisation;
Init. 16 GMMs with uniform data split;
for $i = 0$ **to** $initresegs$ **do**
  Viterbi re-segment;
  re-train clusters;
Main Agglomerative Clustering Loop;
while $C > 1$ **do**
  repeat
    for *all exclusive cluster pairs* **do**
      calculate NCLR
    calc. BIC for candidate pair with min. NCLR;
    if $BIC < 0$ **then**
      merge candidate pair;
      Viterbi re-segment; re-train clusters;
  until $BIC > 0$;
Atypical Cluster Removal Loop;
while $C > 1$ **do**
  repeat
    for $i = 0$ **to** C **do**
      calc. mean and variance of segment
      durations and add to parameter set $\mathcal{N}$;
    for $i = 0$ **to** C **do**
      calc. $D(\mathcal{N}, i)$;
      if $\mu_i < \mu_{global}$ **and** $D(\mathcal{N}, i) < \mathcal{T}$ **then**
        add cluster to candidates;
    if *candidate(s) exist* **then**
      choose cluster with max dist. D;
      delete cluster;
      Viterbi re-segment;
      re-train clusters;
    else
      terminate
  until $maxdist < threshold$;

Algorithm 1: Illustration of the procedure for our system

5. EXPERIMENTAL SETUP

5.1. Data and System Parameters

As is consistent with much of speaker diarization literature, we make use of data from the NIST Rich Transcription (RT) campaigns [9]. These ran annually between 2002 and 2009, focusing on promoting Metadata Extraction (MDE) for speech. We evaluate our methods on the data from RT06, RT07 and RT09 campaigns as these focused on meetings with between 4 and 11 participants. As all the meetings are inherently of the same nature we simply group them all together and present results that are averaged across all.

For reference we used supervised segmentation based on forced-aligned ASR on the available headset channels as described in [10]. This has proved to be more accurate and consistent than the manual segmentations provided by NIST. As we have also previously shown [2], if Speech Activity Detection (SAD) is generally good, the errors do not propagate on to significantly affect speaker error during the clustering process. As such, we use the reference segmentation as SAD input to all systems.

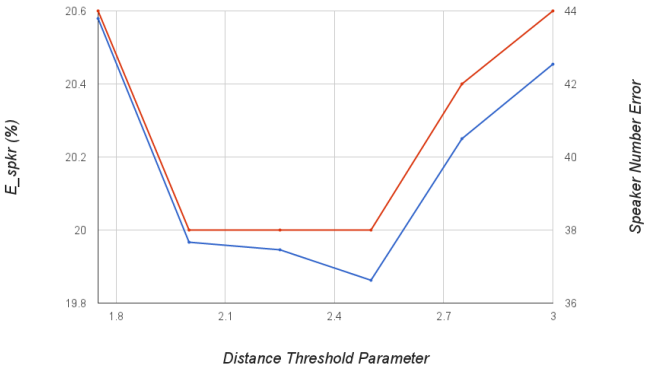


Fig. 2. Segment length distribution: average for oracle speaker segmentation (solid); average of least speaker-pure clusters after standard system termination (dashed).

As we are concerned with only the SDM scenario, we choose a single channel from one of the microphone arrays to provide the audio input. We then extract 19 static MFCCs every 10ms from a 30ms analysis window. While not essential for diarization to function, the features are then mean and variance normalised.

The IDIAP toolkit is used with all the default settings.

5.2. Atypical Cluster Selection Threshold

In order to tune the hyper-parameter used to determine when to stop deleting atypical clusters, we ran the full system without a parameter and forced it to delete clusters until the true number of speakers was met. We could then observe what the typical distances were at this point. We found that they were all typically less than 2.0 and that the previous iterations were typically greater than 3.0. As a result we heuristically tried various thresholds between these 2 values and found that performance was fairly similar across this range but would deteriorate significantly below 2.0 (removing too many clusters) or tend towards the standard system performance above 3.0 (not removing any/enough clusters). In the end we chose to set the threshold parameter to be 2.5 as this had the best combination of speaker attribution error and speaker number error. Results from sweeping this parameter can be seen in Figure 2.

6. RESULTS

6.1. Number of Speakers

In order to evaluate the performance with regards to estimating the number of speakers, we consider the sum of the absolute difference between the hypothesised number of speakers

and the true number of speakers for each meeting (the total number of speakers for all meetings is 119). This is outlined in Eq.2 where M is the number of meetings, R_i is the number of speakers in the reference and H_i is the number of hypothesised speakers.

$$E_{SpkrNum} = \sum_{i=0}^{M} |R_i - H_i| \quad (2)$$

We looked at the sum of the absolute difference between the true number of speakers and that hypothesised by the system. We expected the removal of atypical clusters to tend towards a solution whereby the number of clusters is closer to the true number of speakers. Such behaviour can be observed in our resultsm, as Table 6.2 shows a drop from our standard system with an error of 79 to 38, which more than halves the error.

We can also observe in Figure 3 that the distribution of the difference between the hypothesised number of speakers and the true number of speakers after removing atypical clusters is more centred around zero as compared with the standard system. This shows that the clustering is terminating at a point much closer to the actual number of speakers.

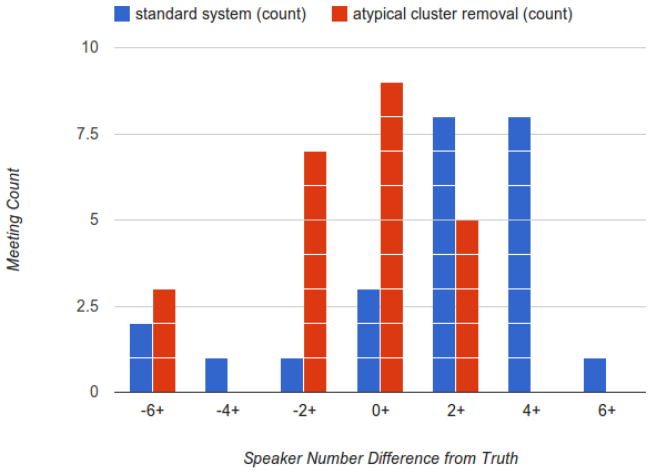


Fig. 3. Distribution of Speaker Number Difference

6.2. Speaker Error Rate

The main metric for system evaluation is the Diarization Error Rate (DER) which is a sum of three contributing factors as shown in Eq.3: speaker misclassification E_{SpkErr}, false alarm E_{FA} (speaker attributed when no speech exists) and missed speech E_{Miss} (speaker not attributed when speech exists).

$$DER = E_{Spkr} + E_{FA} + E_{Miss} \quad (3)$$

	$E_{SpkrNum}$	$E_{SpkErr}(\%)$
IDIAP	48	22.25
Standard System	79	21.64
+ atypical cluster removal ($\mathcal{T} = 2.5$)	38	19.86

Table 1. Results for Speaker Number Error and Average Speaker Attribution Error

As we are using the same oracle SAD for every experiment only E_{SpkErr} will change as this is the part that is influenced by the clustering process and as such we only present results for this error component. There will no false alarms ($E_{FA} = 0$). However, there is a small amount of average missed speech error due to overlapping speech which none of the systems account for ($E_{Miss} = 5.04\%$).

Often we observe that simply getting the right number of speakers may not result in an improved overall clustering. This can be due to incorrectly merging clusters which have a high purity. In such cases can be better to sacrifice the right number of speakers to at least maintain purer clusters. As show in Table 6.2 however, we can see that not only have we improved the number of speakers estimation but we have also improved the speaker error rate.

7. CONCLUSION

We have presented a post-processing method which aims to deal with the issue of surplus clusters that many speaker diariaztion systems suffer from. We have shown that this method improves both the estimated number of speakers as well as the overall speaker error rate. These improvements also provided a better result than a baseline state-of-the-art system, however, it is worth noting that as it is a post-processing step, this method could also be applied to any system in a supplementary manner which may lead to performance gains.

Currently, the method does rely on a tuned hyper-parameter for selecting the ideal threshold for deciding deletion candidates. Further work may lead to the automation of this parameter. As it is uncommon for the systems we used, we neither considered the case of a system which consistently estimates the number of speakers well nor one which underestimates the number of speakers, whereby the application of this method could result in significantly fewer clusters than the true number of speakers. However, in such a scenario the same kind of segment duration distribution exploitation could potentially be used to 'roll back' clustering to an iteration which had a distribution with a better 'speaker-typicality'.

8. REFERENCES

[1] E. Zwyssig, S. Renals, and M. Lincoln, "Determining the number of speakers in a meeting using microphone array features," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, 2012, pp. 4765–4768.

[2] Mark Sinclair and Simon King, "Where are the challenges in speaker diarization?," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, Vancouver, British Columbia, USA, May 2013.

[3] Viet-Bac Le, Odile Mella, and Dominique Fohr, "Speaker diarization using normalized cross likelihood ratio.," in *INTERSPEECH*. 2007, pp. 1869–1872, ISCA.

[4] Deepu Vijayasenan, Fabio Valente, and Hervé Bourlard, "An Information Theoretic Combination of MFCC and TDOA Features for Speaker Diarization.," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 19, no. 2, pp. 431–438, 2011.

[5] Deepu Vijayasenan, Fabio Valente, and Hervé Bourlard, "An Information Theoretic Approach to Speaker Diarization of Meeting Data.," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 17, no. 7, pp. 1382–1393, 2009.

[6] Deepu Vijayasenan, Fabio Valente, and Hervé Bourlard, "Multistream speaker diarization of meetings recordings beyond MFCC and TDOA features," *Speech Commun.*, vol. 54, no. 1, pp. 55–67, Jan. 2012.

[7] Naftali Tishby, Fernando C. Pereira, and William Bialek, "The information bottleneck method," in *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, 1999, pp. 368–377.

[8] G. Sfikas, Constantinos Constantinopoulos, Aristidis Likas, and Nikolas P. Galatsanos, "An Analytic Distance Metric for Gaussian Mixture Models with Application in Image Retrieval.," in *ICANN (2)*, Wlodzislaw Duch, Janusz Kacprzyk, Erkki Oja, and Slawomir Zadrozny, Eds. 2005, vol. 3697 of *Lecture Notes in Computer Science*, pp. 835–840, Springer.

[9] Jonathan G. Fiscus, Jerome Ajot, and John S. Garofolo, "The Rich Transcription 2007 Meeting Recognition Evaluation.," in *CLEAR*, Rainer Stiefelhagen, Rachel Bowers, and Jonathan G. Fiscus, Eds. 2007, vol. 4625 of *Lecture Notes in Computer Science*, pp. 373–389, Springer.

[10] Xavier Anguera, Chuck Wooters, and José M. Pardo, "Robust Speaker Diarization for Meetings: ICSI RT06S Meetings Evaluation System.," in *MLMI*, Steve Renals, Samy Bengio, and Jonathan G. Fiscus, Eds. 2006, vol. 4299 of *Lecture Notes in Computer Science*, pp. 346–358, Springer.

Bibliography

(2006). *Spring 2006 (RT-06S) Rich Transcription Meeting Recognition Evaluation Plan*. NIST.

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 Hansen, J. H. L. and Pellom, B. L., editors, *INTERSPEECH*. ISCA.

Ajmera, J. and Wooters, C. (2003). A robust speaker clustering algorithm. In *Proc. IEEE Workshop Automatic Speech Recognition and Understanding ASRU '03*, pages 411–416.

Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., and Mohri, M. (2007). Openfst: A general and efficient weighted finite-state transducer library. In *Implementation and Application of Automata*, pages 11–23. Springer.

Anguera, X., Woofers, C., and Hernando, J. (2005). Speaker diarization for multi-party meetings using acoustic fusion. In *Proc. IEEE Workshop Automatic Speech Recognition and Understanding*, pages 426–431.

Anguera, X., Woofers, C., and Hernando, J. (2006a). Purity algorithms for speaker diarization of meetings data. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 1, pages I–I. IEEE.

Anguera, X., Wooters, C., and Pardo, J. M. (2006b). Robust Speaker Diarization for Meetings: ICSI RT06s evaluation system. *Interspeech 2006*.

Bell, P., Gales, M., Hain, T., Kilgour, J., Lanchantin, P., Liu, X., McParland, A., Renals, S., Saz, O., Webster, M., et al. (2015a). The mgb challenge: Evaluating multi–genre broadcast media transcription. In *Proceedings of IEEE workshop on Automatic Speech Recognition and Understanding, Scottsdale, AZ.*

Bell, P., Lai, C., Llewellyn, C., Birch, A., and Sinclair, M. (2015b). *A system for automatic broadcast news summarisation, geolocation and translation*. Date of Acceptance: 01/06/2015.

Bell, P., McInnes, F., Gangireddy, S. R., Sinclair, M., Birch, A., and Renals, S. (2013). The UEDIN English ASR System for the IWSLT 2013 Evaluation. *IWSLT, Heidelberg, Germany*.

Birch, A., Durrani, N., and Koehn, P. (2013). Edinburgh SLT and MT System Description for the IWSLT 2013 Evaluation. In *Proceedings of the 10th International Workshop on Spoken Language Translation*, pages 40–48, Heidelberg, Germany.

Boakye, K., Trueba-Hornero, B., Vinyals, O., and Friedland, G. (2008). Overlapped speech detection for improved speaker diarization in multiparty meetings. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing ICASSP 2008*, pages 4353–4356.

Boyd, I. and Freeman, D. K. (1994). Voice activity detection. US Patent 5,276,765.

Bozonnet, S., Evans, N. W., and Fredouille, C. (2010). The lia-eurecom rt'09 speaker diarization system: enhancements in speaker modelling and cluster purification. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 4958–4961. IEEE.

Cettolo, M. F. M., Bentivogli, L., Paul, M., and Stüker, S. (2012). Overview of the IWSLT 2012 evaluation campaign. *Proceedings IWSLT 2012*.

Cho, E., Niehues, J., and Waibel, A. (2012). Segmentation and punctuation prediction in speech language translation using a monolingual translation system. In *IWSLT*, pages 252–259.

Clement, P., Bazillon, T., and Fredouille, C. (2011). Speaker diarization of heterogeneous web video files: A preliminary study. In *Proc. IEEE Int Acoustics, Speech and Signal Processing (ICASSP) Conf*, pages 4432–4435.

Dahl, G. E., Yu, D., Deng, L., and Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):30–42.

Driesen, J., Bell, P., Sinclair, M., and Renals, S. (2013). Description of the uedin system for german asr.

Evans, N., Bozonnet, S., Wang, D., Fredouille, C., and Troncy, R. (2012). A comparative study of bottom-up and top-down approaches to speaker diarization. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(2):382–392.

Eyben, F., Weninger, F., Squartini, S., and Schuller, B. (2013). Real-life voice activity detection with lstm recurrent neural networks and an application to hollywood movies. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 483–487. IEEE.

Friedland, G., Janin, A., Imseng, D., Anguera, X., Gottlieb, L., Huijbregts, M., Knox, M., and Vinyals, O. (2011). The ICSI RT-09 Speaker Diarization System. *IEEE Transactions on Audio, Speech, and Language Processing*, (99). Early Access.

Furui, S. (1986). Speaker-independent isolated word recognition using dynamic features of speech spectrum. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 34(1):52–59.

Gehring, J., Miao, Y., Metze, F., and Waibel, A. (2013). Extracting deep bottleneck features using stacked auto-encoders. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3377–3381. IEEE.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256.

Graciarena, M., Alwan, A., Ellis, D., Franco, H., Ferrer, L., Hansen, J. H., Janin, A., Lee, B. S., Lei, Y., Mitra, V., et al. (2013). All for one: feature combination for highly channel-degraded speech activity detection. In *INTERSPEECH*, pages 709–713.

Gravano, A., Jansche, M., and Bacchiani, M. (2009). Restoring punctuation and capitalization in transcribed speech. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 4741–4744. IEEE.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE.

Hanson, B. and Applebaum, T. (1990). Robust speaker-independent word recognition using static, dynamic and acceleration features: Experiments with lombard and noisy speech. In *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, pages 857–860. IEEE.

Hassan, H., Ma, Y., and Way, A. (2007). MaTrEx: the DCU machine translation system for IWSLT 2007.

Hermansky, H. (1990). Perceptual linear predictive (plp) analysis of speech. *the Journal of the Acoustical Society of America*, 87(4):1738–1752.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97.

Huijbregts, M. and Wooters, C. (2007). The blame game: performance analysis of speaker diarization system components. In *INTERSPEECH*, pages 1857–1860. ISCA.

Huijbregts, M. A. H. (2008). Segmentation, diarization and speech transcription: surprise data unraveled.

Hunt, M. J. (1990). Figures of merit for assessing connected-word recognisers. *Speech Communication*, 9(4):329–336.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the Association for Computational Linguistics Companion Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Larochelle, H., Bengio, Y., Louradour, J., and Lamblin, P. (2009). Exploring strategies for training deep neural networks. *The Journal of Machine Learning Research*, 10:1–40.

Luebke, D. (2008). Cuda: Scalable parallel programming for high-performance scientific computing. In *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 836–838. IEEE.

Matusov, E., Hillard, D., Magimai-Doss, M., Hakkani-Tür, D. Z., Ostendorf, M., and Ney, H. (2007). Improving speech translation with automatic boundary prediction. In *INTERSPEECH*, volume 7, pages 2449–2452.

Matusov, E., Mauser, A., and Ney, H. (2006). Automatic sentence segmentation and punctuation prediction for spoken language translation. In *IWSLT*, pages 158–165.

Mccowan, I., Lathoud, G., Lincoln, M., Lisowska, A., Post, W., Reidsma, D., and Wellner, P. (2005). The AMI Meeting Corpus.

Milner, B. (2002). A comparison of front-end configurations for robust speech recognition. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 1, pages I–797. IEEE.

Mirghafori, N. and Wooters, C. (2006). Nuts and Flakes: a Study of Data Characteristics in Speaker Diarization. In *Proc. IEEE Int Acoustics, Speech and Signal Processing Conf. ICASSP 2006*, volume 1.

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*, 20(2):356–370.

Neubig, G. (2013). Travatar: A Forest-to-String Machine Translation Engine based on Tree Transducers. In *Proceedings of the ACL Demonstration Track*, Sofia, Bulgaria.

Nguyen, T. H., Chng, E., and Li, H. (2008). T-test distance and clustering criterion for speaker diarization. In *INTERSPEECH*, pages 36–39. ISCA.

Ostendorf, M., Favre, B., Grishman, R., Hakkani-Tur, D., Harper, M., Hillard, D., Hirschberg, J., Ji, H., Kahn, J. G., Liu, Y., et al. (2008). Speech segmentation and spoken document processing. *Signal Processing Magazine, IEEE*, 25(3):59–69.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2001). Bleu: a Method for Automatic Evaluation of Machine Translation. Technical report, IBM Research Report.

Peitz, S., Freitag, M., Mauser, A., and Ney, H. (2011). Modeling punctuation prediction as machine translation. In *IWSLT*, pages 238–245.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K.

(2011). The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

Ramırez, J., Segura, J. C., Benıtez, C., De La Torre, A., and Rubio, A. (2004). Efficient voice activity detection algorithms using long-term speech information. *Speech communication*, 42(3):271–287.

Roark, B., Liu, Y., Harper, M., Stewart, R., Lease, M., Snover, M., Shafran, I., Dorr, B., Hale, J., Krasnyanskaya, A., et al. (2006). Reranking for sentence boundary detection in conversational speech. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 1, pages I–I. IEEE.

Ryant, N., Liberman, M., and Yuan, J. (2013). Speech activity detection on youtube using deep neural networks. In *INTERSPEECH*, pages 728–731.

Rybach, D., Gollan, C., Schluter, R., and Ney, H. (2009). Audio segmentation for speech recognition using segment features. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 4197–4200. IEEE.

Seltzer, M. L. and Droppo, J. (2013). Multi-task learning in deep neural networks for improved phoneme recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6965–6969. IEEE.

Seltzer, M. L., Yu, D., and Wang, Y. (2013). An investigation of deep neural networks for noise robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7398–7402. IEEE.

Sinclair, M., Bell, P., Birch, A., and McInnes, F. (2014). A semi-markov model for speech segmentation with an utterance-break prior.

Sinclair, M. and King, S. (2013). Where are the challenges in speaker diarization? In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, Vancouver, British Columbia, USA.

Solomonoff, A., Mielke, A., Schmidt, M., and Gish, H. (1998). Clustering speakers by their voices. In *ICASSP*, pages 757–760. IEEE.

Stolcke, A. and Shriberg, E. (1996). Automatic linguistic segmentation of conversational speech. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 2, pages 1005–1008. IEEE.

Stolcke, A., Shriberg, E., Bates, R. A., Ostendorf, M., Hakkani, D., Plauche, M., Tür, G., and Lu, Y. (1998). Automatic detection of sentence boundaries and disfluencies based on recognized words. In *ICSLP*.

Sun, H., Ma, B., Khine, S. Z. K., and Li, H. (2010). Speaker diarization system for RT07 and RT09 meeting room audio. In *ICASSP*, pages 4982–4985. IEEE.

Tishby, N., Pereira, F. C., and Bialek, W. (2000). The information bottleneck method. *arXiv preprint physics/0004057*.

Traunmüller, H. and Eriksson, A. (1995). The frequency range of the voice fundamental in the speech of male and female adults.

Tsiartas, A., Chaspari, T., Katsamanis, N., Ghosh, P. K., Li, M., Van Segbroeck, M., Potamianos, A., and Narayanan, S. (2013). Multi-band long-term signal variability features for robust voice activity detection. In *INTERSPEECH*, pages 718–722.

Tucker, R. (1992). Voice activity detection using a periodicity measure. *IEE Proceedings I (Communications, Speech and Vision)*, 139(4):377–380.

Ueffing, N., Bisani, M., and Vozila, P. (2013). Improved models for automatic punctuation prediction for spoken and written text. In *INTERSPEECH*, pages 3097–3101.

Vijayasenan, D. and Valente, F. (2012). Diartk: An open source toolkit for research in multistream speaker diarization and its application to meetings recordings. In *INTERSPEECH*.

Vijayasenan, D., Valente, F., and Bourlard, H. (2007). Agglomerative information bottleneck for speaker diarization of meetings data. In *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pages 250–255. IEEE.

Vijayasenan, D., Valente, F., and Bourlard, H. (2009). An information theoretic approach to speaker diarization of meeting data. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17(7):1382–1393.

Vijayasenan, D., Valente, F., and Motlícek, P. (2011). Multistream speaker diarization through Information Bottleneck system outputs combination. In *ICASSP*, pages 4420–4423. IEEE.

Wang, Y.-Y., Acero, A., and Chelba, C. (2003). Is word error rate a good indicator for spoken language understanding accuracy. In *Automatic Speech Recognition and Understanding, 2003. ASRU'03. 2003 IEEE Workshop on*, pages 577–582. IEEE.

Ye, J., Kobayashi, T., Murakawa, M., and Higuchi, T. (2013). Incremental acoustic subspace learning for voice activity detection using harmonicity-based features. In *INTERSPEECH*, pages 695–699.

Young, S. J., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. (2006). *The HTK Book Version 3.4*. Cambridge University Press.

Yu, D. and Seltzer, M. L. (2011). Improved bottleneck features using pretrained deep neural networks. In *INTERSPEECH*, volume 237, page 240.

Zhang, X.-L. and Wu, J. (2013). Deep belief networks based voice activity detection. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21(4):697–710.

Zwyssig, E. P. (2013). Speech processing using digital mems microphones.