TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

Speech Activity and Speaker Change Point Detection for Online Streams

Dissertation

Study programme: P2612 – Electrical Engineering and Informatics
Study branch: 2612V045 – Technical Cybernetics

Author: **Ing. Lukáš Matějů**
Supervisor: Ing. Petr Červa, Ph.D.

Liberec 2019

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Detekce řeči a změny mluvčího v online vysílání

Disertační práce

Studijní program: P2612 – Elektrotechnika a informatika
Studijní obor: 2612V045 – Technická kybernetika

Autor práce: **Ing. Lukáš Matějů**
Vedoucí práce: Ing. Petr Červa, Ph.D.

Liberec 2019

Declaration

I hereby certify I have been informed that my dissertation is fully governed by Act No. 121/2000 Coll., the Copyright Act, in particular Article 60 – School Work.

I acknowledge that the Technical University of Liberec (TUL) does not infringe my copyrights by using my dissertation for the TUL's internal purposes.

I am aware of my obligation to inform the TUL on having used or granted license to use the results of my dissertation; in such a case the TUL may require reimbursement of the costs incurred for creating the result up to their actual amount.

I have written my dissertation myself using the literature listed below and consulting it with my thesis supervisor and my tutor.

At the same time, I honestly declare that the texts of the printed version of my dissertation and of the electronic version uploaded into the IS STAG are identical.

11. 12. 2019 Ing. Lukáš Matějů

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Ing. Petr Červa, Ph.D., for his flawless guidance and continuous support of my Ph.D. study. Besides my supervisor, I am also grateful to Ing. Jindřich Žďánský, Ph.D., for his thorough insights and immense help. This thesis would not be possible without their assistance. Last but not least, I would like to thank my family, friends, and especially Dany for being there for me.

Abstract

Speech Activity and Speaker Change Point Detection for Online Streams

The main focus of this thesis lies on two closely interrelated tasks, speech activity detection and speaker change point detection, and their applications in online processing. These tasks commonly play a crucial role of speech preprocessors utilized in speech-processing applications, such as automatic speech recognition or speaker diarization. While their use in offline systems is extensively covered in literature, the number of published works focusing on online use is limited. This is unfortunate, as many speech-processing applications (e.g., monitoring systems) are required to be run in real time.

The thesis begins with a three-chapter opening part, where the first introductory chapter explains the basic concepts and outlines the practical use of both tasks. It is followed by a chapter, which reviews the current state of the art and lists the existing toolkits. That part is concluded by a chapter explaining the motivation behind this work and the practical use in monitoring systems; ultimately, this chapter sets the main goals of this thesis.

The next two chapters cover the theoretical background of both tasks. They present selected approaches relevant to this work (e.g., used for result comparisons) or focused on online processing.

The following chapter proposes the final speech activity detection approach for online use. Within this chapter, a detailed description of the development of this approach is available as well as its thorough experimental evaluation. This approach yields state-of-the-art results under low- and medium-noise conditions on the standardized QUT-NOISE-TIMIT corpus. It is also integrated into a monitoring system, where it supplements a speech recognition system.

The final speaker change point detection approach is proposed in the following chapter. It was designed in a series of consecutive experiments, which are extensively detailed in this chapter. An experimental evaluation of this approach on the COST278 database shows the performance of approaching the offline reference system while operating in online mode with low latency.

Finally, the last chapter summarizes all the results of this thesis.

Keywords: Deep Neural Networks, Online Streams, Speech Activity Detection, Speaker Change Point Detection, Weighted Finite-State Transducers.

Abstrakt

Detekce řeči a změny mluvčího v online vysílání

Disertační práce je věnována dvěma si blízkým řečovým úlohám a následně jejich použití v online prostředí. Konkrétně se jedná o úlohy detekce řeči a detekce změny mluvčího. Ty jsou často nedílnou součástí systémů pro zpracování řeči (např. pro diarizaci mluvčích nebo rozpoznávání řeči), kde slouží pro předzpracování akustického signálu. Obě úlohy jsou v literatuře velmi aktivním tématem, ale většina existujících prací je směřována primárně na offline využití. Nicméně právě online nasazení je nezbytné pro některé řečové aplikace, které musí fungovat v reálném čase (např. monitorovací systémy).

Úvodní část disertační práce je tvořena třemi kapitolami. V té první jsou vysvětleny základní pojmy a následně je nastíněno využití obou úloh. Druhá kapitola je věnována současnému poznání a je doplněna o přehled existujících nástrojů. Poslední kapitola se skládá z motivace a z praktického použití zmíněných úloh v monitorovacích systémech. V závěru úvodní části jsou stanoveny cíle práce.

Následující dvě kapitoly jsou věnovány teoretickým základům obou úloh. Představují vybrané přístupy, které jsou buď relevantní pro disertační práci (porovnání výsledků), nebo jsou zaměřené na použití v online prostředí.

V další kapitole je předložen finální přístup pro detekci řeči. Postupný návrh tohoto přístupu, společně s experimentálním vyhodnocením, je zde detailně rozebrán. Přístup dosahuje nejlepších výsledků na korpusu QUT-NOISE-TIMIT v podmínkách s nízkým a středním zašuměním. Přístup je také začleněn do monitorovacího systému, kde doplňuje svojí funkcionalitou rozpoznávač řeči.

Následující kapitola detailně představuje finální přístup pro detekci změny mluvčího. Ten byl navržen v rámci několika po sobě jdoucích experimentů, které tato kapitola také přibližuje. Výsledky získané na databázi COST278 se blíží výsledkům, kterých dosáhl referenční offline systém, ale předložený přístup jich docílil v online módu a to s nízkou latencí.

Výstupy disertační práce jsou shrnuty v závěrečné kapitole.

Klíčová slova: detekce řeči, detekce změny mluvčího, hluboké neuronové sítě, online vysílání, vážené konečné stavové převodníky.

Contents

List of Figures

List of Tables

List of Abbreviations

ASR	Automatic Speech Recognition
BIC	Bayesian Information Criterion
BTN	Bottleneck
CLC	Complete-Linkage Clustering
CNN	Convolutional Neural Network
CRF	Conditional Random Field
DNN	Deep Neural Network
DSR	Distributed Speech Recognition
FAR	False Alarm Rate
FBC	Filter Bank Coefficient
FER	Frame Error Rate
GLR	Generalized Likelihood Ratio
GMM	Gaussian Mixture Model
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
HTER	Half-Total Error Rate
KL	Kullback-Leibler
LL	Log-Likelihood
LLR	Log-Likelihood Ratio
LSP	Line Spectrum Pair
LSTM	Long Short-Term Memory
LTSD	Long-Term Spectral Divergence
LTSE	Long-Term Spectral Envelope
MFCC	Mel-Frequency Cepstral Coefficient
MR	Miss Rate
NN	Neural Network
PC	Percent Correct
RNN	Recurrent Neural Network
RTF	Real-Time Factor
SAD	Speech Activity Detection
SC	Speaker Clustering
SCP	Speaker Change Point
SNR	Signal-to-Noise Ratio
TDNN	Time Delay Neural Network
TV	Total Variability
TVR	Television and Radio
UBM	Universal Background Model
VAD	Voice Activity Detection
WAcc	Word Accuracy
WER	Word Error Rate
WFST	Weighted Finite-State Transducer

Introduction

Nowadays, an increasingly overwhelming amount of audio data is produced every day by various media streams (television, radio, etc.) as well as many other sources (e.g., the Internet). Unfortunately, most of this data lacks labels (annotations, tags) of any kind that would be useful for a wide range of applications; in this case, for speech processing. The aforementioned labels vary greatly; they can, e.g., include speech transcription, subtitles, translation, change of speaker, or name of the played song, to name a few. They can even carry time stamps, which can be further utilized for audio searching, indexing, or data retrieval. Speech Activity Detection (SAD; or closely related Voice Activity Detection [VAD]) and Speaker Change Point (SCP) detection (often called speaker segmentation) are among the tasks that can create such labels. The former is a task of identifying and labeling speech and non-speech segments in an utterance while the latter, for a given utterance, finds and labels changes between different speakers (i.e., it is a task of detecting exact moments when a change of speaker occurs). As their output, both of these tasks split the recording into segments (speech/non-speech or speaker-homogeneous) and provide start- and end- time stamps of these newly defined blocks.

In general, speech activity detection and speaker change point detection are closely interrelated tasks. As such, they form an integral preprocessing component of many speech processing applications including, e.g., speaker verification and identification, language, gender or emotion detection, audio indexing and retrieval, or automatic speech transcription. Specifically, in speech transcription, implementation of SAD can significantly speed up the processing as well as increase the overall performance as the non-speech segments are omitted from transcription. This is more beneficial for broadcast streams consisting of a lot of non-speech events (e.g., music stream radios). Finally, SAD usually plays the role of the preprocessor even for SCP detection, which is only run on obtained speech segments.

Speaker change point detection, in conjunction with Speaker Clustering (SC), results in a speaker diarization system. Speaker diarization focuses on answering the question "who spoke when?" (i.e., it breaks down the recording into speaker-homogeneous segments and clusters the segments according to the speaker's identity), and it can be further extended into speaker verification and identification systems. The research is driven by challenges held by the National Institute of Standards and Technology (NIST). Additionally, SCP detection can be employed for tasks such as rich transcription, dialog detection, speaker tracking, multi-speaker detection, and more. Lastly, the extracted speaker-homogeneous segments can also be used as training data for speaker-adaptive approaches to Automatic Speech Recognition (ASR).

The diverse applications of speech activity detection and speaker change point detection make both of these tasks popular research topics. Numerous research groups and research centers compete worldwide and propose novel approaches in pursuit of improving the state-of-the-art results. Challenges are also being held

quite regularly (e.g., segmentation task by NIST) to push the field even further. The popularity of these research topics can also be documented by large amounts of papers accepted at international conferences on signal/speech processing, such as Interspeech or International Conference on Acoustics, Speech and Signal Processing (ICASSP). With the recent boom in deep learning in mind, SAD and SCP detection attract more and more researchers every day, and much exciting work is being published every year.

The remainder of this thesis is organized as follows: In Chap. 1, a summary is presented of the state-of-the-art approaches to both speech activity detection and speaker change point detection. It is supplemented with a compendium of existing systems. Chapter 2 explains the overall motivation for this thesis as well as its practical use for author's lab (SpeechLab) at the Technical University of Liberec (TUL). Ultimately, it also sets the main goals. A detailed overview of selected approaches to SAD and SCP detection is given in Chap. 3 and 4, respectively. Chapters 5 and 6 describe in detail the experimental setup and the designing process from the initial stages of development to the final proposed SAD approach and SCP detection approach. All of these steps are supported by a diverse set of experiments. Portions of the original publications are reused and expanded upon here. Specifically, it is [1–3] for SAD and [4] for SCP detection, all published during the author's Ph.D. studies. Finally, the thesis is concluded in Chap. 7.

1 State of the Art

At present, speech activity detection and speaker change point detection are generally treated as machine learning tasks. Recently, deep learning has extensively been applied to both of these tasks to improve their performance, and subsequently the results achieved. Both of these tasks are usually performed in two consecutive phases: feature extraction and classification. Moreover, both can be run in an offline or online mode. In the former mode, no additional restrictions are applied, and low latency and real-time processing are not vital. However, they become crucial in the latter mode. Furthermore, an online decoder may only perform one left-to-right pass through the input data. These additional restrictions result in a limited amount of published work for online processing.

1.1 Speech (Voice) Activity Detection

As already stated above, the majority of the existing speech activity detection approaches operate in two subsequent phases: feature extraction and speech/non-speech classification.

In the former phase, the classic approaches for feature extraction utilize energy [5], zero-crossing rate [6] or auto-correlation function [7]. The family of more complex features, which have also been successfully applied, includes Mel-Frequency Cepstral Coefficients (MFCCs) [8, 9], multi-resolution cochleagram features [10], pitch related features [11], multi-band long-term signal variability features [12] or i-vectors [13]. Bottleneck (BTN) features extracted from Deep Neural Networks (DNNs) have also been proposed [14, 15]. In practice, various combinations of individual features are often used to achieve the best possible results (e.g., [16–18]).

In the latter phase, various classification algorithms can be used, such as support vector machines [19] or Gaussian Mixture Models (GMMs) [20–22]. In recent years, various deep neural network architectures have been frequently employed, including fully connected feed-forward DNNs [8, 23, 24], Convolutional Neural Networks (CNNs) [25, 26], dilated CNNs [27] or Recurrent Neural Networks (RNNs) [28–30]. More complex approaches, such as jointly trained DNNs [31], boosted DNNs [10], a combination of DNNs and CNNs [32] or a combination of augmented statistical noise suppression and CNNs [33], have also been proposed. Furthermore, an adaptive context attention model was suggested in [34]. The output from a given classifier can also be smoothed to further improve the accuracy of the detection. Over the years, various techniques, such as the Viterbi decoder [8] or Weighted Finite-State Transducers (WFSTs) [35], have been applied for this purpose.

Most of the previously mentioned works primarily aim at offline application, or the focus is not specified in the given publications. The limited amount of approaches developed namely for the online task include, for example, Conditional Random Fields (CRFs; with Viterbi decoder) [36] or accurate endpointing with expected pause duration [37]. An unsupervised approach to real-time VAD was in-

troduced in [38, 39]. Another approach in [40] utilizes short-term features. Recently, a causal voice activity detector based on DNNs has been suggested in [41]. In [42], an online speech activity detector using simultaneously trained neural networks is shown. Finally, the authors of [43] studied the impact of lowering the representation precision of DNN weights and neurons on the accuracy and delay of VAD.

In general, the majority of the papers listed above opt for their own data, which makes a comparison between different SAD approaches much harder. Probably the most commonly utilized dataset is QUT-NOISE-TIMIT [44] corpus. However, its use is limited as well. In 2015, MUSAN [45], a new standardized corpus for the training of SAD, was presented. It has become quite popular since then. Recently, AVA-Speech [46], a densely labeled dataset of speech activity in movies, was published as well.

1.2 Speaker Change Point Detection

In the literature, speaker change point detection commonly utilizes SAD as a pre-processor, and it is thus carried out only on speech segments. Furthermore, it is usually done without any prior knowledge about the identity or even the number of speakers in the recording (i.e., it is treated as a speaker-independent task). Similar to SAD, most of the existing SCP detection approaches are designed in two consecutive phases: feature extraction and change point detection itself.

In the first phase, various types of input features have been applied over the years. In the early years, more straightforward ones were successfully employed, such as zero-crossing rate or pitch [47]. Mel-frequency cepstral coefficients [48, 49] were probably the most commonly used features, followed by Line Spectrum Pairs (LSPs) [50]. Recently, the main focus has shifted to crafting more complex features capturing more speaker-specific information. Nowadays, i-vectors [51, 52] are the go-to features for most state-of-the-art systems. Alternatively, deep neural networks have also been successfully utilized to extract complex features [53, 54]. Furthermore, d-vectors were presented in [55], yielding excellent results. The latest trend goes in the direction of deep speaker embeddings [56–58] designed for end-to-end systems. Lately, x-vectors have successfully been adapted for speaker diarization [59]. In practice, the best results are often achieved by a combination of several of the features mentioned above.

In the second phase, the SCP detection approaches can be divided into three main categories: metric-, model- and hybrid-based. The first type requires a distance metric to be defined first. After that, usually, two adjacent windows are shifted alongside the recording, and the distance between them is computed. If the distance is larger than a predefined threshold (fine-tuning is the main issue), a change point is detected. The most commonly used distance metrics include the Bayesian Information Criterion (BIC) [60–62], the Generalized Likelihood Ratio (GLR) [63], the Gaussian divergence [64], the Kullback-Leibler (KL) divergence [65], or one-class support vector machines [66]. DISTBIC segmentation was proposed in [67] as well. A model-based approach utilizes trained models from labeled audio data

to detect speaker change points. Among the most common approaches, there are the Hidden Markov Models (HMMs) [68], the Gaussian mixture models [69], and the eigenvoice-based models [70]. Deep learning approaches based on DNNs [54, 71], CNNs [72, 73], unidirectional [74], or bidirectional [75, 76] Long Short-Term Memory (LSTM) RNNs all yield excellent results. Finally, hybrid-based approaches combine the metric and model-based ones to employ the advantages of both worlds (e.g., [77]).

Most of the approaches cited so far were designed with regard to the best possible quality of detection, and all of them are, of course, applicable to offline processing. However, the earlier discussed restrictions of online application are usually not taken into account during design, and the usability of these methods for online mode is therefore limited (or not discussed in the respective papers). That means that the number of approaches explicitly designed for real-time processing (e.g., of broadcast news) is much smaller. In the early years, an online two-step SCP detector utilizing the Bayesian fusion method for fusing multiple features was proposed [78, 79]. Other works focused on BIC [80, 81], XBIC [82], Log-Likelihood Ratio (LLR) [83], GMMs [69, 84, 85], or Gaussian mixture model – Universal Background Model (GMM-UBM) [86]. In [87], the authors explored BIC, i-vectors, and within-class covariance normalization for speaker diarization. The use of i-vectors for diarization was also investigated in [88]. Features extracted from Neural Network (NN) were explored in [89]. Finally, the authors in [90] studied in detail the influence of the online environment of several SCP detection approaches on a diarization system.

Several commonly used datasets are cited in the literature for training and evaluation of speaker change point detection. One of the first regularly used datasets was Hub-4 [91]. The French datasets ESTER [92], ETAPE [93], and REPERE [94] have also been frequently utilized. SCP detection can also be evaluated on multilingual database COST278 [95, 96]. Some other notable datasets are CALLHOME and NIST SRE. However, most of the published works report their results on only one preferred dataset making the system comparison rather difficult.

1.3 Existing Systems

The majority of the existing systems cover both speech activity detection and speaker change point detection, as well as additional speech processing tasks. These systems are usually designed for either speaker diarization or speaker recognition. One of the speaker recognition tools is an open-source ALIZÉ Speaker Recognition toolkit [97, 98], which provides support for SAD and SCP detection based on HMMs [99]. Speech activity detection is also covered in the Spear [100] toolkit. LIUM Speaker Diarization [101, 102] is probably the best-known toolkit for speaker diarization. It was initially developed for French ESTER2 evaluation campaign for diarization of broadcast news, and it provides tools for feature extraction (MFCCs), speech activity detection (HMMs), gender detection, speaker segmentation (GMMs, BIC), and speaker clustering. It also comes with a pre-trained broadcast model for immediate use. DiarTK [103] is another toolkit based on GMMs focused on

multistream speaker diarization. Alternatively, pyannote is a reasonably new option (written, as the name suggests, in Python) providing scripts for speech activity detection [75], speaker change point detection [75], speaker embeddings (with pre-trained models) [56], and speaker diarization pipeline [104]. It is based on LSTM RNNs, and it yields promising results. Another newer Python toolkit for speaker diarization is S4D [105]. It supports speech activity detection (based on GMM-HMM via SIDEKIT [106]) and speaker segmentation based on Gaussian divergence and BIC. Recently, speaker diarization based on DNN embeddings (specifically, x-vectors) [107] has been added to the Kaldi toolkit [108]. Pre-trained models are available as well. Finally, a new toolkit explicitly designed for VAD was released and applied in [34]. It provides implementations of several deep learning architectures for modeling, namely adaptive context attention model, DNNs, boosted DNNs, and LSTM RNNs. Other notable systems, such as CMU Segmentation toolkit, AudioSeq or SHoUT toolkit, can be utilized as well, but their newer counterparts usually outperform them.

2 Motivation and Goals

A detailed examination of the current state of the art in speech activity detection, as well as speaker change point detection, reveals two prominent features: a) deep learning is pushing the field further; and b) there is a significant lack of online SAD and SCP detectors. With this information, it is feasible to set up the motivation and consequently, the main goals of this thesis.

2.1 Motivation

Over the past few years, significant breakthroughs [109] have been achieved in deep learning. These breakthroughs have resulted in many novel approaches in various research fields, such as speech recognition [110–112], visual object recognition [113, 114], natural language processing [115, 116], and more, all yielding excellent results as compared with the previously used conventional techniques. These successes have understandably led to further application of deep neural networks to a much more varied range of research tasks. In this case, deep learning is applied to speech activity detection and speaker change point detection. Lately, several papers dealing with this topic have been published for both tasks, yet there is a lot of room for further experimentation, tuning up, and improvements. Performance in the online mode, especially, can be further enhanced.

Speech activity detection and speaker change point detection represent a very active research topic due to their varied use in a wide range of speech processing applications. Over the years, most of the published works have strictly focused on the offline use as it allows more freedom during the design of the detector. It is also easier to tune the performance of an offline system to achieve excellent results (i.e., multiple passes through data, processing of whole recording, a fusion of methods, etc.) than its online counterpart. Moreover, for many applications, it is a perfectly viable and even preferred solution. However, some applications (e.g., Television and Radio [TVR] monitoring systems) need to operate in real time and with low latency. These additional restrictions usually result in somewhat limited performance. Extension of the existing offline methods to their online use is a commonly cumbersome and complicated process, which is even quite often impossible. Moreover, the performance is usually affected as well. When designing an approach that may be used in a real-time application, it is generally more convenient to circumvent these restrictions from the initial stages of development. Online speech activity detection and speaker change point detection approaches (based on deep learning) that would reach results at least comparable with their offline counterparts would be very beneficial for many real-time speech processing applications (e.g., TVR monitoring system) in both commercial and research spheres (i.e., they could push the field further).

2.2 Practical Use in TVR Monitoring System

The author's lab has been focusing on speech processing and speech recognition for a long time. The TVR monitoring system developed at SpeechLab@TUL in cooperation with the NanoTrix company carries out 24/7 online transcription of radio and TV broadcasts in various languages. In the peak hours (during the day), it transcribes up to 120 streams in parallel in real time. During the non-prime hours (mostly at night), it still processes at least 20 online streams every second. The daily average ranges from 60 to 80 simultaneously transcribed online streams. Approximately "133" days (3,196 hours or 750 GB) of recordings are being processed every day. Daily, the biggest chunk of the transcribed data consists of Polish (80 broadcasts monitored), Czech (47) and Slovak (12) broadcasts. However, a wider range of Slavic languages, such as Russian (approximately 20 broadcasts monitored), Bulgarian (20), Croatian (10) or Serbian (10), etc., are being transcribed as well.

Integration of speech activity detection and speaker change point detection approaches into this existing system would be beneficial for many reasons. First, SAD would be used as a preprocessor for online streams to filter out non-speech events and run the transcriber only on speech ones. This should result in a significant reduction of processing time, and it should ease the CPU load as well (if the stream contains a lot of non-speech segments, e.g., music stream radios). It should also yield a better accuracy of transcriptions as the non-speech parts are omitted from being transcribed (i.e., less gibberish). Furthermore, the obtained speech segments would be used as inputs into the SCP detection and potentially other speech processing applications.

Second, the SCP detector would find and label transitions from one speaker to another. These newly defined labels would ease the handling of online streams as they would provide additional information about the content. They would also segment the streams into smaller speaker-homogeneous chunks, which could easily be further utilized. These chunks form a starting point for a full diarization system, which could be extended to speaker verification and identification systems to provide the transcribed streams with even more valuable information. The final detected segments could also be extracted and used as training data for future speaker-adaptive approaches to speech recognition. A diagram showing the joint use of SAD and SCP detection in the final system is shown in Fig. 2.1.

Unfortunately, none of the existing tools providing the SAD and SCP detection functionality is a good fit for the requirements of the TVR monitoring system for several reasons. Firstly, the tools are usually fine-tuned for specific conditions (telephone conversations, broadcasts [e.g., LIUM Speaker Diarization toolkit is tuned on French broadcasts], etc.) which may be unsuitable for TVR monitoring system. In the end, it would be necessary to train new acoustic models on proper data, which could be quite problematic and time-consuming. Secondly, the majority of the standardized tools are based on older technologies (mostly GMM-based), and nowadays, they do not yield state-of-the-art results. There are a few recent approaches based on deep learning, but extensive testing of the performance would be needed. Most importantly, none of the tools is primarily designed for online use, which is crucial

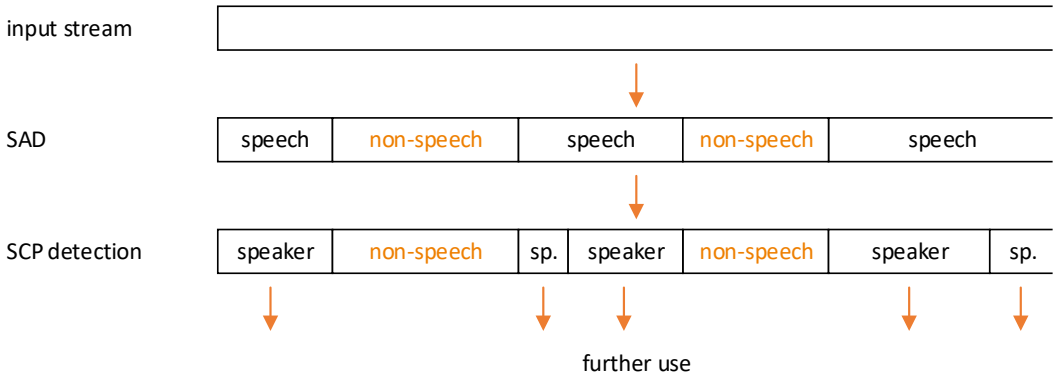

Figure 2.1: A diagram presenting the joint application of SAD and SCP detection.

for TVR monitoring systems. Usually, a whole recording is required, and some of the systems even perform multiple passes through the data (e.g., LIUM Speaker Diarization toolkit). It might not even be possible to adjust the tools to operate in online mode. Lastly, the TVR monitoring system is a distributed computational system, in which every task is represented by a corresponding docker image. This brings further requirements, such as good scalability or fast and stable implementation.

To sum up, integration of any of the existing systems would be a difficult task, and it might not even fulfill all the requirements in the end. For this reason, a preferable solution is a new, fully crafted design of online SAD and SCP detection perfectly fitting the respective TVR monitoring system.

2.3 Goals

The main goals of this thesis are thus to:

I. develop speech activity detection approach and speaker change point detection approach that:
   1. utilize state-of-the-art techniques, specifically including DNNs;
   2. allow for robust speech/non-speech and speaker change point detection;
   3. operate in an online mode with low latency in order to process real-time streams;
   4. can be integrated into the existing TVR monitoring system developed at the author's lab in cooperation with the NanoTrix company;

II. verify the proposed approaches and compare their results on publicly available datasets with selected existing approaches/toolkits.

3 Selected Approaches to Speech (Voice) Activity Detection

Due to the sheer amount of published work focused on the task of speech activity detection (see Sect. 1.1 for an overview of state of the art), this chapter only presents a detailed description of selected SAD (VAD) approaches relevant to this thesis. These approaches were chosen for two main reasons. Either they are utilized for comparison purposes (i.e., with the proposed SAD approach), or they are focused on an online application.

3.1 ITU-T G.729 Annex B

The G.729 [117] is a toll-quality speech coding algorithm adopted by the International Telecommunication Union (ITU). It was designed for multimedia and personal communication services. Later, Annex A for G.279 (G.279A) [118], providing a reduced complexity version of the speech coding algorithm, was developed for digital simultaneous voice over data. Further coding improvements could be achieved by dropping the bit rate when speech is not present. To do this, a voice activity detector identifying speech and silence/noise events needed to be crafted at first. The Annex B for G.279 (G.279B) [119] thus defined a robust frame-based voice activity detector, and subsequently, a low-bit-rate silence compression scheme. Note that this section only covers the VAD algorithm, for more information about the silence coding and reconstruction, refer to the respective paper [119].

A flowchart of the VAD algorithm is shown in Fig. 3.1.

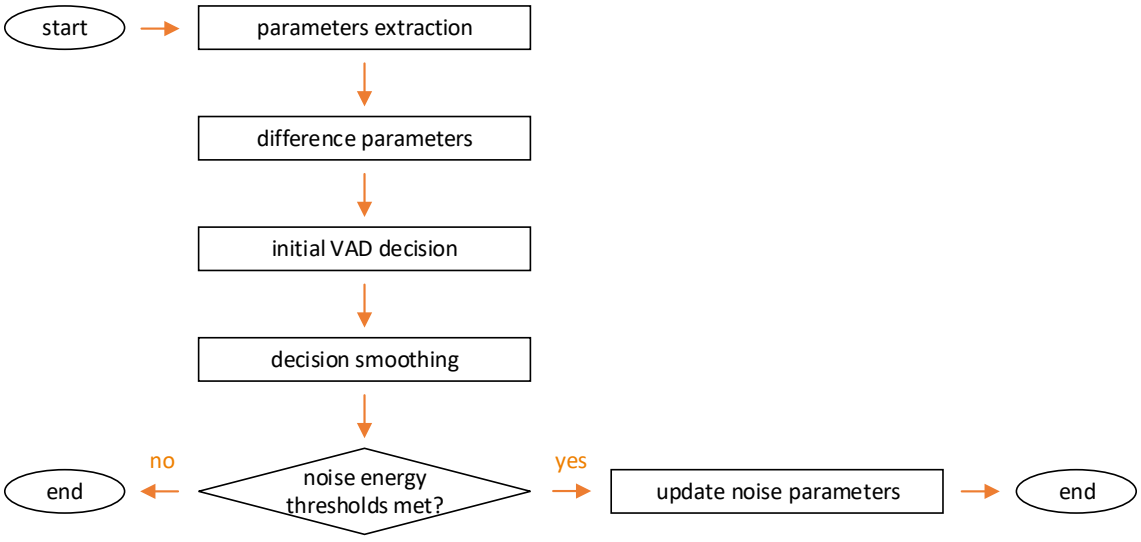

Figure 3.1: A flowchart of the selected voice activity detection algorithm presented in ITU-T G.729 Annex B.

As shown in the flowchart, the VAD algorithm operates in several consecutive phases. In the first one, a set of 4 parameters is extracted. These instantaneous parameters are computed for each frame. They focus on the energy and spectral content of the acoustic signal, and they were chosen based on classification power, robustness, and computational demands. Specifically, the parameters are:

- linear prediction spectrum;
- full-band energy;
- low-band (0 to 1 kHz) energy;
- zero-crossing rate.

Due to the varied nature of the background noise in recordings (e.g., transition from home to a busy street), an estimation of its characteristics is needed. A set of parameters similar to the instantaneous ones is kept for this reason, and it is getting updated over time (i.e., running averages of background noise characteristics). These are called estimated noise parameters.

In the second phase, the final inputs to voice activity detection are computed. These are not the instantaneous parameters, but the differences between them and estimated noise parameters. These final inputs are called spectral distortion, full-band energy difference, low-band energy difference, and zero-crossing difference, and their exact formalism can be found in the respective paper [119].

In the third phase, an initial VAD segmentation is obtained using pattern recognition. For each frame, the four computed difference parameters are projected into a four-dimensional Euclidean space, where parameters for active speech occupy a certain hypervolume while the parameters for non-speech are clustered in another. A piecewise linear three-dimensional decision boundary then separates these hypervolumes and marks the initial speech and non-speech regions. Note that the decision boundary was determined by the authors by visually inspecting a vast number of projected parameters of an extensive dataset.

So far, the VAD segmentation is done for each frame independently. However, the standard duration of speech (or non-speech) events is at least several frames. For this reason, in the fourth phase, a smoothing algorithm utilizing several past frames is applied. In [119], the authors defined four different smoothing rules:

- speech segment is extended to the current frame if the energy of the current frame is above a certain threshold;
- speech segment is extended to the current frame if two previous frames were speech ones, and the absolute energy difference between current and previous frames is under a certain threshold (only applied twice);
- non-speech segment is extended to the current frame if ten previous frames were non-speech ones, and the absolute energy difference between current and previous frames is under a certain threshold;

- current speech frame is corrected to a non-speech one if the energy of the frame is below the noise energy thresholds by a certain margin, and neither of the first two smoothing rules was applied.

After enforcing all these smoothing rules, the final VAD segmentation is obtained.

In the final phase, the running averages of background noise characteristics are updated if the background noise energy thresholds are met.

3.2 ETSI Advanced Front-End

Degradation in the performance of speech recognizer on speech transmitted over mobile channels is to be expected due to, e.g., low bit rate or transmission errors. To prevent this degradation, Distributed Speech Recognition (DSR) systems replace the speech channel with an error protected data channel transmitting parametrized speech to the recognizer. The DSR systems are thus formed by two main parts, a front-end one performing the signal parametrization and transmission and a back-end one doing the transcription. The ETSI Standard (ES) [120, 121] covered the former part. Besides, the standard also described a recommended front-end VAD algorithm, which filters out non-speech segments.

The front-end VAD algorithm can be divided into two stages: frame-based detection and a decision stage. In the former stage, Mel-warped Wiener filter coefficients are applied as inputs, and as authors in [120] stated, the detector exploits the energy associated with voice onset, where the energy for each frame is obtained from:

- energy values across the whole spectrum;
- energy values over a part of the spectrum (containing fundamental pitch);
- acceleration of the variance of energy values (in the lower half of the spectrum).

After computing all three metrics, these are compared with predefined thresholds (see the standard for exact values), and a binary value for each measurement is given: 1 for suspected speech or 0 for suspected non-speech. Note that these metrics were designed to complement each other and to provide the voice activity detector with inputs robust to noise for the latter (decision) stage.

The two main components of the decision stage are a buffer of fixed size (i.e., seven frames in the given standard – 1 current frame, and 6 following frames) and a value called hangover timer. For each frame, the decoder makes a binary decision based on the three input metrics: 1 (suspected speech) if at least one of the metrics is 1, 0 (suspected non-speech) otherwise. The resulting value gets stored in the first component – buffer. Once the buffer is filled up, the decision algorithm can start. For the next frame, the oldest value in the buffer is shifted out, and the new one is inserted. This results in a frame delay of the size of the buffer minus one (i.e., six frames in this case). The second component, the hangover timer, decides the final output. During the decoding, for each frame, the algorithm searches the buffer for the longest sequence of ones (suspected speech). If the sequence is greater than

a given threshold, the hangover timer is set to a predefined positive value (see the standard for the exact numbers). If not, it is lowered by 1. Finally, if the value of the hangover timer is greater than 0, the frame is considered speech, otherwise non-speech. This allows the decoder to smooth the transitions between speech and non-speech events effectively. An overview of the whole recommended front-end VAD algorithm in ETSI Standard is illustrated in Fig. 3.2. More information about the standard and all its components can be found in [120].

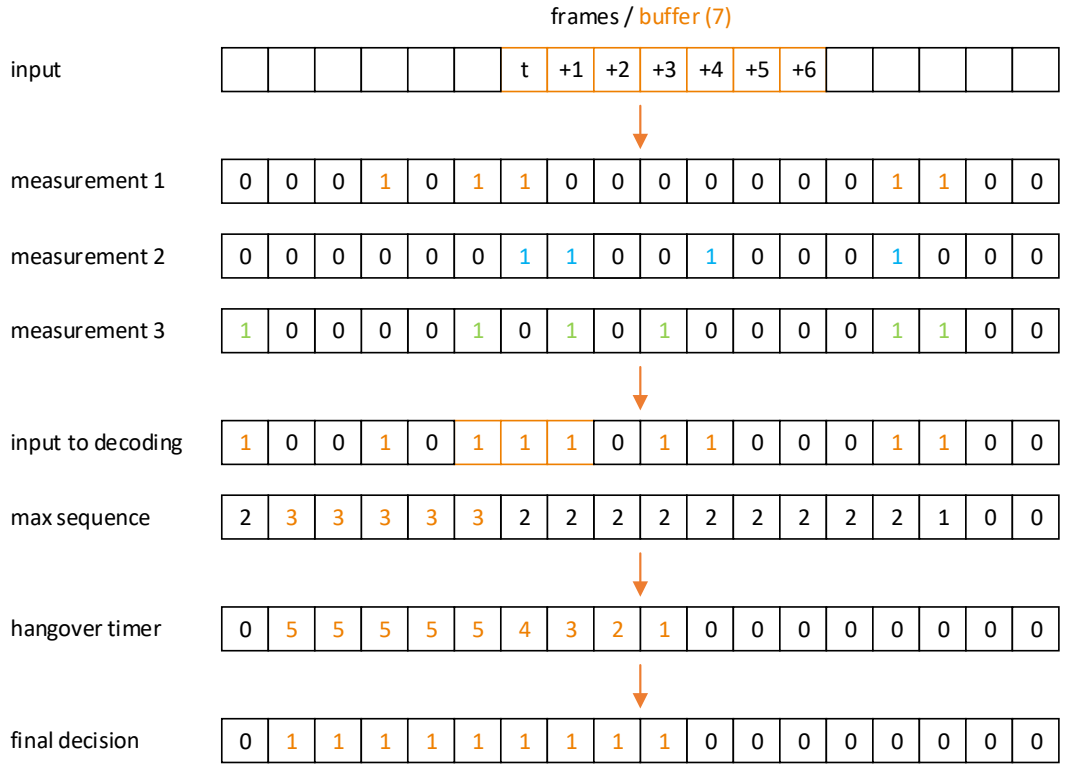

Figure 3.2: An overview of the recommended front-end VAD algorithm in ETSI Standard (ES). In this example, the buffer is set to 7 frames, the sequence of ones to at least 3 frames, and once reached, the hangover timer is set to 5. The final output is in the last row, where ones mark speech frames and zeros non-speech ones.

3.3 Model-Based Likelihood Ratio

Similar to G.279 Annex B (see Sect. 3.1), the authors of [122, 123] focused on crafting a robust voice activity detection algorithm designed for speech coding applications. They explored the possibility of improving the decision rule, which is used to determine the presence or absence of speech in a recording by comparing the statistics of the current frame with estimated noise statistics. To make improvements, the authors first proposed the use of a statistical model in [122], where the decision rule comes from the likelihood ratio test by estimating unknown parameters using the maximum likelihood criterion. Later on (specifically, in [123]), a decision-directed

method for the estimation of the unknown parameters was suggested as a further improvement to the decision rule.

In the second paper [123], the authors also presented a hangover (smoothing) scheme, which is used to smooth the initial voice activity decisions. This scheme is based on hidden Markov models; it models a sequence of frames as a first-order Markov process. After enforcing this smoothing scheme, the final VAD decisions are obtained for each frame.

Finally, according to the results presented in [123], the final VAD approach yielded better performance than the VAD of G.279B, especially under low Signal-to-Noise Ratio (SNR) conditions. It also required fewer optimized parameters.

3.4 Long-Term Spectral Divergence

The main focus of [124], similarly to the previously showcased works in this chapter, was on crafting a noise-robust voice activity detector. However, unlike the previous works, the primary goal here was not an improvement in speech coding but a better performance of a follow-up speech recognition system (i.e., a similar goal to this thesis). Specifically, the authors studied the benefits of utilizing long-term information of speech signals on voice activity detection and, consequently, speech recognition.

The VAD algorithm they proposed utilizes a long-term speech window instead of instantaneous parameters (as opposed to, e.g., VAD of G.279B). It is based on the estimation of the Long-Term Spectral Envelope (LTSE). The speech/non-speech decision rule is then determined by a Long-Term Spectral Divergence (LTSD) between the speech and noise (the LTSE is compared to the average noise spectrum).

Formally, the N-order LTSE of a (noisy) signal $x[n]$, which is segmented into overlapping frames, can be defined as:

$$LTSE_N(k,l) = max(X(k,l+j))_{j=-N}^{j=+N} \ , \tag{3.1}$$

where $X(k,l)$ is an amplitude spectrum for the k band at frame l.

From LTSE, it is possible to express the N-order LTSD as:

$$LTSD_N(l) = 10\log_{10}\left(\frac{1}{NFFT}\sum_{k=0}^{NFFT-1}\frac{LTSE_N^2(k,l)}{N^2(k)}\right) \ , \tag{3.2}$$

where $N(k)$ is the average noise spectrum magnitude for the k band, and $NFFT$ is the total number of bands. The respective paper also suggested that the ideal value of the LTSD order is 6 (the best compromise between high discrimination decision rule and minimalization of the average number of decision errors). More detailed information about finding the ideal order and discrimination power of LTSD can be found in the given paper [124].

The VAD algorithm runs in several phases, which are shown in the flowchart in Fig. 3.3. It begins with an initialization phase during which the mean noise spectrum is estimated by averaging the noise spectrum magnitude. After that, the input signal is segmented into overlapping frames. For each frame, the spectrum

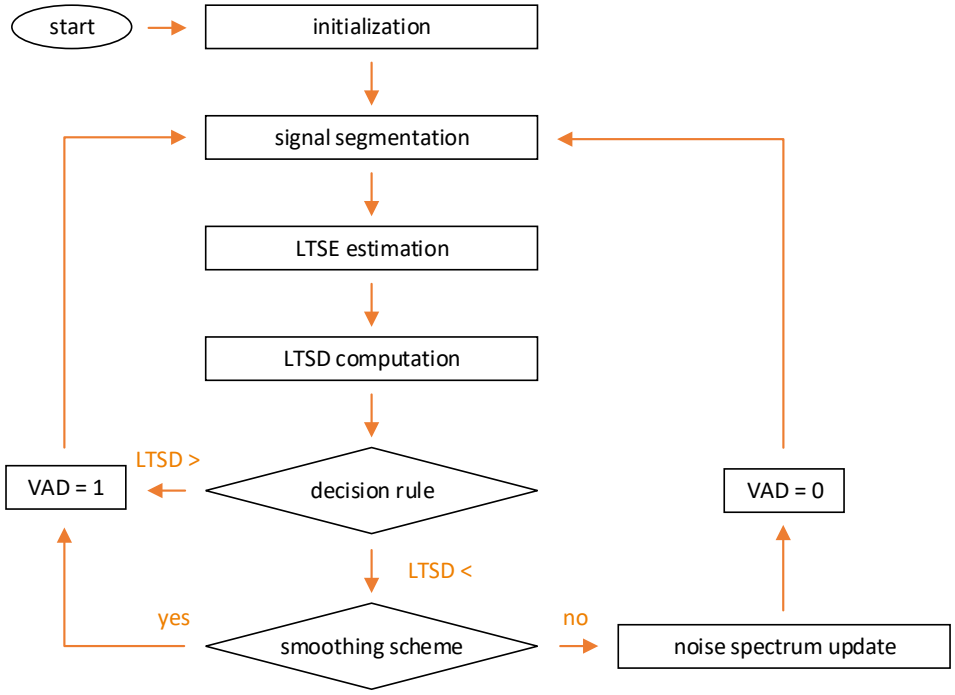

Figure 3.3: A flowchart of the selected VAD algorithm: long-term spectral divergence. VAD = 1 marks speech, VAD = 0 means non-speech.

$X(k, l)$ is obtained by applying a context window of size $2N + 1$ frames. In the two following phases, the LTSE is estimated based on Eq. (3.1) and, consequently, the decision rule is derived from LTSD, which is computed using Eg. (3.2). The decision threshold is also adapted according to the measured noise energy. A hangover scheme is present, as well. However, it is only applied at low SNR levels, where it is used to delay transitions from speech to non-speech. If the noise level is low, the hangover scheme is not active. Last, during non-speech segments, the noise spectrum $N(k)$ is updated. Note that the delay of the VAD algorithm is N frames because the decision for each frame is based on the context window of size $2N+1$ frames around the actual frame (N preceding frames, the current frame, and N following frames).

The authors also compared their proposed VAD algorithm to the most commonly utilized VAD approaches. They focused on two tasks – speech/non-speech discrimination and utilization of VAD in a speech recognition system. Their results show superior performance for both tasks over the other standard VAD approaches, e.g., G.729B. The detailed results are described in the paper [124].

3.5 GMM-Based Approach

Over the years, many approaches to speech (voice) activity detection have utilized Mel-frequency cepstral coefficients (e.g., [8, 9]) as features, and Gaussian mixture models (e.g., [8, 20, 21]) for classification. However, this section showcases the approach detailed in [44] because it is used for comparison purposes later in this thesis.

Additionally, the respective paper presented not only the GMM-based approach but also a standardized QUT-NOISE-TIMIT corpus.

The cepstral features, MFCCs [125], are the first component of the showcased paper. These features are widely utilized for many speech processing applications, including speech recognition or speech activity detection. They are known for their discrimination power. MFCCs are usually computed in several consecutive steps:

1. the windowed input signal is segmented into (overlapping) frames;
2. the magnitude spectrum is obtained by taking fast Fourier transformation;
3. the magnitudes are mapped on Mel-scale using triangular filters, for each filter, the powers (or magnitudes) are summed;
4. the logarithm of the summed powers (or magnitudes) is taken;
5. the final MFCCs are obtained by taking discrete cosine transformation of the logarithm of the summed powers (or magnitudes);
6. optionally, cepstral mean subtraction can be used for normalization, and/or differential (Δ) and acceleration ($\Delta\Delta$) coefficients can be computed.

In [44], the authors did not reveal their specific setting they used for extraction of MFCCs.

As a second component, probabilistic Gaussian mixture models are utilized as a binary classifier. Naturally, the two classifiable classes are speech and non-speech. For this reason, two M-mixture GMMs are used to model the distributions of the two respective classes [126]:

$$P(x_i|H_1) = \sum_{m=1}^{M} c_{1m} \frac{1}{\sqrt{(2\pi)^D |\Sigma_{1m}|}} exp\left(-\frac{1}{2}(x_i - \mu_{1m})' \Sigma_{1m}^{-1}(x_i - \mu_{1m})\right) \quad , \tag{3.3}$$

$$P(x_i|H_0) = \sum_{m=1}^{M} c_{0m} \frac{1}{\sqrt{(2\pi)^D |\Sigma_{0m}|}} exp\left(-\frac{1}{2}(x_i - \mu_{0m})' \Sigma_{0m}^{-1}(x_i - \mu_{0m})\right) \quad , \tag{3.4}$$

where x_i is the input feature vector at the frame i, H_1 and H_0 denote the speech and non-speech hypothesis, D marks the dimensionality of the input, and finally, c_m, μ_m and Σ_m mark the parameters of the mth mixture (note that the exact number of mixtures was not discussed in the showcased paper). These parameters are the mixture weight, mean and covariance, respectively.

In the following step, the Log-Likelihoods (LLs) of the speech and non-speech models at frame i can be computed. From them, it is possible to express the log-likelihood ratio as their difference:

$$LLR_i = \log\left(P\left(x_i|H_1\right)\right) - \log\left(P\left(x_i|H_0\right)\right) \quad . \tag{3.5}$$

Next, in [44], the system is smoothed by a 1-second median filter designed to suppress short-term variation. Finally, a threshold tuned on training data is applied, and the final speech/non-speech segmentation is obtained (i.e., speech output for frames with the LLR greater than the threshold and non-speech for the rest).

The parameters of the models (mixture weights, means, and covariances [c_m, μ_m, Σ_m]) were estimated during the training phase. Unfortunately, the exact training procedure applied in the showcased paper was not given. However, the GMMs for VAD are usually initialized by an iteration of the k-means clustering [127] and then refined with maximum likelihood estimation by running several iterations of the expectation-maximization algorithm [128] (as shown, e.g., in [8, 20]).

The experimental evaluation using the QUT-NOISE-TIMIT corpus shows excellent performance under all noise conditions. The detailed results can be seen in the respective paper [44].

3.6 Subband Noncircularity

Throughout the years, many speech (voice) activity detection approaches have used spectral features. However, most of these approaches only utilize magnitude (or energy) of the complex-valued spectral representation and completely ignore the rest of the information. In [129], the authors explored this additional information as well as its effects on the performance of VAD in a noisy environment. Specifically, they inspected the second-order statistical behavior of complex data. They exploited a property of complex subbands of speech and noise – the second-order noncircularity, which is also known as impropriety. Higher impropriety usually suggests a presence of speech. In their work, two VAD methods were proposed:

- an unsupervised method designed for single-channel data (this method does not rely on non-speech segments to estimate the noise parameters);
- a supervised method designated for two-channel data.

Instinctively, both of these methods employ features based on the impropriety. An exhaustive explanation of the proposed methods (and impropriety-based features) is available in the given paper [129].

The authors reported their achieved results on the QUT-NOISE-TIMIT corpus. The results (for both methods) show excellent performance and at least comparable numbers to approaches introduced in Sect. 3.3 and 3.4 under all noise conditions. The only exception was a scenario with high amounts of reverberation. The detailed results for both single- and two-channel methods are in the respective paper [129].

3.7 Complete-Linkage Clustering

In [22], the authors focused on robust voice activity detection under high noise conditions. They used their previous work based on GMMs and MFCCs [44] (described in Sect. 3.5) as a baseline and extended it with Complete-Linkage Clustering (CLC),

which was incorporated for the purpose of making the final speech/non-speech decision. In addition, the authors also implemented their approach to the task of audio-visual voice activity detection (for specifics, refer to the respective paper [22]).

Because their new work is built upon the older one, both share many similarities. First, the MFCCs are employed as input features. Specifically, 19-dimensional MFCCs (including zero coefficient, delta coefficients, and feature warping) are utilized. Next, two 16-mixture GMMs are used to model the distributions of speech and non-speech. These were trained, as previously, on subsets of the QUT-NOISE-TIMIT corpus (following the recommended training/testing protocol as specified in [44]).

One of the differences lies in the computation of the log-likelihoods of the speech and non-speech models. These are newly not computed per frame, but instead for a sequence of frames (i.e., short segments):

$$LL_X^s = \log P\left(X|H_1\right) = \sum_{i=1}^{I} \log P\left(x_i|H_1\right) \quad , \tag{3.6}$$

$$LL_X^{ns} = \log P\left(X|H_0\right) = \sum_{i=1}^{I} \log P\left(x_i|H_0\right) \quad , \tag{3.7}$$

where segment X is a sequence of I frames (i marks the ith frame of this sequence), H_1 and H_0 denote the speech and non-speech hypothesis, and the definitions of $P\left(x_i|H_1\right)$ and $P\left(x_i|H_0\right)$ were already given in Eq. (3.3) and (3.4), respectively. In practice, the authors set the length of these segments to 5 frames (50 ms). For each segment, the log-likelihoods of the speech and non-speech models can be utilized to express the log-likelihood ratio as:

$$LLR_X = LL_X^s - LL_X^{ns} \quad . \tag{3.8}$$

The completely new component of the proposed approach is the complete-linkage clustering. This agglomerative clustering technique is used to merge the segments based on the computed log-likelihood ratios. To do this, a pairwise dissimilarity score between all pairs of segments needs to be defined first. The authors expressed this score (between two segments X_i and X_j) as:

$$d(i,j) = \frac{1}{|LLR_{X_i} - LLR_{X_j}|} \quad , \tag{3.9}$$

where LLR_{X_i} and LLR_{X_j} are the log-likelihood ratios of segments i and j, respectively. From the definition, this score is lower for similar segments (either speech or non-speech) and higher for more diverse ones (one speech and one non-speech).

The CLC algorithm runs in iterations as follows. First, the dissimilarity scores for each segment pair are computed, and the two most similar segments are merged into a new cluster. Next, the scores between this new cluster and the remaining segments need to be updated. For each remaining segment, the new score is set to the highest dissimilarity score between all of the segments of the new cluster and

the respective remaining segment. After that, this whole process is repeated until only two final clusters remain.

The last step is to decide which of the two clusters represent speech. This can be done by computing the log-likelihood ratios of both clusters. If the LLR of the first cluster is greater than the LLR of the second cluster, it is considered as speech, otherwise non-speech.

Finally, the authors also implemented a hangover scheme based on their previous work [7]. For each speech segment, a portion of the recording before (300 ms) and after (500 ms) the corresponding segment is also labeled as speech. If the speech segment is shorter than 250 ms, and there is no other speech segment in close proximity, it is relabeled as non-speech. After enforcing this smoothing scheme, the final speech/non-speech segmentation is obtained.

The main advantage of this approach is that there is no threshold tuning, and it can be easily applied to multiple audio domains. On the other hand, the algorithm fails if there is no speech (non-speech) event in the recording. However, this can be solved by appending a short dummy speech (non-speech) segment to each recording.

The results reported by the authors yielded overall significant improvements on the QUT-NOISE-TIMIT corpus over the previous GMM-based approach under all noise conditions. These improvements were the most noticeable under high noise conditions. The results are presented in detail in the corresponding paper [22].

3.8 DNN-Based Approach

The authors of [8] focused on applying speech activity detection to videos uploaded to a video-sharing website – YouTube. Usually, the noise conditions under which these YouTube videos are recorded vary greatly. This is not ideal for the standard GMM-based approach to SAD, whose performance starts to degrade quickly if the noise conditions are not static. For this reason, the authors explored the possibility of replacing GMMs with deep neural networks.

In their work, they utilized a portion of HAVIC corpus [130], which was manually annotated for speech, music, noise, and singing at the Linguistic Data Consortium. These annotations were used to define four environments for the speech/non-speech decisions: music present, noise present, singing present, and clean. Two approaches to SAD were trained and evaluated on this data:

- GMM-based approach:

  The baseline GMM-based approach is similar to the approach described in detail in Sect. 3.5. For inputs, it extracts 13-dimensional MFCCs, which are normalized per file to have zero mean and identity variance. The final input feature vector is formed by concatenating the normalized features and their Δ and $\Delta\Delta$ coefficients (i.e., it is 39-dimensional).

  For classification, two 128-mixture GMMs are employed to model the distributions of speech and non-speech. The training of these two GMMs was done

as described earlier in Sect. 3.5. The GMMs were first initialized by an iteration of the k-means clustering and then finalized by 20 iterations of the expectation-maximization algorithm.

To make the speech/non-speech decision, the authors apply two different segmentation schemes. The first one corresponds to the one described in Sect. 3.5. For each frame, the decision is made by comparing the log-likelihood ratio (Eq. (3.5)) to a predefined threshold (tuned on the test set to achieve the best equal error rate). The latter scheme generates decisions per frame by Viterbi decoding [131] of the log-likelihoods of GMMs using a 2-state HMM (speech/non-speech) [8]. The parameters of HMM (state-transition probabilities and state priors) were set to the observed values in training data.

- DNN-based approach:

  The second approach is based on feed-forward deep neural networks. It uses the same 13-dimensional MFCCs, which are also normalized per file to have zero mean and identity variance. In this case, instead of using the Δ and $\Delta\Delta$ coefficients, a 0.8-second context window is exploited. The final input feature vector is thus formed as a concatenation of 40 previous frames, the current frame, and 40 following frames.

  For classification, a binary (speech/non-speech) fully connected feed-forward DNN was trained. For the training phase, its hyper-parameters were set to:

  - 3 hidden layers;
  - 512 neurons per hidden layer;
  - ReLU activation function;
  - 2 output neurons (softmax units)
  - mini-batches size of 50;
  - 0.001 learning rate;
  - momentum of 0.9;
  - 50 epochs.

  Similar to the GMM-based baseline, two different segmentation schemes are used to make the speech/non-speech decision. The first one makes the decision for each frame by comparing the speech state posterior (from DNN) to a predefined threshold (tuned on the test set to achieve the best equal error rate). The latter scheme, as before, generates decisions per frame by Viterbi decoding using a 2-state HMM (speech/non-speech). The inputs to the Viterbi decoding are the log-likelihoods computed by dividing the state posteriors (from DNN) by the state priors (from training data).

From the results the authors published in [8], two major conclusions can be made. First, the segmentation scheme using Viterbi decoding yielded better SAD performance than the thresholding scheme, and second, the DNN-based approach significantly outperformed the GMM-based baseline on data drawn from YouTube. The specific results can be seen in the corresponding paper [8].

3.9 Conditional Random Fields

In [36], the authors focused on developing a new speech activity detection approach designed specifically to be incorporated in an online speech recognition/speaker diarization system. As a baseline, they used their BBN Broadcast Monitoring System, which provides automatic rich transcriptions in real time. The SAD portion of this system was originally done by a phone-class decoder [132]. However, this approach was prone to over-segmenting resulting in, e.g., fragmented words. For this reason, the authors proposed a new SAD approach, which can scale the speech boundary in real time. They also evaluated the influence of this new approach on both speech transcription and speaker diarization tasks in an online environment.

The proposed SAD approach is operated in two phases – silence detection and speech/non-speech labeling. In the first phase, an energy-based silence detection is applied to eliminate the silence frames from the speech/non-speech labeling and to provide initial segmentation change points. The approach uses a dynamic energy threshold which is calculated for 5-second chunks as:

$$E^i = E^i_{min} + K\left(E^i_{max} - E^i_{min}\right) \quad , \tag{3.10}$$

where i is the index of ith 5-second chunk, E^i_{max} and E^i_{min} are the average energies of the top 5% of highest and lowest energies, respectively, and K is a tunable parameter. To estimate this threshold, an initial SAD segmentation is done using GMMs by comparing the speech and non-speech log-likelihoods (see Sect. 3.5). For each chunk, if the initial segmentation says there is less than 1 second of speech, a fixed energy threshold is applied, and the frames with energy lower than the threshold are labeled as silence frames. The adjacent silence frames are joined into segments, and only segments longer than a set duration (D_{min} – second tunable parameter) are considered as the final silence.

For the latter phase (speech/non-speech labeling), which is performed only on non-silence segments, conditional random fields [133] first used for VAD in [134] are utilized. In their work, the authors model sequence-level labels instead of frame-level ones (all frames in the respective segment have the same label). The CRFs give the posterior probability of label sequence y ($y = [y_1 \dots y_t]$) given an input sequence z ($z = [z_1 \dots z_t]$). CRFs thus maximize the conditional probability $P(y|z)$ [36]:

$$P(y|z) = \frac{1}{Z(z)} exp\left(\sum_t \left(\sum_{i,j \in S} w_{ij} f_{ij}\left(y_{t-1}, y_t\right) + \sum_{i \in S, d \in D} v_{id} g_{id}(y_t, z_t)\right)\right) \quad , \tag{3.11}$$

where $Z(z)$ is a constant used for normalization, S marks the possible labels (speech, non-speech), w_{ij} is the weight for state-transition between two neighboring segments, f_{ij} is defined as:

$$f_{ij} = \begin{cases} 1, & y_{t-1} = i \text{ and } y_t = j \\ 0, & \text{otherwise} \end{cases} \quad , \tag{3.12}$$

D is the feature set, v_{id} is the weight of node features (for node features, the log-likelihoods of speech/non-speech GMMs are used), and g_{id} is expressed as:

$$g_{id} = \begin{cases} z_t[d], & y_t = i \\ 0, & \text{otherwise} \end{cases}, \tag{3.13}$$

where $z_t[d]$ is the dth feature of z_t [36]. Finally, the parameters of CRFs are estimated by L-BFGS [135], and Viterbi decoding is utilized to find the most likely sequence of labels y.

In their experimental evaluation, the authors showed that two different SAD configurations (one for speech recognition and one for speaker diarization) obtained by choosing different parameters (K and D_{min}) were beneficial to the performance of both follow-up tasks. In both tasks, the authors were able to outperform their baseline system. As usual, more details are available in the respective paper [36].

3.10 Simultaneously Trained Online Decoder

The main focus of [42] was on improving the performance of online speech activity detection. In this approach, the authors utilized standard signal processing techniques as well as deep learning to simultaneously train the SAD decoder. They also explored different input features, such as spectral flux and spectral variance.

The proposed SAD approach consists of three main components, which are all integrated into the training process. The first component provides early speech activity score given a short context of input features (spectrograms). It is based on a convolutional neural network. The CNN is composed of several convolutional (3), max-pooling (3), and fully connected (5) layers. The stated details are 3×3 window, 6 kernels, and ReLU units for convolutional layers and 256 neurons, and ReLU units for standard fully connected ones. The last fully connected layer is for computing the score, and it has a single neuron.

The goal of the second component is to smooth the early activation score by a 2-second window (within the paper, the authors also experimented with different values). For this task, a single neuron with a linear activation function is employed.

As a part of the third component, a fixed output layer with fixed bias (0) and weights (−1 and 1) is added (i.e., positive posterior indicates speech, negative non-speech). The final component itself is a differentiable decoding process. The max-pooling decoder uses a window of scores (0.5 seconds) and searches for a maximum score. If the maximum score is positive, the output is speech, otherwise non-speech. This results in speech segments with duration, which is at least equal to the length of the window.

The authors presented their promising results on a Czech radio broadcasting corpus. The detailed results, which also provide a comparison of different input features, can be seen in the corresponding paper [42]. In the future, the authors intend to continue improving their approach by introducing additional components.

4 Selected Approaches to Speaker Change Point Detection

Speaker change point detection is probably an even more active research topic than SAD. For this reason, this chapter only presents a detailed description of selected approaches to SCP detection (a brief overview of the state of the art is available in Sect. 1.2). These selected approaches were chosen for two main reasons. Either they serve as a reference system (for comparison purposes with the proposed SCP detection approach), or they are applicable in online mode.

4.1 BIC-Based Approach in LIUM Toolkit

The LIUM Speaker Diarization toolkit[1] [101, 102] is open-source software (written in Java) designed for multi-pass offline speaker diarization of (mostly) broadcast news. As such, it provides tools for feature extraction (MFCCs computation), speech activity detection, gender detection, speaker change point detection, and speaker clustering. At first, it was developed for the French ESTER2 evaluation campaign, where it won the best results for speaker diarization of broadcast news [136]. In [101], the authors also discussed an application to a different domain – telephone conversions. Nowadays, the toolkit is freely available for download, and it comes with pre-trained and fine-tuned models for TV and radio broadcasts. For different domains, the models would need to be crafted from scratch (to match the broadcast performance).

The LIUM toolkit performs the SCP detection by first running feature extraction, which is followed by the application of four major steps – BIC segmentation, BIC clustering, Viterbi decoding, and boundary adjustments (see flowchart in Fig. 4.1).

The toolkit utilizes sphinx4, a Java speech recognition library, for computation of features (MFCCs). Different configurations of MFCCs are applied for different steps of the SCP detection. For the first two steps (i.e., BIC segmentation and BIC clustering), the extracted MFCCs are 13-dimensional with the zeroth coefficient included. No normalization is applied. For the remaining steps, Δ coefficients are added, and the zeroth coefficient is dropped.

The first step, BIC segmentation, is done by performing two consecutive passes through the input recording. The initial pass provides an early metric-based estimation of change points placements, and it closely follows the approach described in [65]. The distance metric applied here is the generalized likelihood ratio, which is computed using Gaussians with full covariance matrices [101]. The GLR is computed between a pair of 2.5-second (neighboring) sliding windows (i and j) as follows [137]:

$$GLR_{i,j} = 2n \log |\Sigma| - n \log |\Sigma_i| - n \log |\Sigma_j| \ , \tag{4.1}$$

[1] https://projets-lium.univ-lemans.fr/spkdiarization/

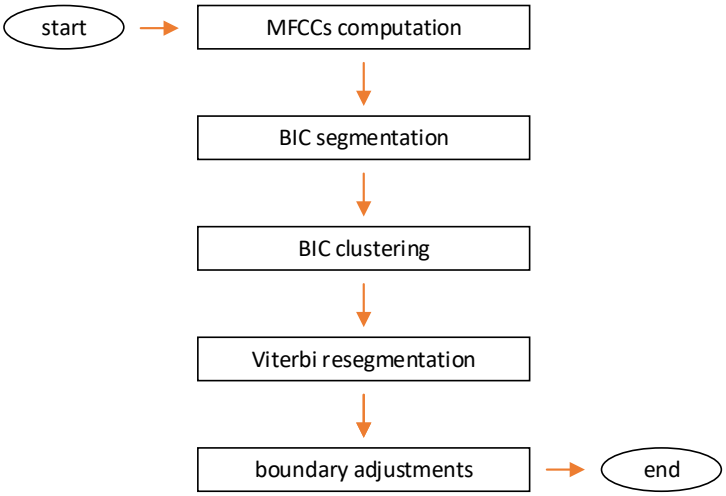

Figure 4.1: A flowchart of the multi-pass offline SCP detection in the LIUM toolkit.

where n is the length of the window ($i = j$), and $|\Sigma_i|$, $|\Sigma_j|$ and $|\Sigma|$ are the determinants of covariances of the first, second and both (merged) windows, respectively. The two neighboring windows are shifted alongside the recording, and the GLR is computed for each shift. The initial change points are then placed in the spots, where the GLR reaches a local maximum. The second pass refines these initial change points by merging consecutive segments of the same speaker based on the Bayesian information criterion. For every two neighboring segments (i and j), ΔBIC is computed using full covariance Gaussians as [101]:

$$\Delta BIC_{i,j} = \frac{n_i + n_j}{2} \log|\Sigma| - \frac{n_i}{2} \log|\Sigma_i| - \frac{n_j}{2} \log|\Sigma_j| - \lambda P \ , \tag{4.2}$$

where n_i and n_j are the lengths of the first and second segment, $|\Sigma_i|$, $|\Sigma_j|$ and $|\Sigma|$ are the determinants of covariances of the first, second and both (merged) segments, respectively, λ is a tunable parameter, and P is a penalty factor, which is set in LIUM toolkit to:

$$P = \frac{1}{2}\left(d + \frac{d\,(d+1)}{2}\right) + \log\left(n_i + n_j\right) \ , \tag{4.3}$$

where d is the dimension of the input features. The two neighboring segments are merged in one if ΔBIC is greater than 0. Alternatively, other distance metrics, such as Gaussian divergence [64] or Kullback-Leibler distance [65], are also implemented in the LIUM toolkit.

In the second step, an algorithm based on hierarchical agglomerative clustering is applied. In the beginning, each of the segments defined in the first step is considered as an initial cluster. After that, ΔBIC is computed for each cluster pair, and the pair with the highest positive ΔBIC score is merged into a new cluster. This is repeated in iterations until there are no cluster pairs with positive ΔBIC. Note that ΔBIC is computed as in Eq. (4.2) with the only exception that i and j represent clusters instead of segments.

The third step performs the resegmentation of the input recording based on the Viterbi decoding. Each of the clusters obtained from the previous step is modeled

by an HMM with one state, represented by a GMM with eight mixtures (learned over the segments of the cluster by the expectation-maximization algorithm). The log-penalty between two HMMs is set empirically. Finally, the Viterbi decoding is run to generate the new speaker change points.

Due to the often imperfect placements of these new speaker change points (e.g., within words), a set of rules defined experimentally is employed in the last step. Specifically, the placements are refined by (slightly) moving the speaker change points towards regions with low energy. This step concludes the SCP detection portion of the LIUM toolkit and results in the final speaker-homogeneous segments.

The follow-up applications (e.g., speech activity detection, gender detection, and speaker diarization), training procedures (for crafting new models), results (of diarization), and programming tips (for functionality extensions) are all well-discussed in the respective papers detailing the LIUM Speaker Diarization toolkit [101, 102].

4.2 Bayesian Fusion Method

The main focus of the authors of [78, 79] was in crafting unsupervised real-time SCP detection (and speaker tracking) approach tuned for broadcast news. For this reason, the authors worked with no prior knowledge of the number or identity of the speakers. The SCP detection portion of the proposed algorithm is covered in three main steps – feature extraction, potential change point detection, and refinement. The authors also proposed a Bayesian fusion method to fuse multiple input features. Speaker tracking is done based on the obtained speaker-homogeneous segments [79].

The SCP detection approach starts with a preprocessing and feature extraction step. The input audio stream is divided into 3-second segments (windows) with a 2.5-second overlap. A voice activity detection algorithm is run on these segments to filter out non-speech parts (the details of the VAD were not given, only references to previous works [138, 139] are available). For each segment, three different types of input features are extracted per 25 ms non-overlapping frames. Specifically, the features are MFCCs, LSPs [140] (both commonly utilized for speech processing), and pitch (used here to differentiate between male and female speakers). Cepstral mean subtraction is applied for normalization purposes.

Since the proposed SCP detection approach is metric-based, a distance needs to be defined first. The authors opted for a distance derived from the Kullback-Leibler divergence [141]. Assuming that each segment is modeled as a Gaussian, KL divergence between two segments (i and j) can be defined as [79]:

$$\begin{aligned} KL_{i,j} = &\frac{1}{2}\,\mathrm{tr}\left[(\Sigma_i - \Sigma_j)\left(\Sigma_j^{-1} - \Sigma_i^{-1}\right)\right] \\ &+ \frac{1}{2}\,\mathrm{tr}\left[\left(\Sigma_j^{-1} - \Sigma_i^{-1}\right)(\mu_i - \mu_j)(\mu_i - \mu_j)^T\right] \quad , \end{aligned} \tag{4.4}$$

where Σ_i and Σ_j are the estimated covariance matrices of the first and second segments, respectively, and μ_i and μ_j are their estimated mean vectors. Since the means can be easily biased by environmental conditions [78], the authors ignore the

second part of Eq. (4.4) and use distance called divergence shape [142]:

$$D_{i,j} = \frac{1}{2} \operatorname{tr}\left[(\Sigma_i - \Sigma_j)\left(\Sigma_j^{-1} - \Sigma_i^{-1}\right)\right] \quad . \tag{4.5}$$

Note that this distance is greater if the two segments come from different speakers.

At the start of the second step (initial SCP detection), a speaker model is estimated for each segment (as they are processed). These speaker models are later used in the refinement step (and speaker tracking). The divergence shape is computed between two neighboring segments using only the LSPs (MFCCs and pitch are utilized for refinement). A speaker change point (between these two segments) is detected if:

$$\begin{aligned} D_{i,i+1} &> D_{i+1,i+2} \\ D_{i,i+1} &> D_{i-1,i} \\ D_{i,i+1} &> \lambda \quad . \end{aligned} \tag{4.6}$$

The first two conditions guarantee that the speaker change point is placed in a local maximum, while the last one prevents false speaker change points in low peaks by applying an automatic threshold. If no speaker change point is detected, the algorithm assumes that both segments come from the same speaker, and it updates the corresponding speaker model (comprised of LSPs, MFCCs, and pitch) accordingly. Explicitly, the speakers are modeled by the GMMs with up to 32 mixtures.

The automatic threshold is set up rather to produce false speaker change points than to omit the real ones. For this reason, the goal of the last refinement step is to eliminate the false speaker change points by merging neighboring speaker models (segments) if the speaker is the same. All features are employed, and MFCCs, LSPs, and pitch distances between the previous speaker model and the model of the current segment are computed. These distances are then fused using a Bayesian decision engine, as shown in Fig. 4.2, to obtain a more reliable decision. If the computed likelihood ratio is greater than a given threshold, the potential speaker change point is considered real; otherwise, it is removed (and the speaker model is updated again).

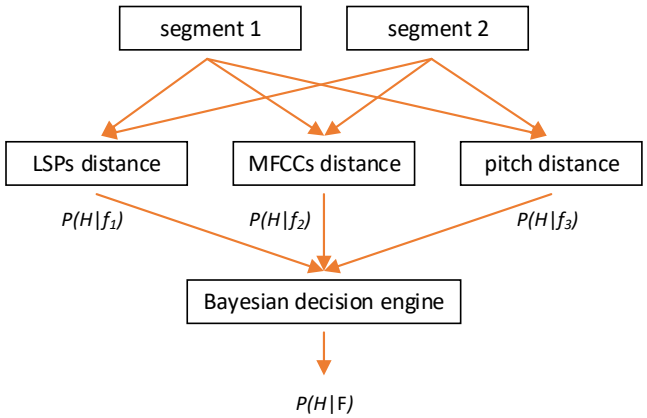

Figure 4.2: A Bayesian fusion method.

The achieved results show that the proposed algorithm is capable of performing SCP detection in real time. It can be approximately 6.5 times faster. The detailed results, as well as more information about the implementation of SCP detection and speaker tracking, can be found in the respective papers [78, 79].

4.3 XBIC

In [82], the authors proposed a novel (probability-based) distance metric designed for unsupervised metric-based speaker change point detection. This metric, called XBIC, computes cross probabilities between two neighboring windows (modeled with a Gaussian distribution) shifted along the speech audio stream. To detect the exact placement of the speaker change points, a two-pass algorithm is used as a complement to the metric. Both the metric and the detection algorithm were developed to be a part of a real-time SCP detection system and thus offer a fast and stable implementation.

Given two HMMs (one for each window) defined by $\lambda_1 = (A_1, B_1, \pi_1)$ and $\lambda_2 = (A_2, B_2, \pi_2)$, the data sets $\Theta_1 = \{\theta_1(1), ..., \theta_1(N_1)\}$ and $\Theta_2 = \{\theta_2(1), ..., \theta_2(N_2)\}$ can be generated by each of the two. The XBIC between two segments (windows) i and j can be then defined as [82]:

$$XBIC_{i,j} = P(\Theta_i|\lambda_j) + P(\Theta_j|\lambda_i) \ , \tag{4.7}$$

where

$$P(\Theta_i|\lambda_j) = \sum_{k=1}^{N_i} \log p(\theta_i(k)|\lambda_j) \quad \text{and} \quad P(\Theta_j|\lambda_i) = \sum_{k=1}^{N_j} \log p(\theta_j(k)|\lambda_i) \ . \tag{4.8}$$

In other words, the XBIC expresses the dissimilarity between two neighboring windows by computing the cross probabilities of each segment given the model of the other segment. Smaller values of XBIC represent more dissimilar segments hinting at potential speaker change point candidates.

Speaker change point detection is started by feature extraction. Specifically, 32-dimensional MFCCs (consisting of 16 static and 16 Δ coefficients) are computed. The following detection algorithm operates in two passes (similarly to [61]). In the first pass, two fixed-sized (neighboring) windows are shifted alongside the recording with a step of 0.1 seconds, and the XBIC between them is computed. Once a speaker change point is found (i.e., XBIC is smaller than a predefined threshold, and it is in local minimum), a second pass is initiated. In the second pass, two smaller fixed-sized (neighboring) windows slide around the detected speaker change point with a smaller step of 0.01 seconds to find its more precise placement (using XBIC). When the second pass is done, the detected speaker change point is considered final, and the algorithm proceeds from the next frame (it also returns to the first-pass stage).

The authors compared the XBIC approach with a standard BIC-based one on a broadcast news database. The results show that XBIC yielded at least comparable results on the Hub-4 [91] evaluation datasets. More details can be found in the given paper [82].

4.4 LLR-Based Approach

The main difficulty of the metric-based approaches to SCD detection (e.g., based on BIC or XBIC) is the need to define and fine-tune a threshold (or penalty factor), which is used to determine if a potential speaker change point is real or false. This generally means that the approaches perform well in the target domain (i.e., where the threshold was tuned), but their performance drops everywhere else. In [83], the authors tried to solve this issue by proposing a robust metric, which does not require such tuning. Their implementation is also suitable for real-time use.

Given two segments containing sequences of feature vectors, $X = \{x_1, ..., x_{N_x}\}$ and $Y = \{y_1, ..., y_{N_y}\}$ (where Z marks the concatenation of X and Y), the LLR can be computed to express the dissimilarity between the two segments by a hypothesis test. The null hypothesis (H_0) assumes that there is no change point between the two segments. The authors model the data Z using a GMM with two components. The estimates of the parameters (θ_Z) are computed by the expectation-maximization algorithm. At last, the log-likelihood of Z under the null hypothesis can be expressed as:

$$L_0 = \sum_{i=1}^{N_x} \log p(x_i|\theta_z) + \sum_{i=1}^{N_y} \log p(y_i|\theta_z) \ , \tag{4.9}$$

where $p(x|\theta)$ is the likelihood of a feature x given θ [83].

The other hypothesis, H_1, assumes the existence of a speaker change point between the two segments. In this case, θ_x and θ_y denote the estimates of the parameters of the Gaussian densities of sequences X and Y, respectively. The log-likelihood L_1 can be then defined as:

$$L_1 = \sum_{i=1}^{N_x} \log p(x_i|\theta_x) + \sum_{i=1}^{N_y} \log p(y_i|\theta_y) \ . \tag{4.10}$$

Finally, the log-likelihood ratio between the two segments can be computed as:

$$d_{LLR} = L_1 - L_0 \ . \tag{4.11}$$

Because the authors set the number of parameters of both models to the same value, the log-likelihoods are directly comparable. That means a positive d_{LLR} suggests a speaker change point. On the other hand, if the value is negative, both the segments should be from the same speaker. Note that there is no threshold tuning.

The speaker change point detection starts with the extraction of 24-dimensional MFCCs from the acoustic stream. The authors chose to implement a window-growing SCP detection approach [60, 143]. At first, a window of initial size is placed at the beginning of the stream. The algorithm assumes that a potential change point is located in the middle of this window (i.e., it divides the window into two same-sized segments). The LLR is computed between the two segments. If it is positive, the potential change point is considered real, the window is moved beyond the decoded change point, and it is also shrunk to 1 second. If the LLR is negative, the window is enlarged (by 1 second), and the procedure is repeated until a change

point is found or the window reaches its maximum allowed size (i.e., 20 seconds). In that case, the start of the window shifts to the right.

The authors compared their proposed approach with an approach based on BIC on the Hub-4 evaluation dataset [91]. The results show that their approach (without tuning) reached comparable performance with the best-tuned BIC-based approach. The detailed results can be found in the given paper [83].

4.5 Adapted GMMs

In metric-based SCP detection, one of the issues that can arise is caused by phonetic variation. Generally, if two (short) segments (windows) belong to the same speaker and at the same time, their phonetic content is different enough, a false change point may be produced by the decoding algorithm. In [69], the authors tackled this issue by proposing a new probabilistic SCP detection approach (based on their previous work [144]). Particular focus was given to the computational efficiency of the introduced algorithm.

The SCP detection approach first extracts 20-dimensional linear predictive cepstral coefficients from the audio stream. After that, two same-sized (2-second) adjacent windows (consisting of speech segments $X = \{x_1, ..., x_N\}$ [left window] and $Y = \{y_1, ..., y_N\}$ [right window]) are shifted alongside the recording (by 0.1 seconds). For each shift, two GMM-based speaker models (θ_X and θ_Y) are obtained from these two windows using a single-step Bayesian adaptation of an independent universal background model θ_{UBM} (a GMM trained on a large amount of data from different speakers). Note that the GMMs are used instead of single Gaussian models to address the phonetic variation issue described earlier. To compare the two windows, the authors use the following metric:

$$p = \log p(Y|\theta_X) - \log p(Y|\theta_{UBM}) + p(X|\theta_Y) - \log p(X|\theta_{UBM}) \ . \tag{4.12}$$

Its value is computed at each shift and if it is greater than a predefined threshold, a speaker change point is placed in between the two windows. The authors called their approach adapted model-based bilateral scoring-based speaker change detection.

The rest of the showcased paper [69] is focused on making the proposed approach as computationally efficient as possible. This process, the real-time settings, as well as detailed results, are all well-presented. The results show an improvement over the standard BIC-based SCP detection on broadcast data.

4.6 i-vectors

The authors of [52] proposed a novel metric-based approach to speaker change point detection designed for the meetings domain using the i-vectors. The i-vectors provide representation for each utterance in the form of a low-dimensional feature vector with a fixed length. The i-vectors were first introduced in [145] for the task of speaker verification, where they achieved excellent (state-of-the-art) results. Since

then, they have been successfully applied to many speech processing tasks, such as speaker recognition [146], speaker diarization [147], or language recognition [148, 149]. In the showcased paper, the authors applied the i-vectors to the task of speaker change point detection.

In [145], Total Variability (TV) space, which contains both the speaker and channel variabilities simultaneously (as opposed to joint factor analysis modeling [150], in which there are two distinct spaces defined – speaker space and channel space), was first introduced. This space is defined by a total variability matrix, which contains the eigenvectors with the largest eigenvalues of the total variability covariance matrix [52]. In this space, given an utterance, a speaker- and channel-dependent GMM supervector M (formed as a concatenation of means of all mixtures) can be modeled as:

$$M = m + Tw \ , \tag{4.13}$$

where m is the speaker- and channel-independent supervector (commonly taken from a large GMM known as the universal background model), T is a rectangular matrix of low rank (defining the TV subspace), and w is a random vector with a standard normal distribution whose components are the total factors [52]. This vector w is called i-vector. Note that the authors trained the UBM (8 Gaussian components) and TV matrix on the AMI Meeting Corpus [151] using the 12-dimensional MFCCs, and the extracted i-vectors are 100-dimensional.

Before the speaker change point detection starts, the input signal is preprocessed by an energy-based SAD, which filters out non-speech segments longer than 0.25 seconds. The authors opted for the window-growing approach to SCP detection [62]. The algorithm starts with a window of an initial size (2 seconds) that is placed at the start of the input. It assumes that the potential change point is placed exactly in the middle of this window. Two i-vectors, one representing the left part and one the right part of the window, are extracted. A cosine distance, defined (between two vectors w_1 and w_2) as:

$$CD_{w_1,w_2} = 1 - \frac{w_1 \cdot w_2}{||w_1||\,||w_2||} \ , \tag{4.14}$$

is computed between the two i-vectors. If the distance is greater than a predefined threshold, the potential speaker change point is considered final. If not, the window is enlarged (by 1 second), and the process is repeated until a change point is found or the size of the window exceeds the allowed maximum (10 seconds). In the latter case, the window is shifted to the right (by 2 seconds). If a change point is detected, the window size is reset to the initial value, and it is placed right after the detected change point. The algorithm stops when the window reaches the end of the input.

The authors evaluated their proposed i-vector-based approach on a subset of the AMI Meeting Corpus. They compared it with the traditional metric-based SCP detection approaches based on the BIC, GLR, KL2, and XBIC distances. As shown in [52], the i-vectors outperformed the other approaches in the meetings domain.

4.7 ASR-Based Approach

In [87], the authors focused on the development of an online speaker diarization system (with ASR). They comprehensively described the steps they took to make the conversion from their design to a final system operating in real time with low latency possible. The limitations of online diarization are well-discussed, and novel ideas were proposed by the authors. The most relevant (to this work) is the use of ASR block to produce inputs utilized by the speaker segmentation. Other ideas include, e.g., a new top-down algorithm for speaker clustering. In the paper, the authors examined two different architectures for an online speaker diarization system differing mainly in the placement of the ASR block.

In the first architecture, the ASR block is placed as the last component, after the speaker diarization block (composed of speaker segmentation and speaker clustering). In this case, the outputs of the diarization are used to improve the quality of speech transcription. In this architecture, the input stream is first preprocessed using a DNN-based SAD [32], which is followed by SCP detection based on BIC [152]. The diarization is concluded by clustering the obtained speaker-homogeneous segments into speaker clusters. Finally, N tandem ASR systems (where N is the number of suspected speakers) transcribe the input stream based on the information retrieved from speaker diarization (i.e., one ASR for each speaker cluster).

However, the authors were unable to improve the performance of the ASR using this architecture. The errors produced by the SAD and SCP detection (which was either making lots of false speaker change points or omitting real change points) influenced the transcriptions negatively. For this reason, the authors decided to modify their architecture.

In the modified architecture, the ASR block is placed in front of the speaker diarization block. In this case, the ASR eliminates most of the speech activity detection errors by accurately end-pointing the transcribed words (i.e., there are rarely words in non-speech segments). It also supplements the speaker segmentation as a speaker change point can only be placed between two neighboring words (boundary). The decision if a boundary is a speaker change point is made based on BIC between two adjacent segments (represented by cepstral features) around the boundary. If the segments are shorter than 2 seconds, T^2 criterion [152] is used instead of BIC. The diarization is concluded by speaker clustering (based on i-vectors and agglomerative or X-means clustering).

This architecture improved the performance and efficiency of the speaker diarization block (i.e., speaker segmentation and speaker clustering). The main drawback is that, in this case, the outputs of diarization are not used for improvements of the ASR. More information about the architecture, its components, and results can be found in the respective paper [87].

4.8 NN-Based Features

Real-time speaker change point detection designed for the conversations domain is the main focus of the work published in [89]. The authors approached this topic by proposing a novel metric-based approach whose main component is an NN-based speaker classifier transforming the input signal into speaker-discriminative features. The proposed system operates in three main phases – data preparation, feature extraction (from the NN), and SCP detection.

The first phase is initiated by two preprocessing steps – amplitude scaling and voice activity detection. The maximum of the absolute amplitude of the input signal is first scaled to 1. After that, voice activity detection is applied to remove all non-speech frames. The VAD algorithm computes short-term energy (to detect silence and environmental noises) and spectral centroid (to detect non-environmental noises, e.g., coughing) for each frame (given a short-term signal). A frame is considered speech if both the short-term energy and spectral centroid are greater than their predefined thresholds. The data preparation phase is concluded by the extraction of 39-dimensional MFCCs (including Δ and $\Delta\Delta$ coefficients) normalized per speaker. Finally, the authors form longer features (100 ms) by concatenating 10 adjacent frames to provide the NN with more context information. The longer features are shifted by 3 frames (see Fig. 4.3 for an illustration).

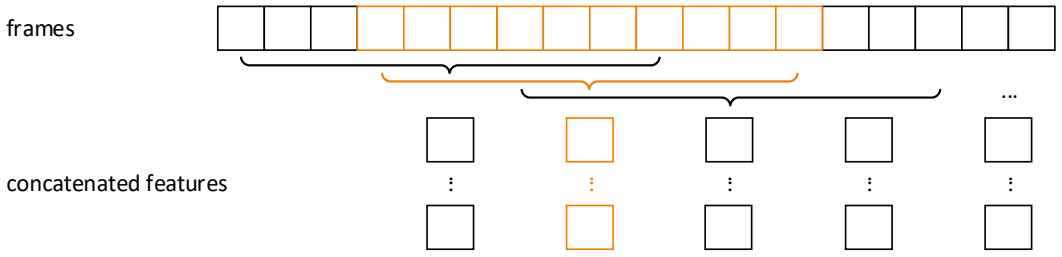

Figure 4.3: An example of the 10-frame concatenation with a step of 3 frames.

The authors trained a fully connected feed-forward neural network for the task of text-independent speaker classification. The NN was trained using data taken from the TIMIT corpus [153]. Specifically, recordings of 200 male speakers were utilized. The hyper-parameters of the NN were set to:

- 1 hidden layer with 200 neurons;
- sigmoid activation function (all layers);
- output layer with 200 neurons (i.e., the number of speakers).

During decoding, for each input feature vector (390-dimensional) fed into the NN, a 200-dimensional output feature vector (with values between 0 and 1) is obtained.

In the SCP detection phase, two fixed-sized adjacent windows (different sizes [0.5, 1, and 2 seconds] were evaluated by the authors) are shifted (by the window size) alongside the recording. For all frames in each window, the speaker-discriminative

features are first extracted from the NN. The distance computed between the two adjacent windows (i and j) is the p-norm distance defined as:

$$d'_{i,j} = \left(\sum_{k=1}^{K} \left(|\overline{d}_i - \overline{d}_j|^p \right) \right)^{\frac{1}{p}} , \tag{4.15}$$

where K is the dimension of one of the (NN) feature vectors (i.e., 200), d_i and d_j are the extracted (NN) feature matrices (with dimensions: window size $\times$ 200), and $\overline{d}_i$ and $\overline{d}_j$ are their mean vectors (i.e., 200-dimensional). If the computed distance is greater than a given threshold, a speaker change point is produced. The authors achieved the best results by using Euclidean distance (i.e., by setting the value of p to 2).

More information about the showcased approach and the results obtained on artificial data taken from the TIMIT corpus can be found in the respective paper [89].

4.9 Deep Speaker Vectors

In [55], the authors focused on crafting a new metric-based SCP detection approach suitable for scenarios with fast speaker changes. These scenarios are much harder to segment because most of the metric-based techniques usually require the two sliding windows to have a significant amount of input context to compute the distance metric reliably (i.e., the fast changes end up as errors). The authors approached this issue by employing a deep neural network to extract deep speaker vectors (d-vectors [154]) to represent the speaker characteristics. These d-vectors are highly speaker-discriminative features, which are computed for each frame. Their application, experimentally confirmed by the authors, allows for using shorter windows for distance calculations.

A fully connected feed-forward deep neural network needs to be trained to extract the d-vectors. It was trained on data of 1,000 randomly selected speakers (gender-balanced, >10 minutes per speaker) from the Fisher corpus [155]. The data was first preprocessed by energy-based VAD, which removed all the non-speech segments. The DNN was trained using the following hyper-parameters:

- 4 hidden layers;
- 200 neurons per hidden layer;
- sigmoid activation function;
- output layer with 1,000 neurons (i.e., the number of speakers).

For input features, the authors followed their previous work [156]:

- 40-dimensional Filter Bank Coefficients (FBCs);
- computed using 20 ms frames of the signal with frame shifts of 10 ms;

- concatenation of n previous frames, the current frame, and n following frames (n is not given in the paper, 10 in [156]);
- computed, as shown in Sect. 3.5 for MFCCs without the last two steps.

After the training of the DNN, the d-vectors can be extracted from any of the hidden layers. However, the deeper layers produce more speaker-discriminative features [55]. For this reason, the authors extract the d-vectors from the very last hidden layer.

The proposed metric-based SCP detection algorithm operates in several steps. First, the 40-dimensional FBCs are computed from the preprocessed (VAD) audio stream. Next, the d-vectors are extracted by feeding the FBCs into the DNN (i.e., a frame-sequence of d-vectors representing the audio stream is obtained). After that, two fixed-sized neighboring windows are shifted along the d-vector sequence, and the distance between them is computed. Each of the windows is represented by a deep speaker vector calculated as a mean of the d-vectors belonging to the respective window. A cosine distance is applied to compute the distance score between the two deep speaker vectors. Finally, the shifting of the two windows along the whole d-vector sequence generates a distance score curve. The change points are then placed in the spots of a local minimum if the minimum is lower than a predefined threshold.

In the experimental evaluation, the authors compared the performance of d-vectors with traditional distance metrics (e.g., BIC, GLR, KL2) in the fast speaker change scenarios, and they also explored different lengths of the shifting windows. The results show that the d-vectors had a great speaker-discriminative ability even for segments with a duration of 0.1 seconds (10 frames), where all other metrics failed. The ideal length of the window was found to be between 0.05 to 0.1 seconds for the task of fast SCP detection. More detailed results are available in the respective paper [55].

4.10 Segmentation in Online Diarization

The authors of [90] explored different approaches to online SCP detection and their effects on the performance of an online speaker diarization system. While the initial placement of the speaker change points is not that crucial in offline diarization (i.e., it can be improved in the final resegmentation stage), it is pivotal for online diarization (where such resegmentation is typically not possible). The authors employed their i-vector-based diarization system (altered to operate in a left-to-right mode suitable for online processing [90]) and mainly studied two different SCP detection approaches proposed in their previous works [73, 157].

The first approach, a standard metric-based one, was proposed in [157]. The SCP detection is run in two consecutive passes. In the first pass, the GLR (see Sect. 4.1, Eq. (4.1)) is computed between two 2-second neighboring windows shifting alongside the recording with a step of 0.1 seconds. If the GLR lies in local maximum and it is greater than a predefined threshold, a speaker change point is produced. In the second pass, longer segments are split in spots, where the GLR has the highest values. Note that this approach requires VAD to filter out non-speech segments.

A convolutional neural network employed as a regressor is the main component of the second approach proposed in [73]. The CNN was trained on spectrograms of the acoustic input data. This data was labeled using a fuzzy labeling technique (instead of the binary one) developed by the authors. An approximately 0.6-second window around the actual change point is labeled non-zero. The value of the labels linearly increases closer to the change point, where it is set to 1. A depiction of fuzzy labeling can be seen in Fig. 4.4. The CNN is comprised of 3 convolutional layers (with ReLU activation function), each followed by a max-pooling layer and a batch normalization layer, and two fully connected layers. The output layer has one neuron with a sigmoid activation function (i.e., the output is between 0 and 1). It represents the likelihood of the presence of a speaker change point. During the SCP detection, a non-maximum suppression with a 0.5-second window is applied to find maximum peaks in CNN output (i.e., potential change points). The final speaker change points are then the peaks greater than a predefined threshold (0.5). Note that in this case, voice activity detection is not needed [90].

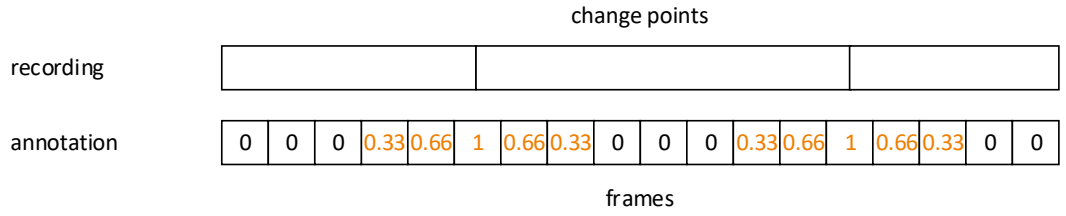

Figure 4.4: An example of the fuzzy labeling technique. In this case, a two-frame window around the change points is labeled as speaker change. The label values linearly decrease further from the actual change points.

For comparison purposes, the authors also implemented a fixed-length segmentation, where the data is divided into 2-second segments with a 1-second overlap.

The authors reported their findings on the CALLHOME American English corpus of telephone speech. They employed the above-described SCP detection approaches and compared their performance in both offline and online i-vector-based speaker diarization systems. The results confirm that precise SCP detection is more vital for the online diarization because, in an offline setting, the placement can be corrected by resegmentation [157] (i.e., the fixed-length segmentation performed on a similar level in offline diarization but failed online). The GLR- and CNN-based approaches yielded fairly comparable results. More details are available in the respective paper [90].

5 Proposed Speech Activity Detection Approach

The final approach to speech activity detection was proposed in a series of consecutive experiments, all described and heavily discussed within this chapter. The majority of this designing process was covered in [1–3], and portions of the respective papers were directly utilized in this thesis. This chapter thus describes evaluation metrics, training and development data, experimental evaluation of all steps taken, evaluation on standardized QUT-NOISE-TIMIT corpus, evaluation in real speech transcription system, and at last, it sets the final SAD approach.

5.1 Evaluation Metrics

In total, seven different commonly utilized metrics were employed for the evaluation of speech activity detection. These metrics can be grouped into three main subsets, each focusing on different aspects of SAD: overall accuracy metrics, change point quality metrics, and performance metrics.

5.1.1 Overall Accuracy Metrics

The main focus of this group of metrics is the accuracy of newly defined speech and non-speech segments on a frame-level (i.e., the recording is treated as a sequence of speech and non-speech frames). In this case, each frame is considered independent, and only a direct comparison between the reference frame and the corresponding decoded frame (frame pair) is evaluated. If the frame pair is matched, it is considered as a hit; otherwise, it counts as a miss (see an example in Fig. 5.1). For this task, four closely related metrics were applied.

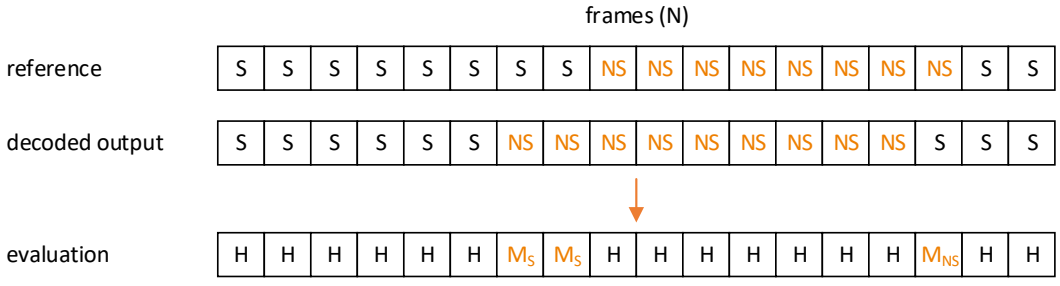

Figure 5.1: An example of utilized frame-based evaluation. S marks speech frames, NS non-speech ones while H expresses hits, and M misses.

The first metric, Frame Error Rate (FER), is defined as follows:

$$FER[\%] = \frac{M}{N} * 100 \ , \tag{5.1}$$

where M is the number of non-matching frames in reference and decoded output, and N is the total number of frames in reference.

The following two metrics are Miss Rate (MR) and False Alarm Rate (FAR) [8]. They represent relevance metrics, specifically false negatives and false positives (the rest of the relevance metrics is not reported as they are complementary to the presented ones).

Miss rate (false negatives) can be expressed as:

$$MR[\%] = \frac{M_{\text{speech}}}{N_{\text{speech}}} * 100 \ , \tag{5.2}$$

where M_{speech} is the number of speech frames classified as non-speech, and N_{speech} is the total number of speech frames in reference.

False alarm rate (false positives) is defined as:

$$FAR[\%] = \frac{M_{\text{non-speech}}}{N_{\text{non-speech}}} * 100 \ , \tag{5.3}$$

where $M_{\text{non-speech}}$ is the number of non-speech frames classified as speech, and $N_{\text{non-speech}}$ is the total number of non-speech frames in reference.

The last metric, Half-Total Error Rate (HTER), is an official metric of QUT-NOISE-TIMIT [44] evaluation protocol. As such, within this thesis, it is only reported for the comparison of results on the respective QUT-NOISE-TIMIT corpus. It is defined as an equal-weighted average of MR and FAR:

$$HTER[\%] = \frac{MR + FAR}{2} \ . \tag{5.4}$$

Finally, the optimal SAD approach should minimize the miss rate while keeping the false alarm rate relatively low. The reason is that the following speech processing system (e.g., SCP detector or speech transcriber) should get all speech frames possible with only a limited amount of non-speech events added (i.e., limiting transitive errors by not omitting any speech).

5.1.2 Change Point Quality Metrics

Change point quality metrics offer an alternative view on the performance of SAD. Instead of a frame-based evaluation, they explore the recording as a sequence of consecutive speech and non-speech events, and more specifically, as the name suggests, they focus on the accuracy of detected (computed) change points between these events. For this task, two distinct metrics, F-measure and $\delta_{2/3}$, were employed.

To define these two metrics, the detected and the reference change points have to be aligned at first [158]. The bidirectional search for the nearest neighbor (between the detected and reference change points) can do this. A detected change point i, and the reference change point j can be considered as a Hit (H) only if:

1. the nearest neighbor of detected change point i is the reference change point j;

2. the nearest neighbor of reference change point j is the detected change point i;

3. the distance between i and j is smaller than a defined threshold (commonly set to 1 second).

The errors are then marked as Insertions (I) and Deletions (D). If a detected change point does not match any of the reference change points, it is tagged as insertion. Similarly, if a reference change point is not matched by any of the detected change points, it is marked as deletion. For an example, see Fig. 5.2. Note that within this thesis, the threshold was set to standard 1 second.

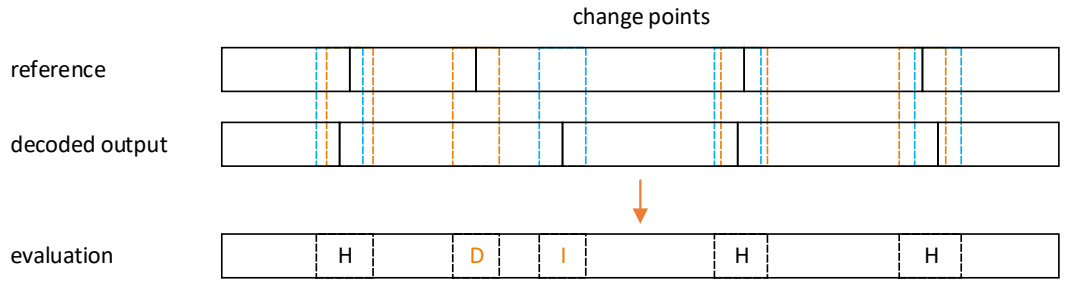

Figure 5.2: An example of aligned detected and reference change points (black lines). H marks hits, I insertions and D stands for deletions. Orange and blue dashed lines indicate the reference and decoded threshold boundaries, respectively.

Given the values of hits, insertions and deletions, Precision (P) and Recall (R) can be expressed. Precision is defined as a ratio between the number of correctly detected change points and the number of detected change points:

$$P[\%] = \frac{H}{H+I} * 100 \ , \tag{5.5}$$

while recall is expressed as a ratio between the number of correctly detected change points and the number of change points in reference:

$$R[\%] = \frac{H}{H+D} * 100 \ . \tag{5.6}$$

Precision and recall are in a contradictory relationship with each other (i.e., when one improves the other one worsens). For this reason and to express the performance with only one value, F-measure (F) is defined. It has a local maximum and can be computed from precision and recall as follows:

$$F[\%] = \frac{2 * R * P}{R+P} \ . \tag{5.7}$$

Given the correctly detected change points (hits), it is also possible to calculate an error value for each hit (in seconds) and sort all the hits according to these calculated values in ascending order. In this work, $\delta_{2/3}$ was utilized. It expresses (in seconds) the maximal error of the alignment for the first two-thirds of the sorted (best) hits. Note that $\delta_{2/3}$ should be as low as possible to provide further speech processor with precisely defined speech segments.

5.1.3 Performance Metrics

The last set of metrics monitor the performance of SAD in an online environment. Two different metrics, Latency (L) and Real-Time Factor (RTF), were utilized.

The former one is defined as an average time between the detected change point, and the moment the decoder outputs the change point label (see Fig. 5.3 for an illustration). Forcing this value to be as low as possible is a crucial part of online necessities for real-time deployment.

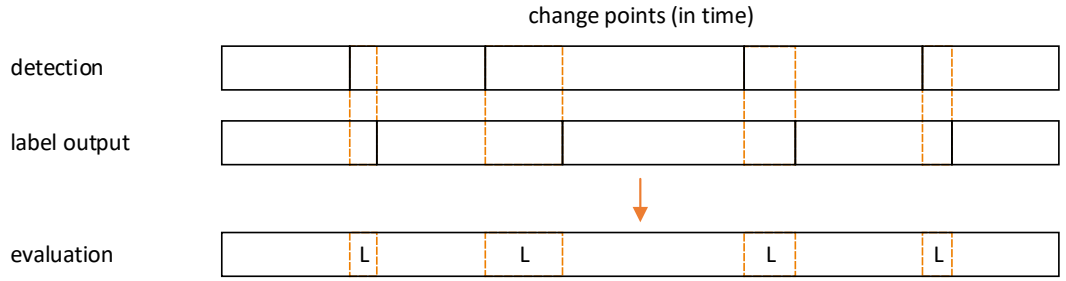

Figure 5.3: An example of latency calculation. The upper row displays the actual change point placements decided by the decoder (black lines). The middle row marks the moments the decoder outputs the labels (black lines), and finally, the bottom row shows the latencies for each change point, which are then averaged.

The latter metric is the real-time factor, and it expresses the speed of decoding:

$$RTF = \frac{PT}{T} \ , \tag{5.8}$$

where PT is the processing time of decoding, and T is the duration of the recording. If the RTF is smaller than 1, the decoder can operate in real time. Therefore, the smaller the value is, the faster the decoding is.

5.2 Data Used

For training, in total, 67 hours of recordings have been gathered and utilized. The speech is represented by 30 hours of clean speech recordings of English and several Slavic languages (Czech, Slovak, Polish, Russian, and Croatian). These recordings originally served as training data for speech transcription systems. The non-speech is modeled by 30 hours of music of different genres with the addition of 7 hours of non-speech events/noises. Lastly, the annotations were done automatically, speech label for clean speech utterances and non-speech one for everything else.

The data used for development consists of 6 hours of TV and radio recordings in several Slavic languages (Czech, Slovak, Polish, and Russian). It contains not only clean speech segments but also segments with music, background noises, jingles, and advertisements. Annotations of this data were obtained in a two-step process. At first, speech and non-speech labels were produced automatically by the baseline DNN-based SAD approach introduced in Sect. 5.3. These obtained labels were then

corrected and fine-tuned by hand. In total, 70% of all frames are marked as speech ones. An example of annotation can be seen in Fig. 5.4. Finally, an overview of the data used is presented in Table 5.1.

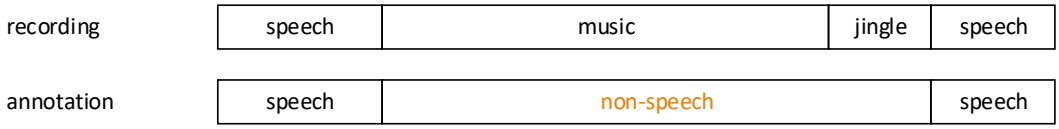

Figure 5.4: An example of annotation of a development recording.

Table 5.1: An overview of utilized data for SAD.

dataset	recordings	hours	change points	speech
training	15,010	67	0	45%
artificial training	14,483	30	0	63%
modified artificial	8,797	30	8,797	62%
development	24	6	337	70%

5.3 Baseline DNN-Based Approach

The baseline speech activity detection approach employed a feed-forward deep neural network with a binary output (speech or non-speech) as a classifier (i.e., without any smoothing). The initial hyper-parameters of the DNN were set to:

- 5 hidden layers;
- 128 neurons per hidden layer;
- ReLU activation function;
- mini-batches size of 1,024;
- 0.08 learning rate;
- 10 epochs.

The features extracted from training data were:

- 39-dimensional log filter bank coefficients;
- computed using 25 ms frames of the signal with frame shifts of 10 ms;
- concatenation of 25 previous frames, the current frame, and 25 following frames (i.e., a 0.5-second context window);

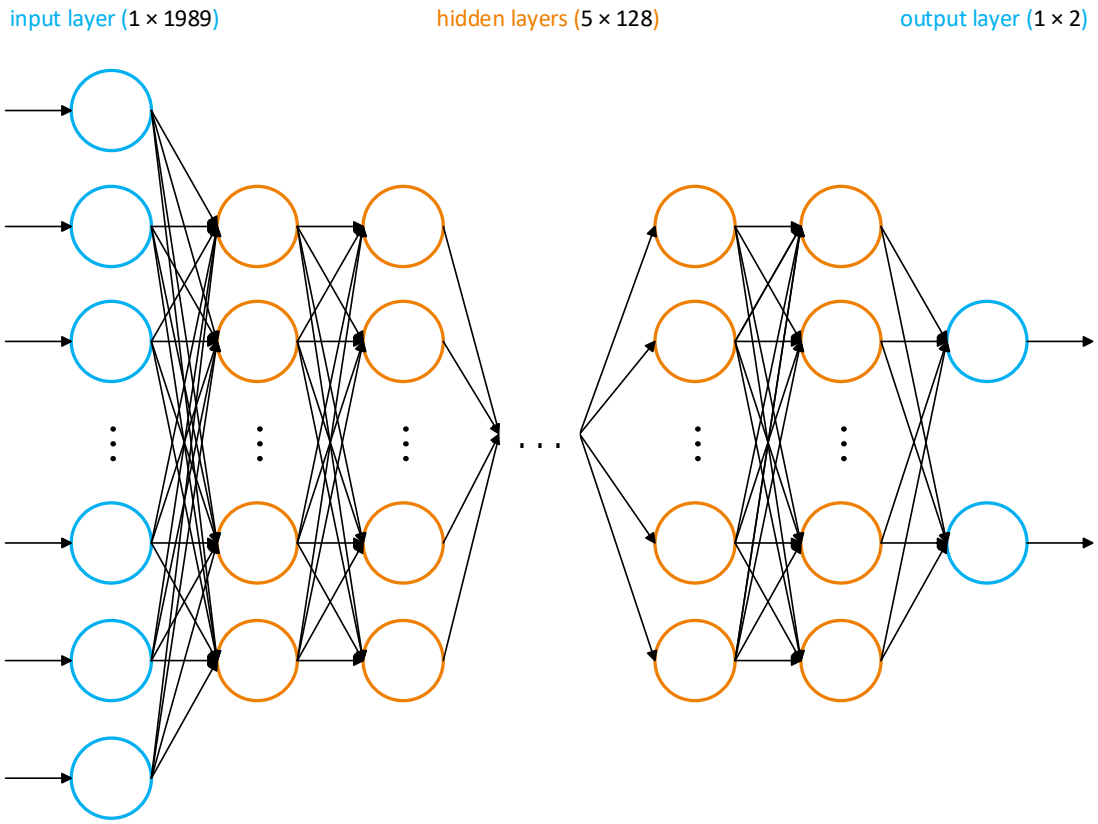

Figure 5.5: A feed-forward DNN used in SAD.

- local normalized within a one-second window.

Finally, an illustration of the trained DNN is in Fig. 5.5.

The performance of the baseline approach is summarized in Table 5.2 (see its first row). It is evident that it missed approximately 4% of speech segments. This fact affects the accuracy of the possible speech transcription system negatively, as the segments incorrectly marked as non-speech would not be transcribed. Another problem of the baseline detector was the time precision of the change-point detection: the achieved value of $\delta_{2/3}$ was 0.42 seconds. This is also due to the fact that it is sometimes hard even for human annotators to determine the exact frame where a state change occurs. The baseline detector also produced a high number of false non-speech segments with a very short duration of one or two frames.

Note that each of the presented deep neural networks (i.e., for all SAD experiments) was trained on GPU using the torch framework[1] unless stated otherwise. The training scripts are available at the author's GitHub[2] for everyone to download and see.

[1] http://torch.ch/
[2] https://github.com/1shark1/nnet/

Table 5.2: Summarized results of the proposed SAD approach described in detail in Chap. 5.

approach	FER [%]	MR [%]	FAR [%]	F [%]	$\delta_{2/3}$ [s]
baseline DNN-based	4.7	3.7	7.1	0.3	0.42
+ basic smoothing	2.9	2.2	**4.7**	28.5	0.27
+ artificial training data	3.1	**0.3**	10.1	41.3	0.34
modified artificial data + context-based smoothing	**2.4**	0.5	7.2	**52.7**	**0.26**

5.4 Smoothing the Output from DNN

As mentioned in the previous section, the baseline detector classified every input frame independently. On the other hand, every speech or non-speech segment usually lasts for at least several frames. Therefore, the following efforts were focused on smoothing the output from the DNN. For this purpose, weighted finite-state transducers were utilized using the OpenFst library[3].

The resulting scheme consists of two transducers. The first models the input signal (see Figure 5.6). The other one is the transduction model and represents the smoothing algorithm (see Figure 5.7). It consists of three states. The first state, denoted by 0, is the initial state. The transitions between states 1 and 2 emit the speech/non-speech labels and are penalized by penalty factors P1 and P2, respectively. Their values (500 and 500) were tuned and determined in several experiments on a different dataset.

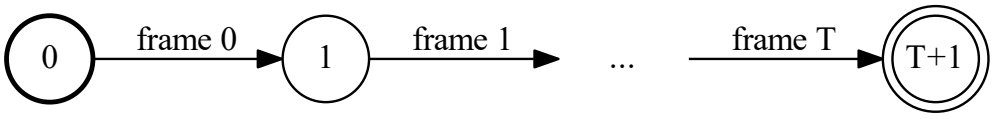

Figure 5.6: A transducer modeling the input signal for SAD.

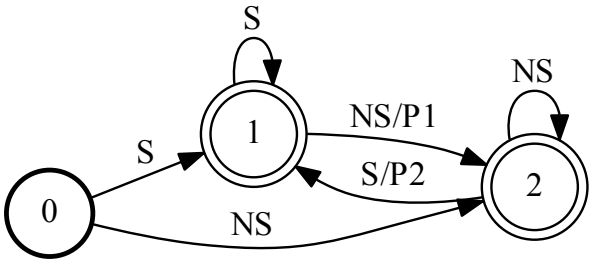

Figure 5.7: A transducer representing the basic smoothing model for SAD.

[3]http://www.openfst.org/twiki/bin/view/FST/WebHome

Given the two described transducers, the decoding process is performed using the on-the-fly composition of the transduction and the input model of unknown size. This is possible since the input is considered to be a linear-topology, unweighted, epsilon-free acceptor. After each composition step, the shortest-path (considering tropical semi-ring) determined in the resulting model is compared with all other alternative hypotheses. When a common path is found among these hypotheses (i.e., with the same output label), the corresponding concatenated output labels are marked as the final fixed output. Since the rest of the best path is not known with certainty, it is denoted as a temporary output (i.e., it can be further refined).

The results obtained with the aid of the DNN-based approach with smoothing are summarized in the second row of Table 5.2. They show an overall significant boost in performance. For example, F-measure improved from 0.3% to 28.5%, MR was reduced from 3.7% to 2.2%, and the value of $\delta_{2/3}$ improved noticeably from 0.42 seconds to 0.27 seconds. Also, note that the performance on clean speech and non-speech (music) data was reported in detail in [1].

5.5 Using Artificial Training Data

The level of MR yielded so far, i.e., around 2%, would still lead to a small increase in the Word Error Rate (WER) of a transcription system (e.g., from 13% to 14%), as the speech frames incorrectly classified as non-speech would be omitted from transcription. Upon closer inspection, the detector specifically misclassified the speech segments with background noise. The reason for this behavior is that the speech data used for DNN training so far were recorded only in clean conditions (i.e., without any background noise).

Hence in the next step, the goal was to employ training data containing non-speech events, such as music or jingles in the background. Due to the lack of such annotated data, an artificial dataset created by mixing 30 hours of clean speech with non-speech recordings was constructed. For this purpose, a larger set of non-speech recordings of a total length of 100 hours was prepared first. After that, every speech recording was mixed with a randomly selected non-speech recording from the prepared set. Note that every non-speech recording used for mixing had to have the same or longer duration than the given input speech recording (the selected non-speech recording was trimmed to match the length of the speech recording) and its volume was increased or decreased to match the desired level of signal-to-noise ratio (which was also selected randomly from an interval between -30 dB and 50 dB).

The labeling of this artificial data was carried out automatically: when the SNR of the recording was higher than a defined threshold of 0 dB, the recording was marked as containing speech. In the opposite case, the recording was labeled as non-speech. The whole process of artificial data preparation and annotation is shown in Fig. 5.8. Lastly, more information about the data is presented in Table 5.1 (see the second row).

The results after using only these 30 hours of mixed training data are shown in the third row of Table 5.2. It is evident that this approach led to an increase in

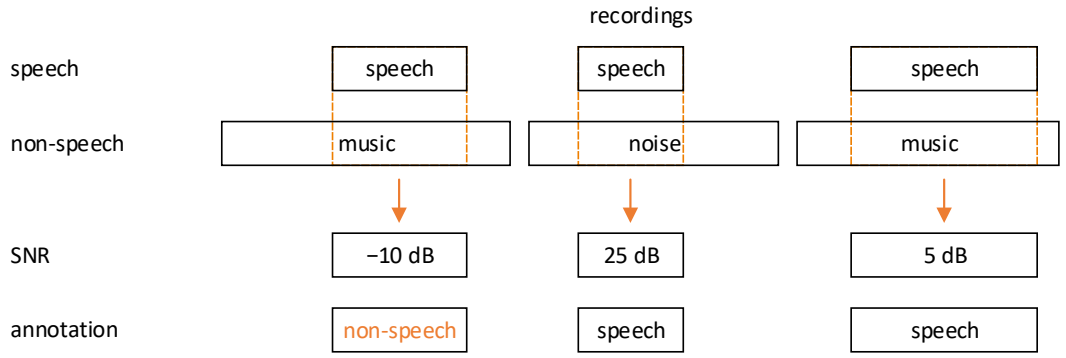

Figure 5.8: An illustration of SAD artificial data mixing.

F-measure and a significant reduction in MR from 2.2% to 0.3%. Unfortunately, these improvements are all accompanied by an increase in FAR and, even more importantly, an increase in $\delta_{2/3}$ from 0.27 seconds to 0.34 seconds. Due to these issues, a further refinement of the smoothing algorithm was investigated.

5.6 Improved Context-Based Smoothing

The proposed refinement of the smoothing scheme is depicted in Fig. 5.9. In this case, both the speech and non-speech events are represented as sequences of three states, where the first and third states (the outer states) model the context. Similarly to smoothing without any context, the penalties are defined just for transitions between the speech and non-speech events, i.e., for transition a) from the end state of speech (*stop_S*) to the start state of non-speech (*start_NS*), and b) from the end state of non-speech (*stop_NS*) to the start state of speech (*start_S*). Their values were fine-tuned on a different dataset.

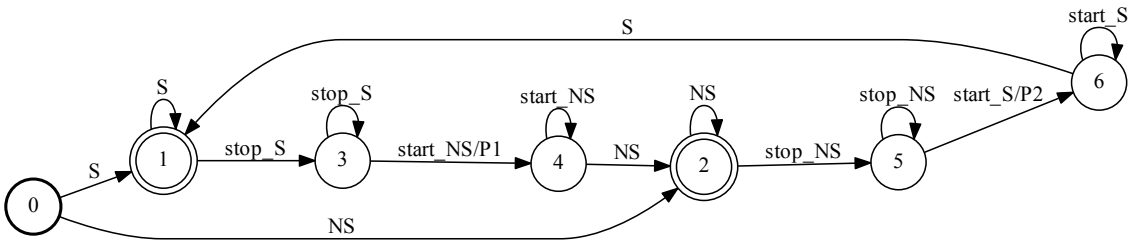

Figure 5.9: A transducer representing the context-based smoothing model for SAD.

To prepare training data containing transitions between speech and non-speech events, the dataset from Sect. 5.5 was modified. At first, two recordings were chosen randomly from the artificial training set: one speech and one non-speech. After that, these two recordings were joined in random order. The resulting recording then contained one of the two possible transitions (i.e., from speech to non-speech or from non-speech to speech) and was annotated automatically as follows:

1. The number of transition frames was derived from the input feature context window (25-1-25).

2. Only the 50 frames at the inner boundary of the two joined recordings were annotated as transitional, i.e., using 25 labels *stop_S* followed by 25 labels *start_NS* or 25 labels *stop_NS* followed by 25 labels *start_S*.

3. All other frames were marked as either speech or non-speech.

An example of two recordings and their annotations is available in Fig 5.10. Further information about the newly defined dataset is gathered in Table 5.1 (see the third row).

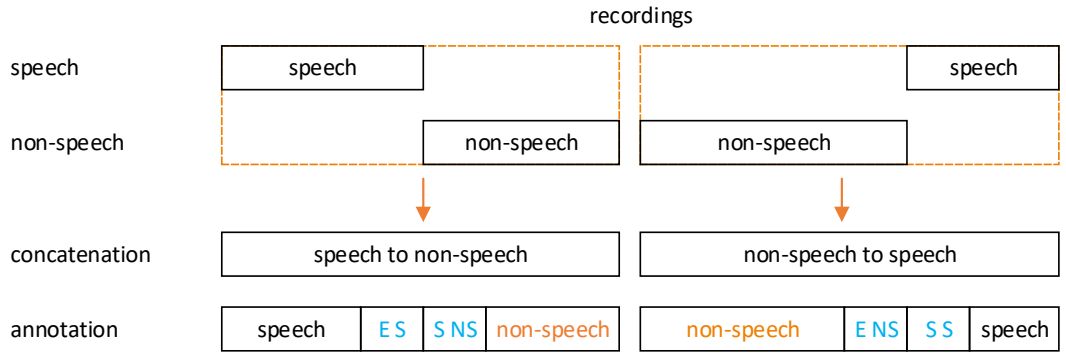

Figure 5.10: An example of the creation and annotation of two newly concatenated recordings. The first one (left) illustrates the transition from speech to non-speech, where E S marks the end of speech while S NS means the start of non-speech. The other one (right) shows an opposite transition, from non-speech to speech, where E NS expresses the end of non-speech and S S stands for the start of the speech.

Finally, the last change associated with the integration of context-based smoothing lies in the deep neural network model. Instead of the original two output neurons, there are now 6 (speech, non-speech and 4 transitional ones: *start_S*, *stop_S*, *start_NS*, *stop_NS*) to match the smoothing scheme and annotation style of data. The rest of the DNN remained unchanged (see Sect. 5.3 for exact values of hyper-parameters).

The results of the experiment with the context-based smoothing (see the fourth row of Table 5.2) show that this approach addresses the issue of an increase in $\delta_{2/3}$, which has emerged due to the use of the artificial training data (see the third row of Table 5.2). The value of $\delta_{2/3}$ was reduced from 0.34 seconds to 0.27 seconds. At the same time, a significant decrease in the FAR, an increase in F-measure, and only a slight decrease in MR by 0.2% were achieved when compared to the previous experiment. After achieving these results, the focus for the next work shifted towards tuning the hyper-parameters of the DNN.

5.7 Tuning of Hyper-Parameters

The tuning of hyper-parameters is crucial, although a laborious part of designing a system based on deep neural networks. It hugely influences the results the target system (i.e., in this case, the speech activity detector) can achieve. There are many tunable hyper-parameters (such as the number and width of hidden layers, activation function, or the number of epochs) that need to be considered during the design. Note that the proposed approach, as described in Sect 5.6, was utilized for all the following experiments exploring several hyper-parameters.

5.7.1 Width of Hidden Layers

The first examined hyper-parameter was the width of the hidden layers (see Fig. 5.11 for an illustration). It affects not only the modeling capabilities of the DNN and consecutively the results but also the computational demands during both training and real-time use.

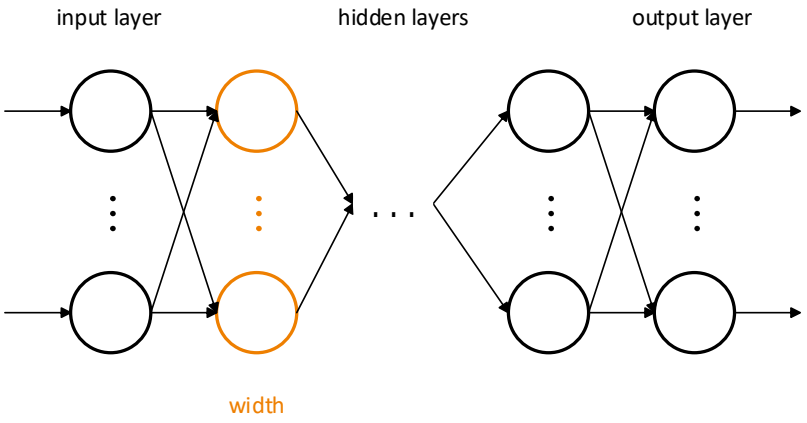

Figure 5.11: An illustration of the width of a hidden layer of DNN.

In total, seven different deep neural networks were trained and experimentally evaluated. The number of neurons per hidden layer ranged from 64 to 256, with a step of 32 neurons. The rest of the hyper-parameters remained the same as described in Sect. 5.3 (and Sect. 5.6, i.e., six output neurons).

The achieved results are summarized in Table 5.3, and two main trends can be pointed out. First, the MR improved with additional neurons, while FAR and F-measure worsened. Second, the deterioration in the FAR was more significant than the improvements in MR, resulting in worse FER with additional neurons.

More specifically, although the smaller DNNs (i.e., 64 and 96 neurons) yielded better results in FAR and F-measure than the original 128-neuron network, the increase in MR and $\delta_{2/3}$ was significant enough to result in a worse performance of following speech transcriber. The improvements in MR of bigger networks (160, 192, 224 and also 256 neurons per layer) were fairly negligible, and their performance was crippled by worsened results in FER, FAR as well as $\delta_{2/3}$. F-measure was also worse for the majority of them except for the DNN with 160 neurons per layer. The

trade-off was not worth it for any of the DNNs. To sum it up, the original network (128 neurons per layer) was the best compromise to use in the final speech activity detector deployed as a speech preprocessor, e.g., in speech transcription.

Table 5.3: Results of experimental evaluation focusing on the width of hidden layers.

width of hidden layers	FER [%]	MR [%]	FAR [%]	F [%]	$\delta_{2/3}$ [s]
64	**2.2**	0.8	**5.8**	56.4	0.28
96	2.3	0.7	6.3	**56.5**	0.28
128	2.4	0.5	7.2	52.7	**0.26**
160	2.6	0.4	8.1	54.3	0.30
192	2.9	**0.3**	9.5	49.5	0.32
224	3.0	**0.3**	10.0	48.6	0.34
256	2.8	0.4	9.1	49.8	0.30

5.7.2 Number of Hidden Layers

The number of hidden layers was the second explored hyper-parameter (an illustration is available in Fig. 5.12). It has similar effects on the performance of SAD as the width of the hidden layers. With additional layers, the modeling capabilities should improve, although the computational and training data demands increase as well. A deeper neural network may also be too complex for the issued task, and it may result in overfitting to training data and thus, worse performance. Vanishing gradient might also become an issue when utilizing very deep networks unless some prevention is implemented (e.g., batch normalization or identity addition).

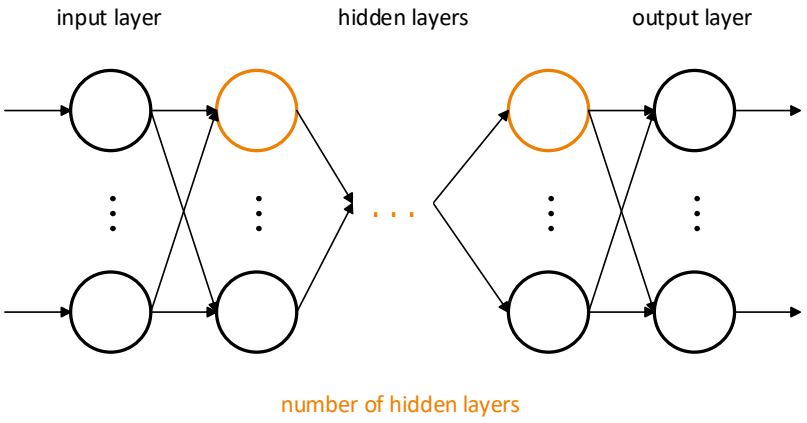

Figure 5.12: An illustration of the number of hidden layers of DNN.

Within this experiment, three distinct deep neural networks with a different number of hidden layers (4, 5, and 6) were trained. As previously, the rest of the hyper-parameters remained unchanged.

The results are presented in Table 5.4. From them, it is evident that the DNN with 5 hidden layers performed the best. The deeper DNN underperformed in all metrics except for a slight (negligible) improvement in MR, and it was too complex for the task with the data used. The more shallow DNN was slightly more interesting than its deeper counterpart. However, the improved F-measure was overweighted by deterioration in FAR and $\delta_{2/3}$. Note that even more shallow DNNs were explored, but they yielded significantly worse results.

Table 5.4: Results of the experiment focused on the number of hidden layers.

number of hidden layers	FER [%]	MR [%]	FAR [%]	F [%]	$\delta_{2/3}$ [s]
4	**2.4**	0.5	7.5	**55.3**	0.27
5	**2.4**	0.5	**7.2**	52.7	**0.26**
6	2.7	**0.4**	8.3	52.1	0.31

5.7.3 Activation Functions of Neurons

The activation function (or transfer function) of neurons is another tunable parameter of DNNs that can influence the performance of SAD. Within this section, three distinct, commonly utilized activation functions were experimentally tested, specifically, sigmoid, hyperbolic tangent, and ReLU transfer functions (see Fig. 5.13 for an illustration). A properly selected activation function (and learning rate) can improve not only the performance of SAD but also the speed of convergence during the training of the DNN.

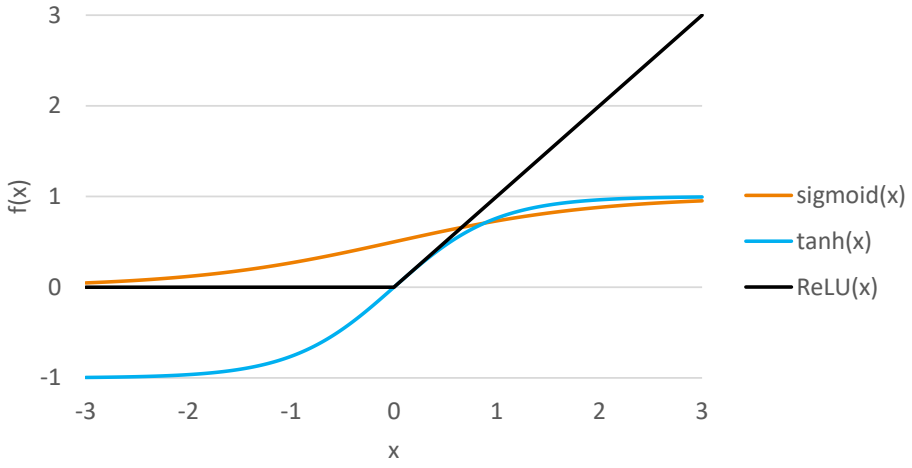

Figure 5.13: An overview of various activation functions.

The results are gathered in Table 5.5. They clearly show that the sigmoid function performed underwhelmingly (i.e., it might need more training epochs to converge). On the other hand, the experiments conducted using ReLU and hyperbolic

tangent activation functions yielded somewhat comparable results. However, the slightly lower MR and $\delta_{2/3}$ were in favor of the ReLU activation function.

Table 5.5: Results of experimental evaluation focused on the use of different activation functions.

activation function	FER [%]	MR [%]	FAR [%]	F [%]	$\delta_{2/3}$ [s]
sigmoid	2.8	0.4	8.9	47.2	0.27
tanh	**2.4**	0.6	**7.2**	**55.7**	0.28
ReLU	**2.4**	**0.5**	**7.2**	52.7	**0.26**

5.7.4 Context Window Size

Another set of experiments were focused on determining the ideal size of the input feature window (illustration in Fig. 5.14). Within this evaluation, five different context window sizes ranging from 0.1 seconds (the input feature vector is formed as a concatenation of 5 preceding frames, the current frame, and 5 following frames [5-1-5]) to 1.6 seconds (80-1-80) were evaluated. An illustration of the 5-1-5 context window is shown in Fig. 5.15. Additional context should results in a better performance of SAD until the context becomes irrelevant (or overfitting occurs). However, improved performance should be accompanied by worse RTF and latency. Note that the settings of other hyper-parameters remained the same.

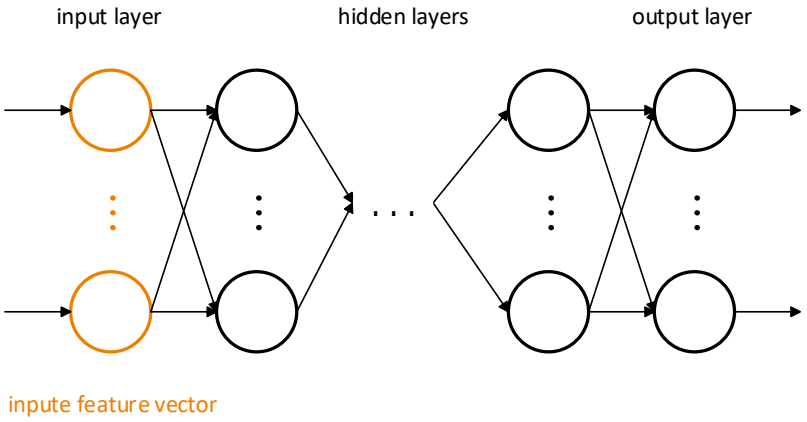

Figure 5.14: An illustration of the context window size of DNN.

Table 5.6 presents all the results. They show that the performance of SAD improved with additional context until the 0.5-second context, where it plateaued and started to degrade. The richer contexts resulted in a significantly worse $\delta_{2/3}$ as well as FAR (for the 0.7-second context) and MR (for the 1.6-second context). These degradations were most likely caused by overfitting to training data. The shortest context window missed a lot of speech (higher MR), which is not suitable for further

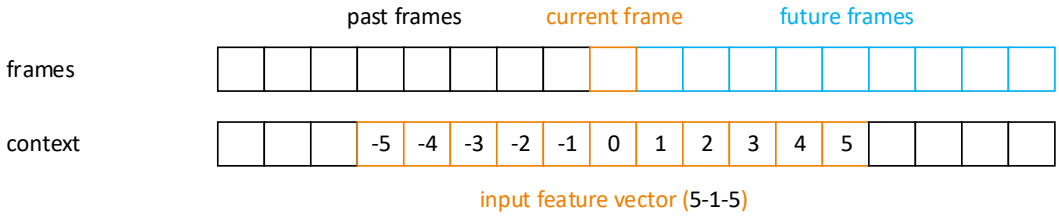

Figure 5.15: An example of a 0.1-second context window size (5-1-5).

speech processing. The context of 0.3 seconds performed quite similarly to the 0.5-second context window, although the accuracy of change points placements ($\delta_{2/3}$) degraded its performance.

Finally, it should also be noted that the computational time needed for decoding increased significantly with additional context. This time was 2 and 1.7 times lower for 0.1-second and 0.5-second contexts, respectively, than for the longest feature vector (1.6-second context). The context window size also influences online use as the decoder has to wait half of the context time to start decoding (i.e., it is waiting for future frames). This wait time is automatically added to latency (e.g., compare 0.05 to 0.8 seconds for the shortest and longest windows, respectively).

Table 5.6: Results showing the influence of the context window size on the performance of SAD.

context [s] (frames)	FER [%]	MR [%]	FAR [%]	F [%]	$\delta_{2/3}$ [s]
0.1 (5-1-5)	2.5	0.6	**7.2**	52.9	**0.22**
0.3 (15-1-15)	**2.4**	**0.4**	7.5	53.3	0.28
0.5 (25-1-25)	**2.4**	0.5	**7.2**	52.7	0.26
0.7 (35-1-35)	2.6	**0.4**	8.2	52.3	0.34
1.6 (80-1-80)	2.7	0.9	7.3	**55.6**	0.48

5.7.5 Number of Epochs

The number of training epochs has an impact not only on the achieved results but also on the amount of time needed to finish the training of the DNN. Too few epochs and the deep neural network might not converge at all, too many and the network might be prone to overfitting. The time needed for training of one epoch is dependent on the architecture of the DNN (and its hyper-parameters) as well as the amount of training data. This experimental evaluation focused on determining the ideal number of epochs for the already chosen hyper-parameters and data.

Figure 5.16 presents the results, specifically, the influence of the number of training epochs on three metrics: FER, MR, and FAR. It is clear that the MR improved with additional epochs while FAR, and more importantly, FER worsened (i.e., the improvements in MR were lesser than the deterioration in FAR). For this reason,

the deep neural network trained within 10 epochs performed as the best compromise yielding the best values in FER, FAR, and $\delta_{2/3}$ as well. It also saves a reasonable amount of training time. In this experiment, one epoch was trained on GPU for approximately 5 minutes (i.e., the final training time was 50 minutes instead of over 4 hours if 50 epochs required).

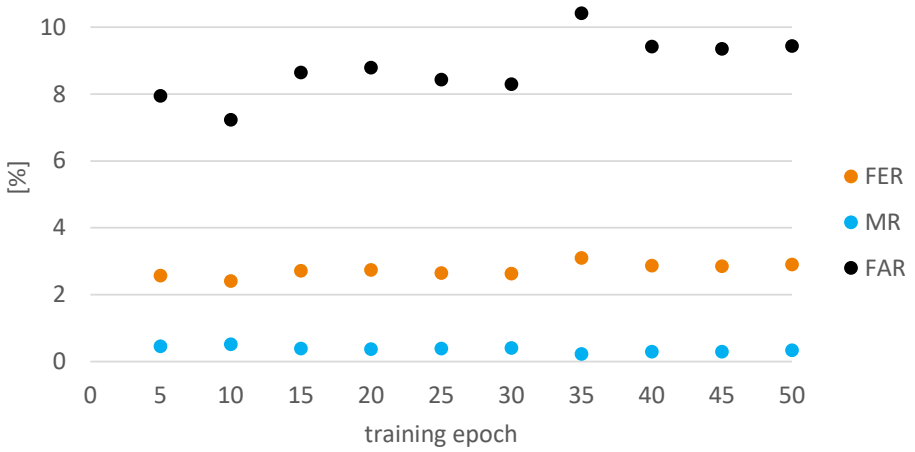

Figure 5.16: A graphical illustration of the influence of the number of training epochs on results of SAD.

5.7.6 Local Normalization

In all the previously concluded experiments, local mean normalization within 1-second long window was applied. This normalization is useful to smooth out sudden changes (e.g., speaker change points, hesitation, cough, etc.) in the input data, which can cause unexpected transitions between speech and non-speech events. However, local normalization increases computational demands during both training and evaluation phases, and it also adds latency as it is computed from 50 previous frames, the current frame, and 50 following frames (see Fig. 5.17 for an illustration). This experiment explored if the local mean normalization is indeed necessary for the proposed SAD approach. As usual, the rest of the DNN parameters remained unchanged.

The results are summarized in Table 5.7. They show that local mean normalization is an essential component of the proposed SAD approach. Without it, the results noticeably worsened in all observed metrics. For this reason, latency and computational demands could not be improved this way.

Table 5.7: Results of the experiment focusing on the use of local mean normalization.

normalization	FER [%]	MR [%]	FAR [%]	F [%]	$\delta_{2/3}$ [s]
no	3.4	0.8	10.0	35.6	0.29
yes	**2.4**	**0.5**	**7.2**	**52.7**	**0.26**

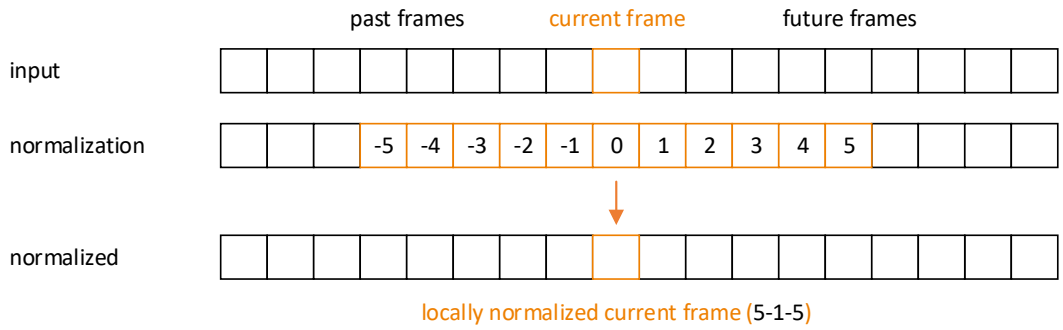

Figure 5.17: An example of local mean normalization within a 0.1-second long window (5-1-5).

5.8 Complex Architectures

Over the years, many different and more complex architectures have been proposed and utilized for a wide range of tasks. Architectures, such as convolutional neural networks, recurrent neural networks, or Time Delay Neural Networks (TDNN), all became a well-known term. Furthermore, new architectures are still being investigated every year. In the following experiments, the focus was on the application of CNNs and TDNNs to the task of speech activity detection and comparison of their results with feed-forward DNNs.

5.8.1 Convolutional Neural Networks

Convolutional neural networks known for their modeling capabilities were the first explored architecture. Within this experiment, three different CNNs (denoted as small, medium, and large) were trained and compared with a feed-forward DNN. All three CNNs were formed by two convolutional layers followed by three fully connected layers with 128 neurons per layer (1,024 for the large CNN). The inputs were composed of 51 feature maps (i.e., 0.5-second context), each 39×1 in size (i.e., the same FBCs as input features). The first convolutional layer consisted of 105 feature maps (16 for the small CNN) 39×1 in size. It was followed by a 3:1 max-pooling layer. The second layer was comprised of 157 feature maps (32 for the small CNN) 13×1 in size. The rest of the hyper-parameters were set as previously (see Sect. 5.3), and the CNNs were also trained using the torch framework.

The results are presented in Table 5.8. They show that all three CNNs achieved a lower miss rate than the standard feed-forward DNN. However, this improvement was overweighted by a general deterioration in all other observed metrics. Between the CNNs, the one denoted as big performed the best yielding F-measure of 55.6% and $\delta_{2/3}$ of 0.29 seconds. Lastly, in this case, CNNs were unable to extract more information from the training data to achieve significant improvements in performance.

Table 5.8: Results comparing CNNs and TDNN with a feed-forward DNN.

architecture	FER [%]	MR [%]	FAR [%]	F [%]	$\delta_{2/3}$ [s]
DNN	**2.4**	0.5	**7.2**	52.7	0.26
CNN (small)	2.8	**0.3**	9.0	48.7	0.32
CNN (medium)	2.6	0.4	8.3	49.7	0.31
CNN (large)	2.6	0.4	8.4	**55.6**	0.29
TDNN	2.8	0.7	7.9	50.7	**0.14**

5.8.2 Time Delay Neural Networks

The ability to model long-term dependencies between acoustic events can be crucial for many speech processing applications as it provides the target decoder with additional context information to base its decision on. Conservatively, recurrent neural networks are often used for this task. However, training RNNs can be a very time-consuming process. Another architecture that models long-term dependencies well and is trained in a similar amount of time as feed-forward DNNs are time delay neural networks [159].

The utilized TDNN was designed to match the feed-forward DNN. For this reason, it consisted of 5 hidden layers, each with 128 neurons. The input context of each layer required to compute output at one time step was five previous inputs, the current input, and five following inputs (from the previous layer). This setting matched the input context window size of the feed-forward DNN (i.e., 0.5 seconds). An example (of smaller input context of each layer) is illustrated in Fig. 5.18. As usual, the rest of the hyper-parameters remained unchanged, and FBCs were utilized as input features. Note that this network was trained using PyTorch[4].

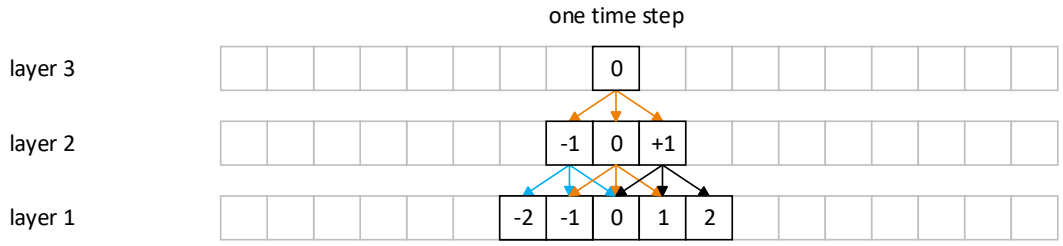

Figure 5.18: An example of a 1-1-1 input context in a 3-layer TDNN.

The results are shown in Table 5.8 (see the last row). They show an overall decrease in performance in all metrics except for $\delta_{2/3}$, which improved significantly. However, this improvement in the accuracy of change point placement was the only benefit of using the TDNN over the feed-forward DNN. Furthermore, the global deterioration in other metrics would result in a notable worse performance of, e.g., speech transcriber.

[4]https://pytorch.org/

5.9 Online Performance

An online performance of the proposed speech activity detection approach was closely monitored throughout the whole design and experimental evaluation. This performance is crucial for the approach to be integrated into the target TVR monitoring system without any issues. The proposed SAD approach averaged RTF of 0.01 and 2-second latency. Note that processor Intel Core i7-3770K @ 3.50GHz was used for all the computations. The achieved performance is well suited for seamless use in real-time processing applications, such as the target TVR monitoring system, without any major delay.

5.10 Evaluation on QUT-NOISE-TIMIT Corpus

So far, all of the experiments were conducted only using the development dataset, which was designed explicitly within this work. That means that the dataset is not suitable for a direct comparison of the proposed SAD approach (as described in Sect. 5.6) with different approaches already presented in the literature because it has not been used anywhere else or even released to the general public. However, this comparison is crucial to discover the full potential of the proposed SAD approach. For this reason, a standardized QUT-NOISE-TIMIT [44] corpus was utilized.

The evaluation on the QUT-NOISE-TIMIT corpus shows the performance of the proposed approach in comparison with five approaches already presented in [44] and two techniques reaching the state-of-the-art results [22, 129]. The five approaches are: standardized VAD system ITU-T G.729 Annex B [119], standardized advanced front-end ETSI [120, 121], Sohn's likelihood ratio test [123], Ramirez's long-term spectral divergence [124] and GMM-based approach with the use of MFCCs [44]. The latter two techniques are voice activity detection using subband noncircularity [129] and complete-linkage clustering for VAD [22]. Note that all these seven approaches were described in detail in their respective sections in Chap. 3.

5.10.1 QUT-NOISE-TIMIT Corpus

The main idea behind the creation of the QUT-NOISE-TIMIT [44] corpus was a lack of standardized datasets suitable for training and testing of SAD approaches in various target environments and under different SNR conditions. For this purpose, more than 10 hours of background noises across 10 different unique locations were gathered by the authors, and a corpus called QUT-NOISE was formed. These background noises covered five different but common scenarios (specifically cafe, car, home, reverb, and street). Each scenario was also recorded in two different source locations:

- cafe – outdoor cafe environment or indoor shopping food-court;
- car – windows down or up;

- home – kitchen or living room;
- reverb – indoor pool or partially enclosed carpark;
- street – inner-city or outer-city traffic-light controlled intersections.

The QUT-NOISE background noises were mixed with a clean speech from TIMIT corpus [153] creating 600 hours of new recordings with varying amount of speech segments, length (60 or 120 seconds) and SNR level (−10, −5, 0, 5, 10 or 15 dB). These new recordings then formed the standardized QUT-NOISE-TIMIT corpus. After that, the final corpus was evenly split into two groups (A and B) to provide training and testing subsets.

5.10.2 Evaluation Protocol

An evaluation protocol for the QUT-NOISE-TIMIT corpus was also provided in the given paper [44], and it was successfully followed by other works as well [22, 129]. It states the following points. During the training phase, no information about the target scenario is given to the system. The only available prior knowledge is the SNR level of the target environment: low noise (10, 15 dB), medium noise (0, 5 dB), or high noise (−10, −5 dB). For each target environment, group A is used for training and group B for testing and vice-versa. The distribution of the data is shown in Table 5.9. Finally, the decoded speech and non-speech segments are aligned with QUT-NOISE-TIMIT ground truth labels, and miss rate, false alarm rate, and half-total error rate are computed. See Fig. 5.19 for an example of the evaluation protocol in the low-noise target environment.

Table 5.9: An overview of the distribution of recordings in QUT-NOISE-TIMIT corpus.

target environment	group	recordings	hours	change points	speech
low noise		8,000	200	189,396	46.1%
medium noise	A & B	8,000	200	189,452	45.8%
high noise		8,000	200	190,064	46.1%
all	A & B	24,000	600	568,912	46.0%

For the following experiments, the proposed SAD approach followed this evaluation protocol, and it was trained as described in Sect. 5.6 except for not utilizing the artificial data (i.e., only the data from the QUT-NOISE-TIMIT corpus was used). The employed DNN and context-based WFST remained unchanged. Additionally, for each target SNR, the performance of the proposed SAD approach in different scenarios was explored as well.

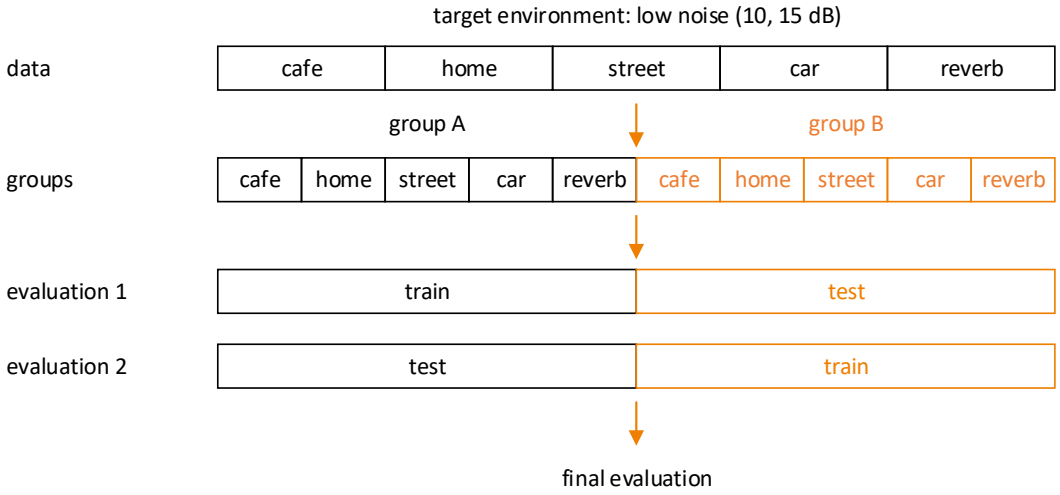

Figure 5.19: An example of the evaluation protocol for the QUT-NOISE-TIMIT corpus (low-noise target environment).

5.10.3 Low-Noise Conditions

For the experiment under low-noise conditions, recordings with SNR levels of 10 and 15 dB were utilized. The comparison of the proposed approach with other systems can be seen in the left part of Fig. 5.20. As the results show, the proposed SAD approach outperformed all other systems by a fair margin. The absolute reduction in the HTER was more than 2% over the formerly best complete-linkage clustering. The exact achieved value of the HTER was 2.6%. For the rest of the metrics, see the first row of Table 5.10. Note that the accuracy of the placement of change points was also impressive (F-value of 71.3% and $\delta_{2/3}$ of 0.05 seconds). In conclusion, the proposed SAD approach yielded state-of-the-art results under low-noise conditions.

Table 5.10: Summarized results of the proposed SAD approach on the QUT-NOISE-TIMIT corpus in all target environments. Overall results and results in each of the scenarios across all target environments are shown as well.

target e.	scenario	HTER [%]	FER [%]	MR [%]	FAR [%]	F [%]	$\delta_{2/3}$ [s]
low		2.6	2.7	2.0	3.2	71.3	0.05
medium	all	5.8	5.8	5.8	5.8	61.4	0.11
high		17.0	16.4	24.0	10.0	41.0	0.22
	cafe	12.8	12.7	14.2	11.5	63.6	0.20
	car	3.0	3.0	2.9	3.1	80.1	0.08
all	home	8.1	8.4	4.0	12.2	68.4	0.12
	reverb	13.5	12.6	24.7	2.3	65.6	0.17
	street	4.9	4.7	7.1	2.7	77.8	0.09
all	all	8.5	8.3	10.6	6.3	58.0	0.12

The right side of Fig. 5.20 shows the performance of the proposed approach in all scenarios. The most straightforward scenarios under low-noise conditions were car and street with only a small number of errors. On the other hand, the most problematic scenarios were home (highest HTER) and reverb (a lot of missed speech forcing additional errors in the potential follow-up transcription).

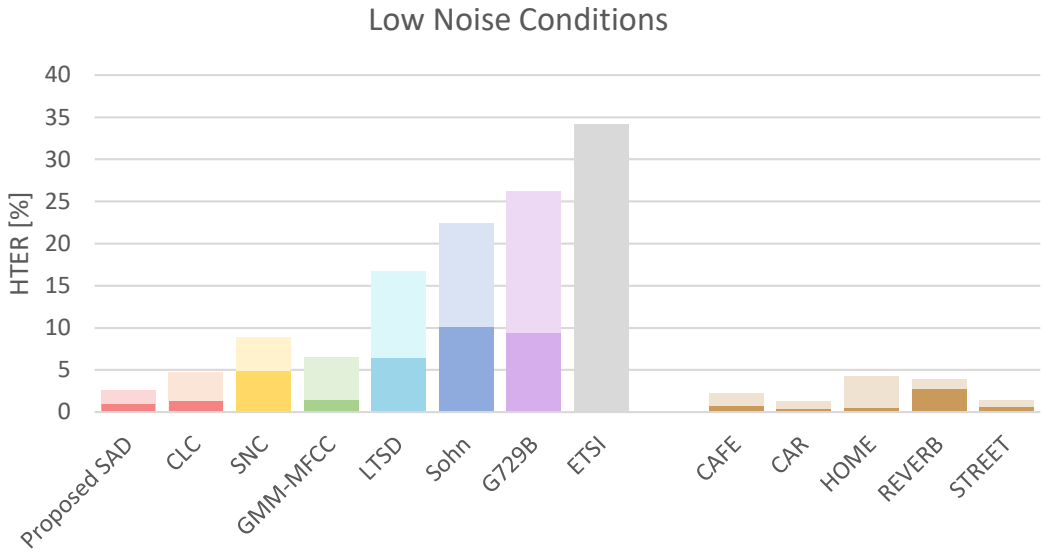

Figure 5.20: An evaluation of QUT-NOISE-TIMIT corpus in the low-noise target environment. In left: a comparison of results of the proposed approach with various SAD approaches. In right: a detailed performance of the proposed approach in all scenarios. The contribution of MR and FAR to HTER bars is displayed by darker and lighter shades, respectively.

5.10.4 Medium-Noise Conditions

Recordings with SNR levels of 0 and 5 dB were employed for the experiment under medium-noise conditions. The achieved results are shown in the left part of Fig. 5.21. Similarly to the experiment conducted under low-noise conditions, the proposed SAD approach yielded the best results, outperformed the other systems and thus reached the state-of-the-art results. Again, the absolute reduction in the HTER was over 2% (the exact achieved value was 5.8%, for other metrics see the second row of Table 5.10) over the second-best complete-linkage clustering approach. Lastly, the worsened conditions (medium noise) caused an increase of over 3% in HTER over the results obtained under low-noise conditions for the proposed SAD approach.

The right part of Fig. 5.21 compares the performance of the proposed approach in various scenarios. The car and street scenarios remained the least problematic (3% in HTER), while cafe and reverb were the most troubling scenarios. Also, note that the reverb scenario resulted in the most omitted speech causing many additional errors in potential speech processing applications.

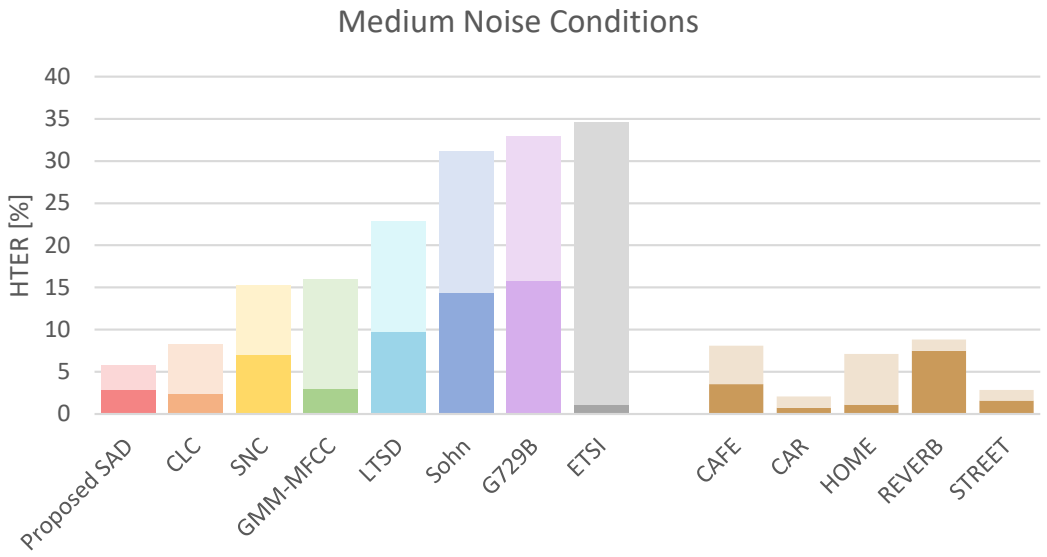

Figure 5.21: An evaluation of QUT-NOISE-TIMIT corpus in the medium-noise target environment. In left: a comparison of results of the proposed approach with various SAD approaches. In right: a detailed performance of the proposed approach in all scenarios. The contribution of MR and FAR to HTER bars is displayed by darker and lighter shades, respectively.

5.10.5 High-Noise Conditions

The most challenging target environment was based on recordings with SNR levels of −10 and −5 dB. The left part of Fig. 5.22 shows the comparison of the results of the proposed approach with various SAD approaches. Under high-noise conditions, the complete linkage clustering approach surpassed all other systems, including the proposed SAD approach (by approximately 2% in the HTER). However, the proposed approach still notably outperformed all other systems (by at least 10% in HTER). Specifically, the yielded HTER was 17% (other metrics are summarized in the third row of Table 5.10). Sadly, this was an increase of over 11% over the results achieved under medium-noise conditions. Furthermore, most of the errors produced by the decoder resulted in omitted speech (i.e., higher miss rate) culminating in even more errors in further speech processing. However, the proposed SAD approach was not designed and fine-tuned for such high-noise conditions, and better results could be potentially achieved with a modified DNN and WFST configuration.

The performance of the proposed approach in all scenarios is depicted in the right part of Fig. 5.22. The results show that the car scenario was the easiest one to correctly detect speech (HTER slightly over 5.5%) while cafe and reverb scenarios remained the toughest ones (HTER close to 28%). The reverb scenario was even more problematic as the majority of errors resulted in omitted speech.

Finally, Table 5.10 also presents the results of the proposed SAD approach in all scenarios of QUT-NOISE-TIMIT corpus across all target environments as well as overall results. The results confirm that scenarios reverb and cafe were the most complicated while scenarios car and street were the most straightforward ones.

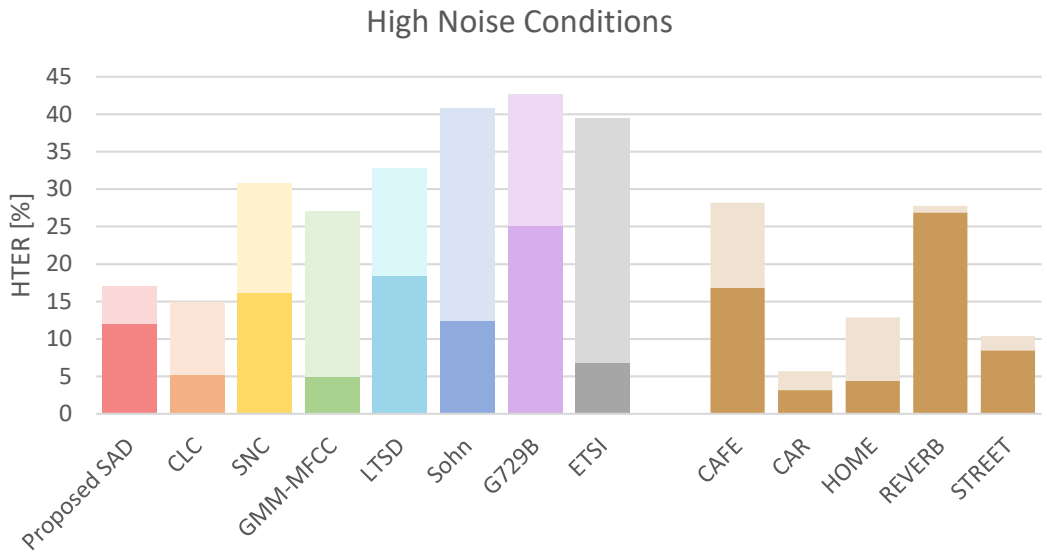

Figure 5.22: An evaluation of QUT-NOISE-TIMIT corpus in the high-noise target environment. In left: a comparison of results of the proposed approach with various SAD approaches. In right: a detailed performance of the proposed approach in all scenarios. The contribution of MR and FAR to HTER bars is displayed by darker and lighter shades, respectively.

5.10.6 Online Performance

In addition to the standard HTER evaluation, the real-time performance of the proposed SAD approach on QUT-NOISE-TIMIT corpus was also monitored. The approach averaged a real-time factor of 0.02, with latency being 1.8 seconds. The detailed results in all target environments, as well as averages in all scenarios, are presented in Table 5.11. The first three rows prove that the decoder is capable of making the final decisions much faster if the conditions are mild (i.e., 0.4-second difference in latency between the low- and high-noise conditions). In other words, latency is worse if the overall performance of the proposed approach is worse as well (e.g., both HTER and latency were the weakest under high-noise conditions as in reverb and cafe scenarios).

5.11 Evaluation in Real Speech Transcription System

Given the findings and results from all previous experiments, the final proposed speech activity detection approach (i.e., with the context-based smoothing model as introduced in Sect. 5.6) was integrated into the TVR monitoring system developed at the author's lab in cooperation with the NanoTrix company and thus evaluated in a real speech transcription system.

Table 5.11: Results summarizing the real-time performance of the proposed approach on the QUT-NOISE-TIMIT corpus.

target environment	scenario	RTF	latency [s]
low noise			1.6
medium noise	all	0.02	1.8
high noise			2.0
	cafe		1.9
	car		1.7
all	home	0.02	1.8
	reverb		2.0
	street		1.7
all	all	0.02	1.8

5.11.1 Experimental Setup

Four commonly utilized metrics were applied to evaluate the performance of speech transcription. The first three, word error rate, Word Accuracy (WAcc) and Percent Correct (PC), focus on the quality of transcriptions and are defined as follows:

$$WER[\%] = \frac{I + S + D}{N} * 100 \ , \tag{5.9}$$

$$WAcc[\%] = 100 - WER \ , \tag{5.10}$$

$$PC[\%] = 100 - \frac{S + D}{N} * 100 \ , \tag{5.11}$$

where I is the number of insertions (words the recognizer added to its output), D stands for deletions (not transcribed words), S marks substitutions (words that were mistaken), and N is the total number of words in the reference text (see Fig. 5.23 for an illustration). Note that, word accuracy and word error rate are complementary metrics. The only difference between percent correct and the other two metrics is that inserted words are not considered as errors in PC. The final metric was the real-time factor (see Sect. 5.1.3) utilized to evaluate the real-time performance of speech transcription. In summary, the integration of the proposed SAD approach into the target speech transcription system should ideally lead to significant improvements in RTF while keeping the quality metrics at least around the same level.

For evaluation, two datasets of Czech broadcasts have been utilized (see Table 5.12 for an overview). The first dataset represents 4 hours (22,204 words) recorded from a Czech live news TV channel. Approximately 60% of its content consists of speech segments. The length of the other dataset is 8 hours, it contains 7,212 words, and speech frames form only 10% of its content. This dataset represents a broadcast of a Czech local radio station.

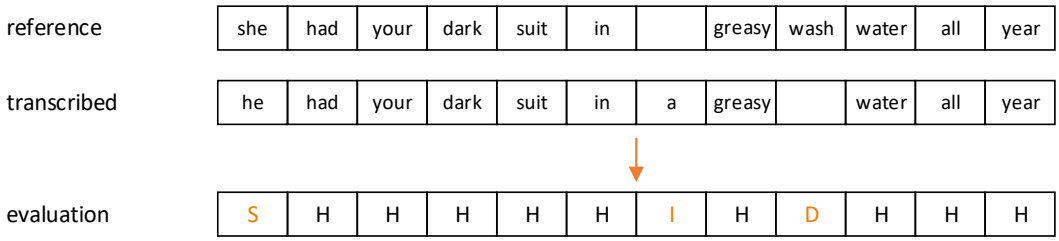

Figure 5.23: An example of alignment between reference and transcription in speech transcription evaluation.

Table 5.12: An overview of utilized evaluation datasets for speech transcription.

dataset	hours	speech	words
live news TV channel	4	60%	22,204
local radio station	8	10%	7,212

The transcription system developed at the author's lab in cooperation with the NanoTrix company employed an acoustic model based on a Hidden Markov Model – Deep Neural Network (HMM-DNN) hybrid architecture [110], where the baseline Gaussian mixture model was trained as context-dependent, speaker-independent and contained 3,886 physical states. The data for training of this model contained 270 hours of clean speech recordings of Czech. The hyper-parameters used for the DNN training were set as follows:

- 5 hidden layers;
- decreasing number of neurons per layer (1,024-1,024-768-768-512);
- ReLU activation function;
- mini-batches size of 1,024;
- 0.08 learning rate;
- 35 epochs.

The input features were:

- 39-dimensional log filter bank coefficients;
- computed using 25 ms frames of the signal with frame shifts of 10 ms;
- concatenation of 5 previous frames, the current frame, and 5 following frames;
- local normalized within a one-second window.

Note that fine-tuning of these hyper-parameters was discussed and published in [160] and later on, further extended.

The linguistic part of the system was composed of a lexicon and a language model. The lexicon contained 550,000 entries with multiple pronunciation variants, and the language model was based on N-grams. For practical reasons (mainly with respect to the immense vocabulary size), the system used bigrams. However, 20% of all word-pairs actually included sequences containing three or more words, as the lexicon contains 4,000 multi-word collocations. The unseen bigrams were backed-off by Kneser-Ney smoothing [161].

5.11.2 Experimental Evaluation

Within the performed experiments, both evaluation datasets were transcribed a) with and b) without the use of the proposed speech activity detection approach. The obtained results are presented in Table 5.13. They reveal that the utilization of the proposed approach was advantageous on both evaluation datasets. The yielded PC and WER (WAcc) show that SAD limited the insertions coming from the non-speech parts and omitted hardly any speech parts. The proposed SAD approach allowed the transcription system to operate with improved accuracy and, at the same time, RTF was almost two times, and more than ten times lower for the first and second evaluation datasets, respectively. Of course, the reason for this difference is that the data in the second dataset contains fewer speech segments. All presented RTF values were measured using processor Intel Core i7-3770K @ 3.50GHz. Just a reminder, the RTF of the proposed SAD approach was 0.01 and could be considered negligible. The latency was around 2 seconds (details in Sect. 5.9). In conclusion, the transcription system complemented with the proposed speech activity detection approach can be utilized for online speech transcription without any major delay.

Table 5.13: An evaluation of the proposed SAD approach in a real speech transcription system.

dataset	SAD	WER [%]	WAcc [%]	PC [%]	RTF
live news TV channel	yes	**12.4**	**87.6**	**89.7**	**0.42**
	no	12.7	87.3	**89.7**	0.77
local radio station	yes	**14.0**	**86.0**	**88.5**	**0.08**
	no	17.9	82.1	88.4	0.83

6 Proposed Speaker Change Point Detection Approach

Inspired by the proposed speech activity detection approach, the final speaker change point detection approach was proposed in several successive experiments heavily detailed within this chapter. This development was published in [4], and portions of the paper were reused in this thesis. Ultimately, this chapter describes the evaluation metrics, training, development and evaluation data, experimental evaluation of all steps taken, evaluation on all languages of the COST278 database, and finally, it sets the final SCP detection approach.

6.1 Evaluation Metrics

The evaluation metrics for speaker change point detection were close to identical to the ones used for SAD due to the similarity of both tasks. The overall accuracy metrics are the only exception because framewise evaluation is not particularly valuable for change point detection (i.e., the main concern is not the content of the detected segments but the actual placement of transitions between them). Therefore, the metrics for SCP detection can be divided into only two subsets: change point quality metrics and performance metrics. In total, 6 already presented metrics were observed.

For the former subset, four metrics, specifically precision, recall, F-measure, and $\delta_{2/3}$, were employed. Precision and recall were additionally reported to provide more information about the errors the decoder makes (i.e., falsely detected change points result in worsened precision while undetected change points yield worse recall). More information and formalism of these metrics can be found in Sect. 5.1.2.

The latter group consists of two previously introduced metrics: latency and real-time factor. The same processor, Intel Core i7-3770K @ 3.50GHz, was used to do the computations. The detailed description of these metrics is available in Sect. 5.1.3.

6.2 Data Used

For training, 20,000 recordings, each with an average length of 5 seconds, have been prepared with the help of automatic Czech TV/radio broadcast data transcriptions. Each of these recordings contains exactly one speaker change point (i.e., the set consists of 20,000 speaker transitions). These transitions can be divided into four distinct groups (female to female, female to male, male to female, and male to male). Each of them is represented by 5,000 change points. Note that each recording was extracted from a whole utterance, and there are no artificial cuts or changes in channels.

The annotations of this data (for SCP detection) were generated in a fully automated way. The frame corresponding to the actual change point, as well as the safety collar frames around it, were labeled as change points. This safety collar was set to 1 second (100 frames), i.e., 50 frames before and 50 frames after the actual change point were considered as speaker transition frames. That is due to the fact that a) determining the precise change point is quite often an ambiguous task (silence, crosstalk, etc.), and b) it is necessary to provide DNN training with enough information about the speaker transitions. The remaining frames were labeled as no change point. An example of annotation of one recording is shown in Fig. 6.1.

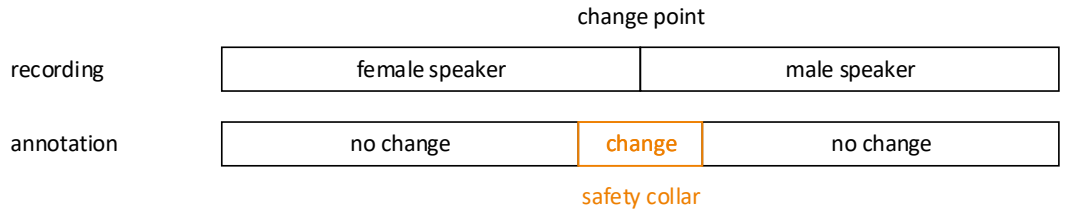

Figure 6.1: An example of an annotation of training data for SCP detection.

For development purposes, the Czech train subset of standardized COST278 [95, 96] pan-European broadcast news database has been utilized. Accurate annotations are provided by the database. For evaluation, the Czech test subset of COST278 has been employed. It consists of four recordings of different Czech broadcasts (ČT1, Nova and Prima) in a total length of 90 minutes. It contains not only clean speech segments but also segments with background noise and jingles. In total, 379 speaker change points are labeled within the data. Finally, an overview of the utilized data is available in Table 6.1.

Table 6.1: An overview of utilized data for SCP detection.

dataset	recordings	hours	change points
training	20,000	30	20,000
artificial cuts	60	10	14,340
speaker-homogeneous	7,000	10	0
enhanced training	27,060	50	34,340
development	9	3	827
evaluation	4	1.5	379

6.3 Reference Results

To obtain reference results with an offline system, publicly available LIUM Speaker Diarization toolkit [101, 102] was used. The SCP detection portion of the system is

covered by BIC segmentation and BIC clustering, followed by segmentation based on Viterbi decoding and boundary adjustments (more information can be found in Sect. 4.1). The system is also supplemented with a pre-trained model fine-tuned for TV and radio broadcasts (i.e., the same target task as this thesis). During the evaluation, the LIUM toolkit was operated with an RTF of 0.016, achieving reference results in F-measure of 84.6% and $\delta_{2/3}$ of 0.13 seconds (see the first row in Table 6.2 for more detailed results).

Table 6.2: Summarized results of the proposed SCP detection approach described in Chap. 6.

approach	P [%]	R [%]	F [%]	$\delta_{2/3}[s]$	RTF	L [s]
LIUM toolkit	**89.9**	80.0	84.6	**0.13**	**0.016**	-
DNN + WFST decoder	59.4	63.6	61.4	0.24	0.022	2.4
+ enhanced data	67.0	70.7	68.8	0.21	0.022	2.3
+ Δ MFCC	72.8	74.7	73.7	0.19	0.024	**1.9**
+ CNN	79.3	77.8	78.6	0.17	0.054	**1.9**
+ 2.5-second context window	80.3	81.8	81.1	0.17	0.054	2.3
+ 1-second long transition model	82.7	81.8	82.2	0.17	0.065	2.9
+ tuned for offline use	86.7	**84.4**	**85.6**	0.18	0.079	4.8

6.4 Initial Approach Based on DNN and WFST

The initial SCP detection approach was inspired by the proposed SAD approach designated for online use (as described in detail in Chap. 5). This SCP detection approach was based on DNN trained as a binary classifier (change point or no change point) and WFST designed as an online decoder detecting speaker transitions given the output from the DNN.

The binary DNN was trained using the following hyper-parameters:

- 2 hidden layers;
- 64 neurons per hidden layer;
- ReLU activation function;
- mini-batches size of 1,024;
- 0.08 learning rate;
- 15 epochs.

The utilized input features were:

- 39-dimensional MFCCs;

- computed using 25 ms frames of the signal with frame shifts of 10 ms;
- concatenation of 100 previous frames, the current frame, and 100 following frames (i.e., a 2-second context window);
- not local normalized.

An illustration of the employed feed-forward deep neural network is depicted in Fig. 6.2. Note that all the DNNs (i.e., for all SCP detection experiments) were trained on GPU using the PyTorch framework.

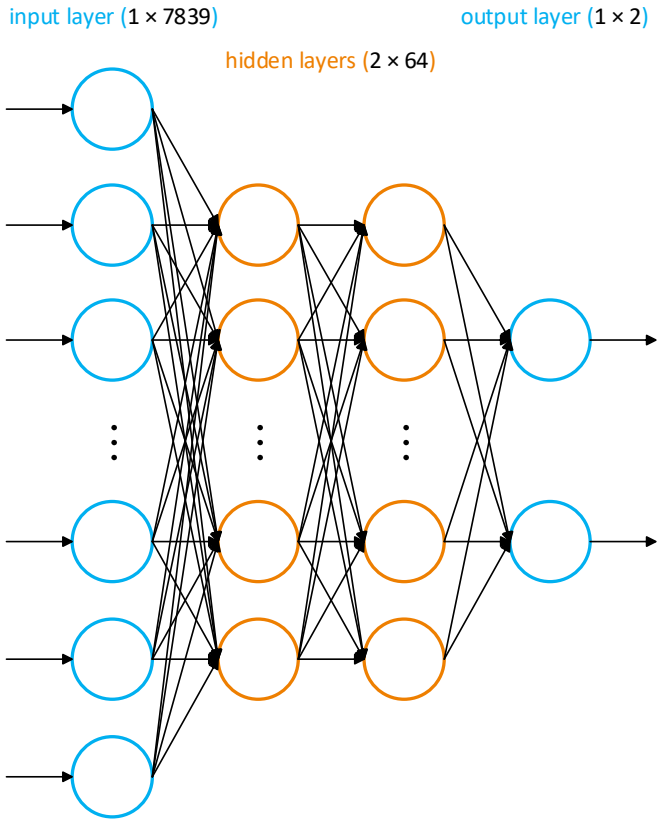

Figure 6.2: A feed-forward DNN used in the SCP detection.

As stated above, WFSTs were utilized (using the OpenFst library) as an online decoder. The decoding scheme consists of two transducers. The first one models the input signal (see Fig. 6.3), while the second one is the transduction model and represents the change point detection (see Fig. 6.4). It consists of two states, 0 and 1. The transitions between states 0/1 emit labels the start/end change points. The resulting change point is placed in the middle between these two labels. The transitions are also penalized by factors P1 and P2, whose values were fine-tuned on the development set. The decoding process was done in the same way as for SAD, as described in detail in Sect. 5.4.

The results are presented in the second row of Table 6.2. They show that the decoder was capable of operating in real time with an RTF value of 0.022. This

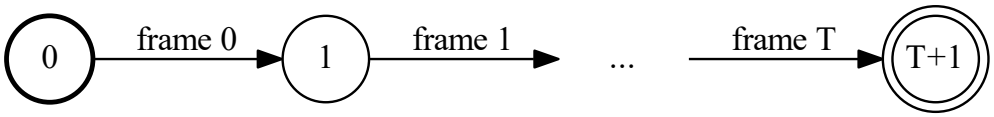

Figure 6.3: A transducer modeling the input signal for SCP detection.

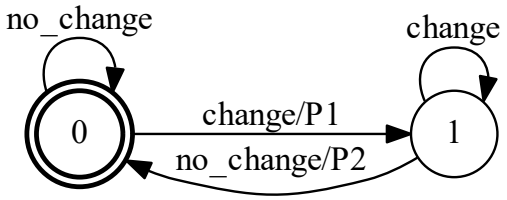

Figure 6.4: A transducer representing the transduction model for SCP detection.

value, combined with the latency of 2.4 seconds, allowed it to be seamlessly used in an online environment. Although the achieved results provided a decent starting point, the precision was particularly weak and overshadowed by LIUM toolkit (i.e., 59.4% vs. 89.9%). Therefore, the next goal was to improve the quality of the SCP detection.

6.5 Enhanced Training Dataset

After thoroughly evaluating the results obtained so far, two types of errors were the most prominent. The first one was represented by change points omitted due to the quick artificial transitions between speakers (e.g., director cuts in broadcast news) while the second type resulted in change points falsely detected because of a silence longer than 0.5 seconds in speaker-homogeneous segments (caused by deep breaths or hesitation). As a solution to the first issue, 10 hours of recordings were prepared by artificially joining utterances of two different speakers. In total, 14,340 change points with a uniform distribution between all transition types (female-female, female-male, male-female, and male-male) were thus added to the DNN training dataset. An example of such recording with annotation is shown in Fig. 6.5. To reduce the latter type of errors, another 10 hours of additional training data were prepared. This data focuses on speaker-homogeneous segments with frequent occurrences of long silences (see Fig. 6.6 for an example of recording and its annotation). More information about the enhanced training dataset is presented in Table 6.1.

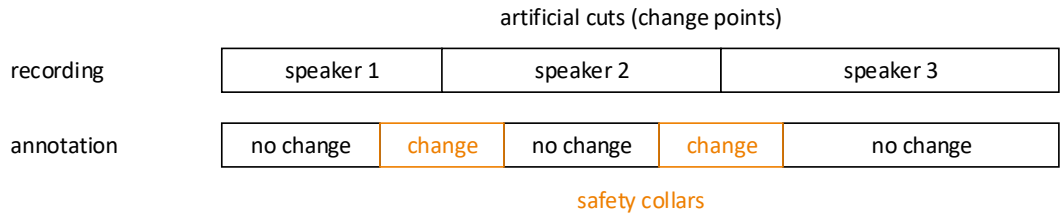

Figure 6.5: An example of additional data, rich in artificial cuts (with annotation).

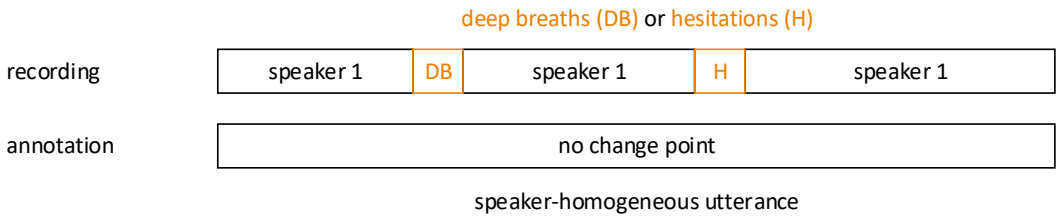

Figure 6.6: An example of additional data: speaker-homogeneous recording with deep breaths and hesitations. Annotation is shown in the second row.

The results gathered in the third row of Table 6.2 show that the use of enhanced training dataset led to significant improvement in all of the evaluation metrics observed. For example, the F-measure value got boosted up from 61.4% to 68.8%, while $\delta_{2/3}$ was enhanced to 0.21 seconds. Additionally, the average latency was slightly reduced, namely, from 2.4 seconds to 2.3 seconds.

6.6 Acoustic Features

In the next experiments, several feature extraction techniques were explored. In addition to the 39-dimensional MFCCs, 13-dimensional MFCCs with Δ and $\Delta\Delta$ coefficients (i.e., a 39-dimensional feature vector as well), and 39-dimensional bottleneck features were also utilized. As suggested, e.g., in [162–164], BTN features were extracted from DNN trained to discriminate physical states (senones) of a Czech tied-state triphone acoustic model. This deep extractor was trained on 270 hours of clean speech recordings of the Czech language. The hyper-parameters were set as follows:

- 5 hidden layers (the third one being the bottleneck layer);
- 1024 neurons per hidden layer (39 for the bottleneck layer);
- ReLU activation function (sigmoid for the bottleneck layer);
- mini-batches size of 1,024;
- 0.08 learning rate;
- 50 epochs.

The input features were:

- 39-dimensional log filter bank coefficients;
- computed using 25 ms frames of the signal with frame shifts of 10 ms;
- concatenation of 5 previous frames, the current frame, and 5 following frames;
- local normalized within a one-second window.

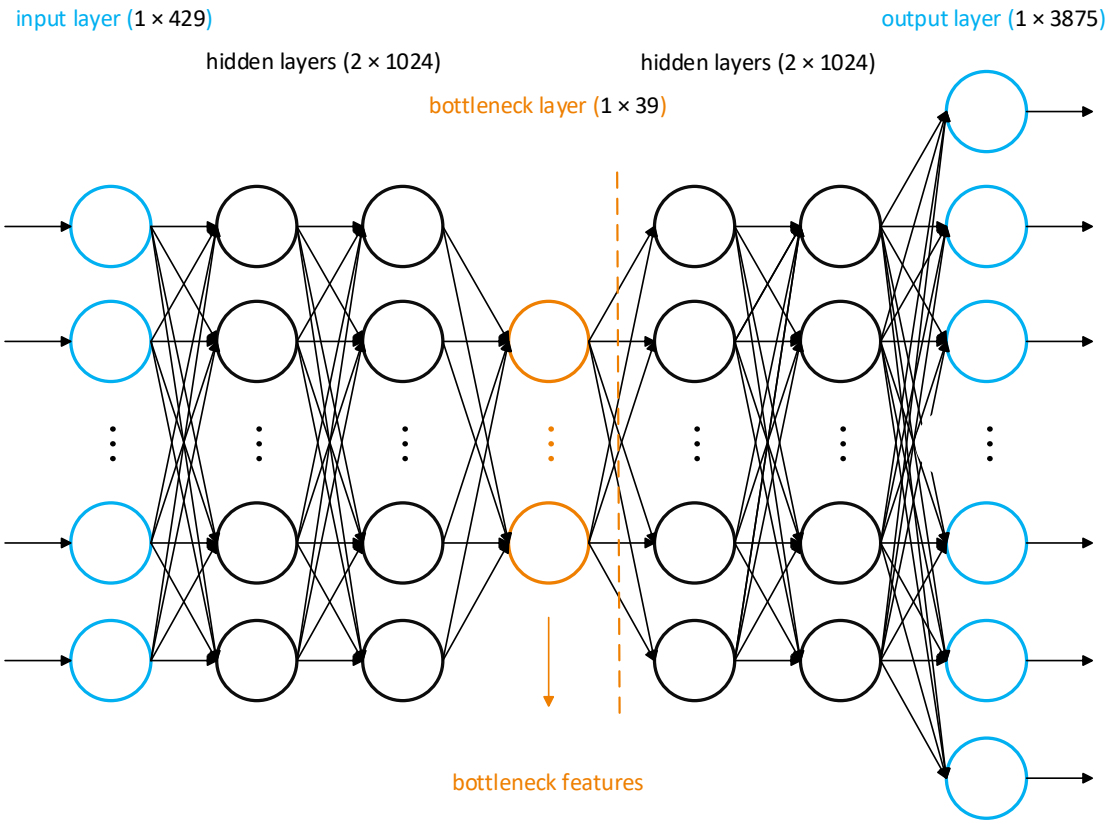

Figure 6.7: An overview of the deep bottleneck feature extractor.

An illustration of the BTN feature extractor can be seen in Fig. 6.7. Furthermore, more detailed information about the extractor and its performance in spoken language identification can be found in [165].

The results obtained are shown in Table 6.3. They show that the BTN features yielded significantly worse results in all of the observed metrics (e.g., the F-measure value dropped from 68.8% to 56.7%) and that they are more suitable for the tasks of language and speaker identification. On the contrary, the MFCCs with the Δ and $\Delta\Delta$ coefficients outperformed the originally chosen MFCC configuration. Both the quality and real-time performance of SCP detection improved (e.g., the latency was reduced from 2.3 seconds to 1.9 seconds because the decoder was able to make the final decisions more rapidly). A likely reason is additional information provided by the Δ and $\Delta\Delta$ coefficients.

Table 6.3: Results of the experiment exploring various feature extraction techniques.

features	P	R	F [%]	$\delta_{2/3}[s]$	RTF	L $[s]$
MFCCs	67.0	70.7	68.8	0.21	**0.022**	2.3
MFCCs + Δ + $\Delta\Delta$	**72.8**	**74.7**	**73.7**	**0.19**	0.024	**1.9**
BTNs	53.7	60.1	56.7	0.26	0.070	2.9

6.7 Convolutional Neural Networks

In the next step, more complex neural network architecture – CNN – was investigated. This architecture was employed for its feature representation and modeling capabilities. The utilized CNN was composed of two convolutional and two fully connected layers. The inputs consisted of 201 feature maps (i.e., 2-second context windows as before) in size of 39×1. The first convolutional layer was comprised of 105 feature maps at a size of 39×1, followed by a 3:1 max-pooling layer; the second one had 157 feature maps at a size of 13×1. The rest of the hyper-parameters was set as stated in Sect. 6.4.

The results are summarized in the fifth row of Table 6.2. The utilization of the CNNs yielded an overall improvement in all quality metrics (e.g., the F-measure value increased from 73.7% to 78.6%). The latency remained constant while the deterioration in RTF could be considered negligible (i.e., it is still significantly smaller than 1). For these reasons, CNNs were thus utilized for all follow-up experiments.

6.8 Context Window Size

The next experiments focused on the size of the input feature window. This additional context should result in a higher quality of the SCP detection at the cost of worse latency. Initially, a 2-second window had been chosen (with 100 preceding frames, a current frame, and 100 following frames). In this experimental evaluation, the sizes ranging from 1 second up to 4 seconds were explored.

The results are in Table 6.4. As expected, the performance (i.e., F-measure and $\delta_{2/3}$) was further improved with the additional context (e.g., up to F-measure of 81.7%). On the contrary, the latency of the system was worsened with more context information by up to 2 seconds. The RTF remained relatively constant (with just a slight deterioration with more information) throughout the evaluation.

Table 6.4: Results exploring the influence of the context window size on SCP detection.

context [s] (frames)	P	R	F [%]	$\delta_{2/3}[s]$	RTF	L [s]
1 (50-1-50)	71.3	69.4	70.3	0.21	**0.053**	**1.4**
1.5 (75-1-75)	71.0	72.8	71.9	**0.14**	**0.053**	1.7
2 (100-1-100)	79.3	77.8	78.6	0.17	0.054	1.9
2.5 (125-1-125)	80.3	81.8	81.1	0.17	0.054	2.3
3 (150-1-150)	80.0	**83.1**	81.5	0.17	0.054	2.6
3.5 (175-1-175)	**80.5**	82.6	81.5	0.16	0.055	3.1
4 (200-1-200)	80.4	**83.1**	**81.7**	0.16	0.055	3.5

6.9 WFST with a Forced Length of Transition

In the follow-up series of experiments, the aim was to further improve the results achieved so far by introducing WFST with a forced transition model. This model was designed to reflect the annotation style of the training data. As stated in Sect. 6.2, a 1-second (100 frames) window around the actual change point was labeled as speaker transition frames. However, during the decoding, the real duration of the transition between two speakers substantially varied.

Therefore, in this experiment, the duration of the transition was forced to be exactly 1 second at first. For this purpose, the transduction model was modified (see in Fig. 6.8) to correspond to the duration of the forced transition: it consists of two main states (0 and 1) and 98 transition states (shown as …).

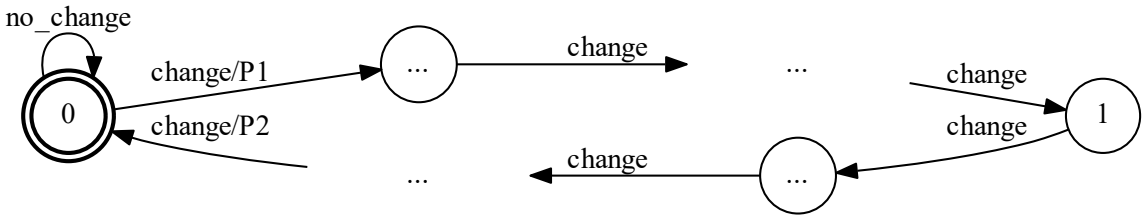

Figure 6.8: A transducer representing the transduction model with the forced transition for SCP detection.

This scheme works as follows: when a speaker change occurs, the decoder moves frame by frame from state 0 through half of the transition states to state 1. Here, a new change point label is provided, and the decoder moves backward to state 0, where it waits until the next change occurs. Note that, during this process, the penalty factors P1 and P2 (tuned on the development set) are in place as well.

The results are summarized in Table 6.5. First, a CNN with a context size of 2.5 seconds was used. Next, not only the forced length of the transition at 1 second but also several other values in a range from 0.5 up to 2 seconds were evaluated. The results show two contradictory trends: the quality of detection increased with the additional duration, while the RTF and latency values were worsened. Therefore, the optimal value of the duration strongly depends on the target application.

Table 6.5: Results of the experiment studying varied durations of forced transitions in the WFST.

forced duration [s]	P	R	F [%]	$\delta_{2/3}[s]$	RTF	L [s]
0.5	77.2	75.2	76.2	**0.13**	**0.057**	**2.2**
1	82.7	**81.8**	82.2	0.17	0.065	2.9
1.5	83.5	81.5	82.5	0.16	0.072	3.7
2	**84.2**	81.5	**82.8**	0.17	0.079	4.5

6.9.1 Online Application

For online application, the primary limiting factor is latency. In this environment, with the forced length of 1 second and total latency below 3 seconds, the proposed approach still allows for performing speaker change point detection with an accuracy level approaching the offline reference system (see the penultimate row of Table 6.2). As such, the proposed SCP approach is ready to be integrated into the TVR monitoring system.

6.9.2 Offline Application

For offline application, where the latency and real-time processing are not an issue, it is possible to tune the proposed SCP detection approach to improve the achieved results even further. For instance, a system based on CNN, the context window size of 3 seconds, and WFST with a forced length of 2 seconds yielded an F-measure value of 85.6% and a $\delta_{2/3}$ value of 0.18 seconds (with the latency at 4.8 seconds). These results are available for comparison in the last row of Table 6.2.

6.10 Local Normalization

So far, all of the presented experiments were conducted without the use of local mean normalization. However, this normalization technique was undoubtedly beneficial to the performance of the proposed speech activity detection approach (see Sect. 5.7.6 for more details) at the cost of slightly worsened latency (0.5 seconds). Naturally, in the following experiment, the effects of the local mean normalization on SCP detection were assessed.

The results of this experimental evaluation are summarized in Table 6.6. They show that local mean normalization is not a viable technique for SCP detection. Except for $\delta_{2/3}$, all of the other observed metrics notably worsened (e.g., F-measure dropped from 82.8% to 59.4% and latency increased by 0.6 seconds). The likely reason for this is that the transitions between two speakers (i.e., similar events) get blurred, making them considerably harder for the decoder to detect. This works much better for speech activity detection where the differences between speech and non-speech events are more significant (i.e., they do not get blurred enough) and at the same time, the transitions inside the speech (e.g., two speakers) or non-speech events (e.g., two songs) get blurred enough to not result in false change points.

Table 6.6: Results of the experiment focusing on the use of local mean normalization for SCP detection.

forced duration [s]	P	R	F [%]	$\delta_{2/3}[s]$	RTF	L $[s]$
no	**82.7**	**81.8**	**82.2**	0.17	**0.065**	**2.9**
yes	58.2	60.7	59.4	**0.12**	0.066	3.5

6.11 Evaluation on Whole COST278 Database

Until now, all of the conducted experiments were evaluated only on the Czech test subset of the COST278 database. Within this experiment, both the proposed approach and the reference system were employed for SCP detection on the whole test dataset, and their performance on a varied set of languages available in the COST278 database was monitored.

Furthermore, all the previous experiments were trained on data specifically selected and prepared in this work for the task of SCP detection. For a final experiment, the whole train subset of the COST278 database (previously used as development data) was utilized as intended (i.e., training data) to explore the influence of diverse training data on the final performance of the proposed approach.

6.11.1 COST278 Database

The COST278 pan-European Broadcast News Database [95, 96] was created as a joint work of several European institutions to support research focusing on design and evaluation of speaker segmentation and clustering algorithms for broadcast news. As such, it provides approximately 40 hours of recordings (of broadcast news) divided into 11 distinct datasets according to the target language. The available languages are Basque, Belgian Dutch, Czech, Spanish, Galician, Greek, Croatian, Hungarian, Portuguese, Slovenian, and Slovak. Additionally, each dataset was split into two disjoint subsets, one for training and one for testing. Finally, for each recording, an annotation providing information necessary for tasks such as speech activity detection, speaker change point detection, gender detection, or speaker verification and identification was prepared. Speech transcriptions were also made available.

6.11.2 Experimental Setup

As already stated before, for the following evaluation, the proposed SCP detection approach was trained only on the training dataset of the COST278 database. Two configurations of the proposed approach were explored – one designed for online use (as presented in Sect. 6.9.1) and one designated for offline applications (described in Sect. 6.9.2). In summary, both configurations utilized MFCCs with the Δ and $\Delta\Delta$ coefficients, the CNN instead of the feed-forward DNN, an extended context size (2.5/3 seconds for online/offline use), and the WFST-based decoder with a forced transition (1/2 seconds for online/offline use). The evaluation was done on all 11 languages of the COST278 test subset, and the results were compared with the LIUM toolkit. The goal was to see if the proposed single-pass approach (without clustering) can compete with an offline reference tool. Plus, the final experiment explored different training data – the COST278 training subset and a mixture of COST278 data and the data specifically selected and prepared for this work (see Sect. 6.2 and 6.5). The online configuration was employed for this experiment.

6.11.3 Online Comparison

The first explored configuration was the one designed for online use. The results of the comparison with the reference system are presented in the first two rows of Table 6.7. They show that both approaches performed on a relatively similar level. LIUM toolkit yielded an F-measure value of 73.5% and a $\delta_{2/3}$ value of 0.21 seconds, while the proposed approach scored an F-measure value of 73.1% and a $\delta_{2/3}$ value of 0.15 seconds, with the latency at 2.9 seconds.

Table 6.7: Summarized results comparing the proposed SCP detection approach with the reference system on the whole COST278 database.

approach	P [%]	R [%]	F [%]	$\delta_{2/3}[s]$	RTF	L [s]
LIUM toolkit	66.1	**82.7**	73.5	0.21	**0.016**	-
proposed approach – online	73.2	72.9	73.1	**0.15**	0.064	**2.9**
proposed approach – offline	**75.8**	75.4	**75.6**	0.19	0.079	4.7

Figure 6.9 depicts the detailed results for all COST278 languages. The easiest ones were four closely related Slavic languages – Czech, Slovenian, Croatian and Slovak. Basque and Spanish for the LIUM toolkit and Belgian Dutch and Basque for the proposed SCP detection approach were the most difficult instances.

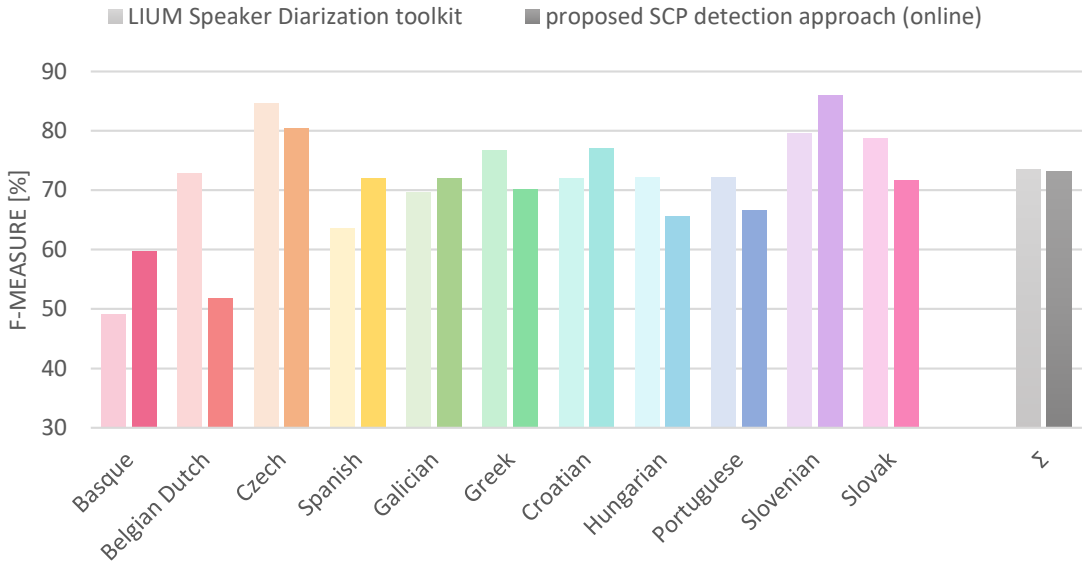

Figure 6.9: A comparison of the proposed SCP detection approach (tuned for online use) with the reference system on the whole COST278 database. Lighter columns mark the reference system while the darker ones indicate the proposed approach.

6.11.4 Offline Comparison

The second explored configuration was the one designated for offline use. The third row of Table 6.7 presents the achieved results in detail. Specifically, these slightly improved results show that the proposed offline approach yielded an F-measure value of 75.6% and a $\delta_{2/3}$ value of 0.19 seconds with the latency at 4.7 seconds.

The detailed comparison of the evaluation on all COST278 languages with LIUM toolkit is shown in Fig. 6.10. For the proposed offline approach, the easiest languages were Slovenian, Czech, and Slovak, three closely related Slavic languages. Belgian Dutch and Basque remained to be the most challenging datasets.

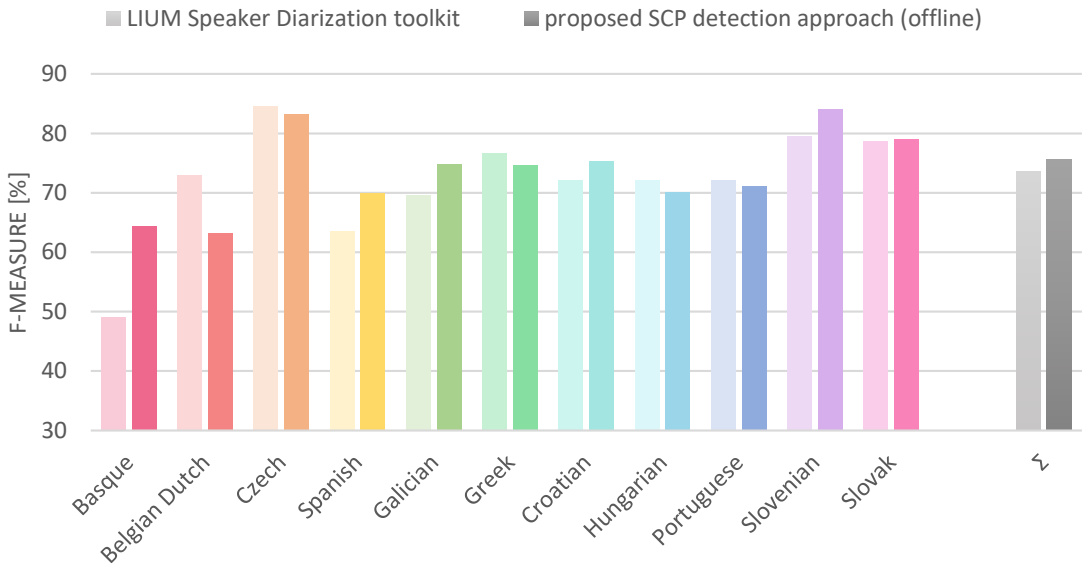

Figure 6.10: A comparison of the proposed SCP detection approach (tuned for offline use) with the reference system on the whole COST278 database. Lighter columns mark the reference system while the darker ones indicate the proposed approach.

6.11.5 Training Data

For the final experiment, the influence of different training data on the performance of the online configuration of the proposed SCP detection approach was explored. This experimental evaluation was done on the whole COST278 test subset. Within the evaluation, the proposed approach was trained using three different training datasets: a) a dataset consisting of the data specifically selected and prepared for this work (enhanced training data); b) a COST278 training dataset; and c) a dataset combining both previous datasets.

The global results are summarized in Table 6.8. They show that the best overall performance had the systems utilizing the training dataset of the COST278 database. This was most likely caused by two major things. First, these systems were trained on more diverse data making them more robust to different languages. Second, the train and test subsets of the COST278 database share many similarities, including, e.g., the same languages or speakers.

Table 6.8: Summarized results studying the influence of different training data on the performance of the proposed SCP detection approach tuned for online use.

training data	P [%]	R [%]	F [%]	$\delta_{2/3}[s]$	RTF	L [s]
enhanced data	58.9	59.2	59.0	0.18	**0.064**	**2.9**
COST278 train set	**73.2**	**72.9**	**73.1**	0.15	**0.064**	**2.9**
combined data	70.6	70.5	70.6	**0.14**	**0.064**	**2.9**

Figure 6.11 shows the detailed results for all languages of the COST278 database. Expectedly, the system trained only on data specifically selected and prepared for this work outperformed other systems on the Czech test subset. This is by design due to the selection of Czech broadcast training data to fit the target application of the TVR monitoring system (i.e., most of the transcriptions are in Czech). Next, both the systems trained on the training subset of the COST278 database performed on a somewhat similar level. The one using combined data improved on Czech (understandably) and Slovak (closely related language to Czech) languages but worsened on more distant ones. Finally, and most importantly, the choice of training data forms an integral part in the process of designing the proposed approach, and it is strongly dependant on the target application.

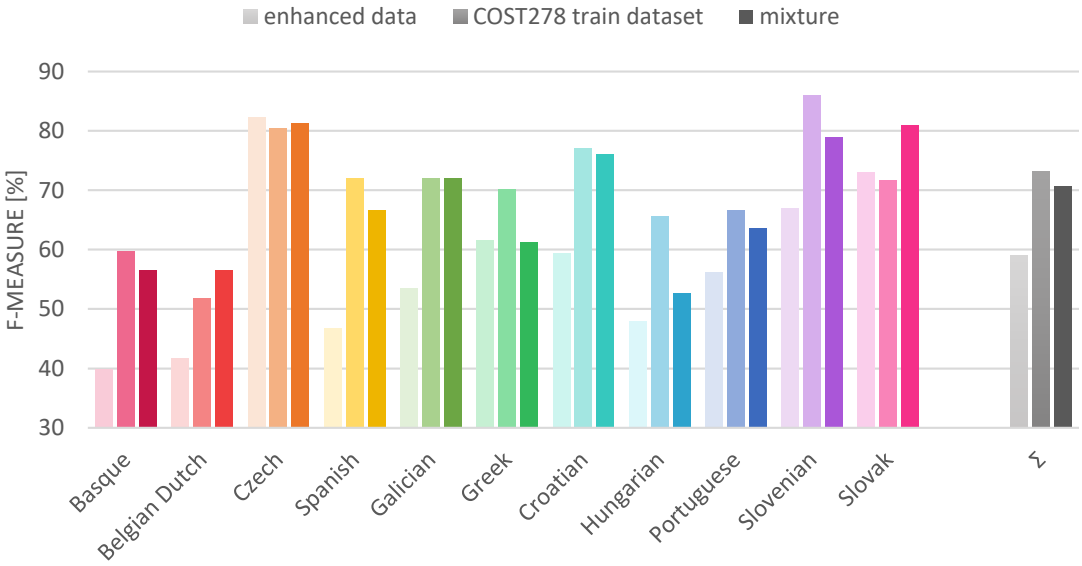

Figure 6.11: A comparison of the proposed SCP detection approach (tuned for online use) trained on the enhanced data, training dataset of COST278 database, and combined data. The lightest columns mark the system trained on the enhanced data; the middle columns indicate the system trained on the training subset of the COST278 database, and finally, the darkest columns denote the system trained on both datasets.

7 Conclusions

Within the scope of this thesis, the tasks of speech activity detection and speaker change point detection with the focus on modern technologies and their application in an online monitoring system as speech preprocessors have been explored. A novel approach has been proposed for speech activity detection as well as for speaker change point detection. This thesis closely follows and describes the development of both of these approaches from the initial to the final stages. All the steps taken are discussed in detail and backed up by a diverse set of experiments. Ultimately, both of these approaches have been designed to be integrated into the TVR monitoring system developed at SpeechLab@TUL in cooperation with the NanoTrix company, and they both support a crucial online mode.

Speech Activity Detection

The final proposed speech activity detection approach is based on two main components: a feed-forward deep neural network and a context-based weighted finite-state transducer. The first component, DNN, functions as a frame classifier (speech/non-speech and context states), while the latter component, WFST, is an online decoder which smooths the outputs of the classifier. The network is trained on log filter bank coefficients of artificially created data by mixing speech and non-speech recordings at various levels of SNR. The data has also been enriched by various noises. This design yields state-of-the-art results under low- and medium-noise conditions on the standardized QUT-NOISE-TIMIT dataset. Moreover, it also operates with a low real-time factor as well as low latency, which makes it a suitable option for online processing. An evaluation in a real speech transcription system has yielded a significant improvement in RTF as well as a slight boost in accuracy of the transcription.

The initial research introducing the main concept and a simple transduction model was presented in [1] at SIGMAP 2016 held in Lisbon. The improved and final context-based transduction model was introduced in [2] at ICASSP 2017 organized in New Orleans. Finally, an extended version detailing more experiments with QUT-NOISE-TIMIT corpus was published in [3].

Potential improvements could be focused on improving the latency even further. This could be achieved by, e.g., designing a different transduction model or employing diverse deep classifiers and fine-tuning their hyper-parameters. Additionally, more complex features could be crafted. Lastly, enrichment of training data by various broadcast noises could achieve more robust speech activity detection and yield even better speech/non-speech segmentation.

Speaker Change Point Detection

The final design of the proposed speaker change point detection approach is inspired by the proposed speech activity detection design. It consists of two main components: a convolutional neural network and a weighted finite-state transducer with a

forced length of transition. The convolutional neural network plays the role of a binary frame classifier (change point/no change point) while the weighted finite-state transducer is utilized as an online decoder smoothing the output of CNN. The decoder also enforces the duration of the transition from one speaker to another. The network is trained on TV/radio broadcast data complemented by artificial examples to reduce different types of errors. Safety collar frames are labeled around the actual change points to improve the performance of the system, and MFCCs with Δ and $\Delta\Delta$ are used as input features. On data taken from the COST278 database, the proposed approach achieves results approaching the offline multi-pass reference system (LIUM Speaker Diarization toolkit) while operating online with low latency.

The whole research explaining in detail the proposed speaker change point detection approach was presented in [4] at Interspeech 2019 conference in Graz.

The performance of the SCP detection approach could be further improved by implementing online clustering, which should diminish falsely predicted transitions between speakers. It is a common practice in the literature. An exploration of more robust features or different deep neural network architectures (e.g., time delay convolutional neural networks are gaining in popularity nowadays) could yield progress as well. Similarly to SAD, other transduction WFST models could be designed. Finally, additional varied training data could be collected to craft a more robust approach yielding even better results for diverse languages.

Summary of Research Contributions

Within this thesis, the following has been covered:

- an overview of the current state of the art in both speech activity detection and speaker change point detection with additional focus on existing toolkits;
- a detailed description of selected approaches to the SAD and SCP detection relevant to this work or focused on the online application;
- a detailed description of the design and development of the proposed SAD approach performing robust speech/non-speech detection;
- experimental tuning of the proposed SAD approach;
- an evaluation of the proposed SAD approach and its comparison with various SAD approaches on the standardized QUT-NOISE-TIMIT corpus;
- an evaluation of the proposed SAD approach in a real speech transcription system;
- a detailed description of the design and development of the proposed SCP detection approach performing speaker change point detection;
- experimental tuning of the proposed SCP detection approach;
- an evaluation of the proposed SCP detection approach and its comparison with a reference system on the standardized COST278 database;

- an evaluation of the online performance of both SAD and SCP detection approaches.

Summary of Practical Use Contributions

The main contribution of this thesis to the field of practical applications is the ability to integrate the proposed speech activity detection and speaker change point detection approaches into the TVR monitoring system developed at the author's lab in cooperation with the NanoTrix company.

The proposed SAD approach is now fully integrated into this TVR monitoring system. Last month, approximately 4,130 days (99,100 hours or 2.3 TB) of recordings were transcribed in the processing time of 1,333 days (32,000 hours). Considering the real-time factor of the speech transcriber being around one, the deployment of SAD (as a preprocessor) resulted in significantly saved processing time. Approximately two-thirds of the data was non-speech and thus omitted from the transcription. This was supplemented by a slight increase in accuracy of the transcriber as the non-speech parts were not transcribed into gibberish.

The proposed SCP detection approach is now ready to be integrated into this TVR monitoring system. Once done, it will be used to label speaker-homogeneous segments in multiple online broadcast streams (i.e., it will break the streams into smaller chunks, each containing only one speaker). By doing this, it will provide the transcribed data with additional information that could be further utilized and expanded upon. It will also form a stepping stone for further diarization functionality.

In general, both the SAD and SCP detection approaches can be used for any application that needs speech preprocessing, even the ones requiring online use.

Future Work

The fully implemented speech activity detection and speaker change point detection approaches are the first steps in the process of designing a speaker diarization system and successively speaker verification and identification systems and integrating them into a TVR monitoring system. In conjunction with SAD, the SCP detector produces an ever-growing amount of labels for speaker-homogeneous speech segments. These newly defined segments will be utilized for, e.g., language identification (the online version is already being worked on while the offline version was published in [165] at Interspeech 2018), gender, or emotion recognition. Their application to speaker-adaptive speech recognition is also planned in the future.

References

[1] L. Mateju, P. Cerva, and J. Zdansky. "Study on the Use of Deep Neural Networks for Speech Activity Detection in Broadcast Recordings". In: *13th International Joint Conference on e-Business and Telecommunications (ICETE 2016) - Volume 5: SIGMAP, Lisbon, Portugal, July 26-28, 2016*. SciTePress, 2016, pp. 45–51.

[2] L. Mateju, P. Cerva, J. Zdansky, and J. Malek. "Speech Activity Detection in Online Broadcast Transcription Using Deep Neural Networks and Weighted Finite State Transducers". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, Louisiana, USA, March 5-9, 2017*. IEEE, 2017, pp. 5460–5464.

[3] L. Mateju, P. Cerva, and J. Zdansky. "Investigation into the Use of WFSTs and DNNs for Speech Activity Detection in Broadcast Data Transcription". In: *E-Business and Telecommunications - 13th International Joint Conference, ICETE 2016, Lisbon, Portugal, July 26-28, 2016, Revised Selected Papers*. Springer, 2017, pp. 341–358.

[4] L. Mateju, P. Cerva, and J. Zdansky. "An Approach to Online Speaker Change Point Detection Using DNNs and WFSTs". In: *INTERSPEECH 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, September 15-19, 2019*. ISCA, 2019, pp. 649–653.

[5] G. Evangelopoulos and P. Maragos. "Speech Event Detection Using Multiband Modulation Energy". In: *INTERSPEECH 2005 - Eurospeech, 9th European Conference on Speech Communication and Technology, Lisbon, Portugal, September 4-8, 2005*. ISCA, 2005, pp. 685–688.

[6] B. Kotnik, Z. Kacic, and B. Horvat. "A Multiconditional Robust Front-End Feature Extraction with a Noise Reduction Procedure Based on Improved Spectral Subtraction Algorithm". In: *INTERSPEECH 2001 - Eurospeech, 7th European Conference on Speech Communication and Technology, Aalborg, Denmark, September 3-7, 2001*. ISCA, 2001, pp. 197–200.

[7] H. Ghaemmaghami, B. Baker, R. Vogt, and S. Sridharan. "Noise Robust Voice Activity Detection Using Features Extracted from the Time-Domain Autocorrelation Function". In: *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*. ISCA, 2010, pp. 3118–3121.

[8] N. Ryant, M. Liberman, and J. Yuan. "Speech Activity Detection on YouTube Using Deep Neural Networks". In: *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*. ISCA, 2013, pp. 728–731.

[9] K. Sriskandaraja, V. Sethu, P. N. Le, and E. Ambikairajah. "A Model Based Voice Activity Detector for Noisy Environments". In: *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 2297–2301.

[10] X. Zhang and D. Wang. "Boosted Deep Neural Networks and Multi-Resolution Cochleagram Features for Voice Activity Detection". In: *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*. ISCA, 2014, pp. 1534–1538.

[11] Y. Shao and Q. Lin. "Use of Pitch Continuity for Robust Speech Activity Detection". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, Alberta, Canada, April 15-20, 2018*. IEEE, 2018, pp. 5534–5538.

[12] A. Tsiartas, T. Chaspari, N. Katsamanis, P. K. Ghosh, M. Li, M. V. Segbroeck, A. Potamianos, and S. Narayanan. "Multi-Band Long-Term Signal Variability Features for Robust Voice Activity Detection". In: *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*. ISCA, 2013, pp. 718–722.

[13] T. Kinnunen, A. Sholokhov, E. el Khoury, D. A. L. Thomsen, M. Sahidullah, and Z. Tan. "HAPPY Team Entry to NIST OpenSAD Challenge: A Fusion of Short-Term Unsupervised and Segment i-Vector Based Speech Activity Detectors". In: *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, California, USA, September 8-12, 2016*. ISCA, 2016, pp. 2992–2996.

[14] J. Ma. "Improving the Speech Activity Detection for the DARPA RATS Phase-3 Evaluation". In: *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*. ISCA, 2014, pp. 1558–1562.

[15] L. Ferrer, M. Graciarena, and V. Mitra. "A phonetically aware system for speech activity detection". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*. IEEE, 2016, pp. 5710–5714.

[16] X. Zhang and J. Wu. "Deep Belief Networks Based Voice Activity Detection". In: *IEEE Transactions Audio, Speech & Language Processing* 21.4 (2013), pp. 697–710.

[17] M. V. Segbroeck, A. Tsiartas, and S. Narayanan. "A robust frontend for VAD: exploiting contextual, discriminative and spectral cues of human voice". In: *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*. ISCA, 2013, pp. 704–708.

[18] M. Graciarena, L. Ferrer, and V. Mitra. “The SRI System for the NIST OpenSAD 2015 Speech Activity Detection Evaluation”. In: *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, California, USA, September 8-12, 2016*. ISCA, 2016, pp. 3673–3677.

[19] J. W. Shin, J. Chang, and N. S. Kim. “Voice Activity Detection Based on Statistical Models and Machine Learning Approaches”. In: *Computer Speech & Language* 24.3 (2010), pp. 515–530.

[20] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Vesely, and P. Matejka. “Developing a Speech Activity Detection System for the DARPA RATS Program”. In: *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*. ISCA, 2012, pp. 1969–1972.

[21] A. Misra. “Speech/Nonspeech Segmentation in Web Videos”. In: *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*. ISCA, 2012, pp. 1977–1980.

[22] H. Ghaemmaghami, D. Dean, S. Kalantari, S. Sridharan, and C. Fookes. “Complete-Linkage Clustering for Voice Activity Detection in Audio and Visual Speech”. In: *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 2292–2296.

[23] L. Wang, C. Zhang, P. C. Woodland, M. J. F. Gales, P. Karanasou, P. Lanchantin, X. Liu, and Y. Qian. “Improved DNN-based segmentation for multi-genre broadcast audio”. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*. IEEE, 2016, pp. 5700–5704.

[24] I. Jang, C. Ahn, J. Seo, and Y. Jang. “Enhanced Feature Extraction for Speech Detection in Media Audio”. In: *INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*. ISCA, 2017, pp. 479–483.

[25] G. Saon, S. Thomas, H. Soltau, S. Ganapathy, and B. Kingsbury. “The IBM Speech Activity Detection System for the DARPA RATS Program”. In: *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*. ISCA, 2013, pp. 3497–3501.

[26] R. Zazo, T. N. Sainath, G. Simko, and C. Parada. “Feature Learning with Raw-Waveform CLDNNs for Voice Activity Detection”. In: *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, California, USA, September 8-12, 2016*. ISCA, 2016, pp. 3668–3672.

[27] S. Chang, B. Li, G. Simko, T. N. Sainath, A. Tripathi, A. van den Oord, and O. Vinyals. "Temporal Modeling Using Dilated Convolution and Gating for Voice-Activity-Detection". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, Alberta, Canada, April 15-20, 2018*. IEEE, 2018, pp. 5549–5553.

[28] T. Hughes and K. Mierle. "Recurrent Neural Networks for Voice Activity Detection". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, British Columbia, Canada, May 26-31, 2013*. IEEE, 2013, pp. 7378–7382.

[29] F. Eyben, F. Weninger, S. Squartini, and B. W. Schuller. "Real-Life Voice Activity Detection with LSTM Recurrent Neural Networks and an Application to Hollywood Movies". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, British Columbia, Canada, May 26-31, 2013*. IEEE, 2013, pp. 483–487.

[30] D. Karakos, S. Novotney, L. Zhang, and R. M. Schwartz. "Model Adaptation and Active Learning in the BBN Speech Activity Detection System for the DARPA RATS Program". In: *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, California, USA, September 8-12, 2016*. ISCA, 2016, pp. 3678–3682.

[31] Q. Wang, J. Du, X. Bao, Z. Wang, L. Dai, and C. Lee. "A Universal VAD Based on Jointly Trained Deep Neural Networks". In: *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 2282–2286.

[32] S. Thomas, G. Saon, M. V. Segbroeck, and S. S. Narayanan. "Improvements to the IBM Speech Activity Detection System for the DARPA RATS Program". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*. IEEE, 2015, pp. 4500–4504.

[33] Y. Obuchi. "Framewise speech-nonspeech classification by neural networks for voice activity detection with statistical noise suppression". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*. IEEE, 2016, pp. 5715–5719.

[34] J. Kim and M. Hahn. "Voice Activity Detection Using an Adaptive Context Attention Model". In: *IEEE Signal Processing Letters* 25.8 (2018), pp. 1181–1185.

[35] H. Chung, S. J. Lee, and Y. Lee. "Endpoint Detection Using Weighted Finite State Transducer". In: *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*. ISCA, 2013, pp. 700–703.

[36] C. Gao, G. Saikumar, S. Khanwalkar, A. Herscovici, A. Kumar, A. Srivastava, and P. Natarajan. “Online Speech Activity Detection in Broadcast News”. In: *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*. ISCA, 2011, pp. 2637–2640.

[37] B. Liu, B. Hoffmeister, and A. Rastrow. “Accurate Endpointing with Expected Pause Duration”. In: *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 2912–2916.

[38] D. Cournapeau and T. Kawahara. “Evaluation of real-time voice activity detection based on high order statistics”. In: *INTERSPEECH 2007, 8th Annual Conference of the International Speech Communication Association, Antwerp, Belgium, August 27-31, 2007*. ISCA, 2007, pp. 2945–2948.

[39] D. Cournapeau, S. Watanabe, A. Nakamura, and T. Kawahara. “Using online model comparison in the Variational Bayes framework for online unsupervised Voice Activity Detection”. In: *2010 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2010, Dallas, Texas, USA, March 14-19, 2010*. IEEE, 2010, pp. 4462–4465.

[40] M. H. Moattar and M. M. Homayounpour. “A Simple but Efficient Real-Time Voice Activity Detection Algorithm”. In: *17th European Signal Processing Conference, EUSIPCO 2009, Glasgow, United Kingdom, August 24-28, 2009*. IEEE, 2009, pp. 2549–2553.

[41] I. Tashev and S. Mirsamadi. “DNN-based Causal Voice Activity Detector”. In: *Information Theory and Applications Workshop*. University of California – San Diego, Feb. 2016.

[42] J. Zelinka. “Deep Learning and Online Speech Activity Detection for Czech Radio Broadcasting”. In: *Text, Speech, and Dialogue - 21st International Conference, TSD 2018, Brno, Czech Republic, September 11-14, 2018*. Springer, 2018, pp. 428–435.

[43] J. H. Ko, J. Fromm, M. Philipose, I. Tashev, and S. Zarar. “Limiting Numerical Precision of Neural Networks to Achieve Real-Time Voice Activity Detection”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, Alberta, Canada, April 15-20, 2018*. IEEE, 2018, pp. 2236–2240.

[44] D. Dean, S. Sridharan, R. Vogt, and M. Mason. “The QUT-NOISE-TIMIT Corpus for the Evaluation of Voice Activity Detection Algorithms”. In: *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*. ISCA, 2010, pp. 3110–3113.

[45] D. Snyder, G. Chen, and D. Povey. “MUSAN: A Music, Speech, and Noise Corpus”. In: *CoRR* abs/1510.08484 (2015).

[46] S. Chaudhuri, J. Roth, D. P. W. Ellis, A. C. Gallagher, L. Kaver, R. Marvin, C. Pantofaru, N. Reale, L. G. Reid, K. W. Wilson, and Z. Xi. "AVA-Speech: A Densely Labeled Dataset of Speech Activity in Movies". In: *INTERSPEECH 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, September 2-6, 2018*. ISCA, 2018, pp. 1239–1243.

[47] D. Wang, L. Lu, and H. Zhang. "Speech segmentation without speech recognition". In: *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2003, Hong Kong, April 6-10, 2003*. IEEE, 2003, pp. 468–471.

[48] R. Sinha, S. E. Tranter, M. J. F. Gales, and P. C. Woodland. "The Cambridge University March 2005 Speaker Diarisation System". In: *INTERSPEECH 2005 - Eurospeech, 9th European Conference on Speech Communication and Technology, Lisbon, Portugal, September 4-8, 2005*. ISCA, 2005, pp. 2437–2440.

[49] S. Meignier, D. Moraru, C. Fredouille, J. Bonastre, and L. Besacier. "Step-by-Step and Integrated Approaches in Broadcast News Speaker Diarization". In: *Computer Speech & Language* 20.2-3 (2006), pp. 303–330.

[50] L. Lu, H. Zhang, and H. Jiang. "Content Analysis for Audio Classification and Segmentation". In: *IEEE Transactions Speech and Audio Processing* 10.7 (2002), pp. 504–516.

[51] B. Desplanques, K. Demuynck, and J. Martens. "Factor Analysis for Speaker Segmentation and Improved Speaker Diarization". In: *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 3081–3085.

[52] L. V. Neri, H. N. B. Pinheiro, T. I. Ren, G. D. C. Cavalcanti, and A. G. Adami. "Speaker Segmentation Using i-vector in Meetings Domain". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, Louisiana, USA, March 5-9, 2017*. IEEE, 2017, pp. 5455–5459.

[53] K. Chen and A. Salman. "Learning Speaker-Specific Characteristics With a Deep Neural Architecture". In: *IEEE Transactions Neural Networks* 22.11 (2011), pp. 1744–1756.

[54] A. Sarkar, S. Dasgupta, S. K. Naskar, and S. Bandyopadhyay. "Says Who? Deep Learning Models for Joint Speech Recognition, Segmentation and Diarization". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, Alberta, Canada, April 15-20, 2018*. IEEE, 2018, pp. 5229–5233.

[55] R. Wang, M. Gu, L. Li, M. Xu, and T. F. Zheng. “Speaker Segmentation Using Deep Speaker Vectors for Fast Speaker Change Scenarios”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, Louisiana, USA, March 5-9, 2017*. IEEE, 2017, pp. 5420–5424.

[56] H. Bredin. “TristouNet: Triplet Loss for Speaker Turn Embedding”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, Louisiana, USA, March 5-9, 2017*. IEEE, 2017, pp. 5430–5434.

[57] A. Jati and P. G. Georgiou. “Speaker2Vec: Unsupervised Learning and Adaptation of a Speaker Manifold Using Deep Neural Networks with an Evaluation on Speaker Segmentation”. In: *INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*. ISCA, 2017, pp. 3567–3571.

[58] A. Zhang, Q. Wang, Z. Zhu, J. W. Paisley, and C. Wang. “Fully Supervised Speaker Diarization”. In: *2019 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*. IEEE, 2019, pp. 6301–6305.

[59] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, and S. Khudanpur. “Diarization is Hard: Some Experiences and Lessons Learned for the JHU Team in the Inaugural DIHARD Challenge”. In: *INTERSPEECH 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, September 2-6, 2018*. ISCA, 2018, pp. 2808–2812.

[60] S. S. Chen and P. S. Gopalakrishnan. “Speaker, Environment and Channel Change Detection and Clustering Via the Bayesian Information Criterion”. In: *DARPA Broadcast News Transcription and Understanding Workshop*. 1998, pp. 127–132.

[61] P. Sivakumaran, J. Fortuna, and A. M. Ariyaeeinia. “On the use of the Bayesian information criterion in multiple speaker detection”. In: *INTERSPEECH 2001 - Eurospeech, 7th European Conference on Speech Communication and Technology, Aalborg, Denmark, September 3-7, 2001*. ISCA, 2001, pp. 795–798.

[62] M. Cettolo, M. Vescovi, and R. Rizzi. “Evaluation of BIC-Based Algorithms for Audio Segmentation”. In: *Computer Speech & Language* 19.2 (2005), pp. 147–170.

[63] H. Gish, M. H. Siu, and R. Rohlicek. “Segregation of Speakers for Speech Recognition and Speaker Identification”. In: *1991 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1991, Toronto, Ontario, Canada, May 14-17, 1991*. IEEE, 1991, pp. 873–876.

[64] C. Barras, X. Zhu, S. Meignier, and J. Gauvain. “Multistage Speaker Diarization of Broadcast News”. In: *IEEE Transactions Audio, Speech & Language Processing* 14.5 (2006), pp. 1505–1512.

[65] M. A. Siegler, U. Jain, B. Raj, and R. M. Stern. “Automatic Segmentation, Classification and Clustering of Broadcast News Audio”. In: *DARPA Speech Recognition Workshop*. 1997, pp. 97–99.

[66] B. Fergani, M. Davy, and A. Houacine. “Speaker diarization using one-class support vector machines”. In: *Speech Communication* 50.5 (2008), pp. 355–365.

[67] P. Delacourt and C. Wellekens. “DISTBIC: A speaker-based segmentation for audio data indexing”. In: *Speech Communication* 32.1-2 (2000), pp. 111–126.

[68] S. Meignier, J. Bonastre, and S. Igounet. “E-HMM Approach for Learning and Adapting Sound Models for Speaker Indexing”. In: *2001: A Speaker Odyssey - The Speaker Recognition Workshop, Crete, Greece, June 18-22, 2001*. ISCA, 2001, pp. 175–180.

[69] A. S. Malegaonkar, A. M. Ariyaeeinia, and P. Sivakumaran. “Efficient Speaker Change Detection Using Adapted Gaussian Mixture Models”. In: *IEEE Transactions Audio, Speech & Language Processing* 15.6 (2007), pp. 1859–1869.

[70] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, and C. Vair. “Stream-Based Speaker Segmentation Using Speaker Factors and Eigenvoices”. In: *2008 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2008, Las Vegas, Nevada, USA, March 30 - April 4, 2008*. IEEE, 2008, pp. 4133–4136.

[71] V. Gupta. “Speaker Change Point Detection Using Deep Neural Nets”. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*. IEEE, 2015, pp. 4420–4424.

[72] M. Hruz and M. Kunesova. “Convolutional Neural Network in the Task of Speaker Change Detection”. In: *Speech and Computer - 18th International Conference, SPECOM 2016, Budapest, Hungary, August 23-27, 2016*. Springer, 2016, pp. 191–198.

[73] M. Hruz and Z. Zajic. “Convolutional Neural Network for Speaker Change Detection in Telephone Speaker Diarization System”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, Louisiana, USA, March 5-9, 2017*. IEEE, 2017, pp. 4945–4949.

[74] M. India, J. A. R. Fonollosa, and J. Hernando. “LSTM Neural Network-Based Speaker Segmentation Using Acoustic and Language Modelling”. In: *INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*. ISCA, 2017, pp. 2834–2838.

[75] R. Yin, H. Bredin, and C. Barras. “Speaker Change Detection in Broadcast TV Using Bidirectional Long Short-Term Memory Networks”. In: *INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*. ISCA, 2017, pp. 3827–3831.

[76] M. Hruz and M. Hlavac. “LSTM Neural Network for Speaker Change Detection in Telephone Conversations”. In: *Speech and Computer - 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18-22, 2018*. Springer, 2018, pp. 226–233.

[77] D. Moraru, S. Meignier, C. Fredouille, L. Besacier, and J. Bonastre. “The ELISA Consortium Approaches in Broadcast News Speaker Segmentation During the NIST 2003 Rich Transcription Evaluation”. In: *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2004, Montreal, Quebec, Canada, May 17-21, 2004*. IEEE, 2004, pp. 373–376.

[78] L. Lu and H. Zhang. “Speaker Change Detection and Tracking in Real-Time News Broadcasting Analysis”. In: *10th ACM International Conference on Multimedia 2002, Juan les Pins, France, December 1-6, 2002*. ACM, 2002, pp. 602–610.

[79] L. Lu and H. Zhang. “Unsupervised Speaker Segmentation and Tracking in Real-Time Audio Content Analysis”. In: *Multimedia Systems* 10.4 (2005), pp. 332–343.

[80] M. Kotti, L. P. M. Martins, E. Benetos, J. S. Cardoso, and C. Kotropoulos. “Automatic Speaker Segmentation using Multiple Features and Distance Measures: A Comparison of Three Approaches”. In: *2006 IEEE International Conference on Multimedia and Expo, ICME 2006, Toronto, Ontario, Canada, July 9-12, 2006*. IEEE, 2006, pp. 1101–1104.

[81] M. Grasic, M. Kos, and Z. Kacic. “Online speaker segmentation and clustering using cross-likelihood ratio calculation with reference criterion selection”. In: *IET Signal Processing* 4.6 (2010), pp. 673–685.

[82] X. Anguera. “Xbic: Real-time cross probabilities measure for speaker segmentation”. In: *ICSI* (2005), pp. 1–8.

[83] J. Ajmera, I. McCowan, and H. Bourlard. “Robust speaker change detection”. In: *IEEE Signal Processing Letters* 11.8 (2004), pp. 649–651.

[84] K. Markov and S. Nakamura. “Never-ending learning system for on-line speaker diarization”. In: *2007 IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU 2007, Kyoto, Japan, December 9-13, 2007*. IEEE, 2007, pp. 699–704.

[85] J. T. Geiger, F. Wallhoff, and G. Rigoll. “GMM-UBM Based Open-Set Online Speaker Diarization”. In: *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*. ISCA, 2010, pp. 2330–2333.

[86] T. Wu, L. Lu, K. Chen, and H. Zhang. “Universal Background Models for Real-time Speaker Change Detection”. In: *9th International Conference on Multi-Media Modeling, MMM 2003, Taiwan, January 7-10, 2003*. IEEE, 2003, pp. 135–149.

[87] D. Dimitriadis and P. Fousek. “Developing On-Line Speaker Diarization System”. In: *INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*. ISCA, 2017, pp. 2739–2743.

[88] W. Zhu and J. W. Pelecanos. “Online Speaker Diarization Using Adapted i-vector Transforms”. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*. IEEE, 2016, pp. 5045–5049.

[89] Z. Ge, A. N. Iyer, S. Cheluvaraja, and A. Ganapathiraju. “Speaker Change Detection Using Features through a Neural Network Speaker Classifier”. In: *2017 Intelligent Systems Conference (IntelliSys), 2017, London, United Kingdom, September 7-8, 2017*. IEEE, 2017, pp. 1111–1116.

[90] M. Kunesova, Z. Zajic, and V. Radova. “Experiments with Segmentation in an Online Speaker Diarization System”. In: *Text, Speech, and Dialogue - 20th International Conference, TSD 2017, Prague, Czech Republic, August 27-31, 2017*. Springer, 2017, pp. 429–437.

[91] R. Stern. “Specifications of the 1996 Hub-4 Broadcast News Evaluation”. In: *DARPA Speech Recognition Workshop*. 1997.

[92] S. Galliano, E. Geoffrois, G. Gravier, J. Bonastre, D. Mostefa, and K. Choukri. “Corpus Description of the ESTER Evaluation Campaign for the Rich Transcription of French Broadcast News”. In: *Fifth International Conference on Language Resources and Evaluation, LREC 2006, Genoa, Italy, May 22-28, 2006*. ELRA, 2006, pp. 139–142.

[93] O. Galibert, J. Leixa, G. Adda, K. Choukri, and G. Gravier. “The ETAPE Speech Processing Evaluation”. In: *Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*. ELRA, 2014, pp. 3995–3999.

[94] O. Galibert and J. Kahn. “The First Official REPERE Evaluation”. In: *First Workshop on Speech, Language and Audio in Multimedia, Marseille, France, August 22-23, 2013*. CEUR-WS.org, 2013, pp. 43–48.

[95] A. Vandecatseye, J. Martens, J. P. Neto, H. Meinedo, C. Garcia-Mateo, J. Dieguez-Tirado, F. Mihelic, J. Zibert, J. Nouza, P. David, M. Pleva, A. Cizmar, H. Papageorgiou, and C. Alexandris. “The COST278 Pan-European Broadcast News Database”. In: *Fourth International Conference on Language Resources and Evaluation, LREC 2004, Lisbon, Portugal, May 26-28, 2004*. ELRA, 2004, pp. 873–876.

[96] J. Zibert, F. Mihelic, J. Martens, H. Meinedo, J. P. Neto, L. D. Fernandez, C. Garcia-Mateo, P. David, J. Zdansky, M. Pleva, A. Cizmar, A. Zgank, Z. Kacic, C. Teleki, and K. Vicsi. "The COST278 broadcast news segmentation and speaker clustering evaluation - overview, methodology, systems, results". In: *INTERSPEECH 2005 - Eurospeech, 9th European Conference on Speech Communication and Technology, Lisbon, Portugal, September 4-8, 2005*. ISCA, 2005, pp. 629–632.

[97] J. Bonastre, F. Wils, and S. Meignier. "ALIZE, a Free Toolkit for Speaker Recognition". In: *2005 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2005, Philadelphia, Pennsylvania, USA, March 18-23, 2005*. IEEE, 2005, pp. 737–740.

[98] A. Larcher, J. Bonastre, B. G. B. Fauve, K. Lee, C. Levy, H. Li, J. S. D. Mason, and J. Parfait. "ALIZE 3.0 - Open Source Toolkit for State-of-the-Art Speaker Recognition". In: *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*. ISCA, 2013, pp. 2768–2772.

[99] S. Bozonnet, N. W. D. Evans, and C. Fredouille. "The Lia-Eurecom RT'09 Speaker Diarization System: Enhancements in Speaker Modelling and Cluster Purification". In: *2010 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2010, Dallas, Texas, USA, March 14-19, 2010*. ISCA, 2010, pp. 4958–4961.

[100] E. el Khoury, L. E. Shafey, and S. Marcel. "Spear: An open source toolbox for speaker recognition based on Bob". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*. IEEE, 2014, pp. 1655–1659.

[101] S. Meignier and T. Merlin. "LIUM SpkDiarization: an Open Source Toolkit for Diarization". In: *CMU SPUD Workshop, Dallas, Texas, USA, March 13, 2010*. 2010.

[102] M. Rouvier, G. Dupuy, P. Gay, E. el Khoury, T. Merlin, and S. Meignier. "An Open-Source State-of-the-Art Toolbox for Broadcast News Diarization". In: *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*. ISCA, 2013, pp. 1477–1481.

[103] D. Vijayasenan and F. Valente. "DiarTk : An Open Source Toolkit for Research in Multistream Speaker Diarization and its Application to Meetings Recordings". In: *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*. ISCA, 2012, pp. 2170–2173.

[104] R. Yin, H. Bredin, and C. Barras. "Neural Speech Turn Segmentation and Affinity Propagation for Speaker Diarization". In: *INTERSPEECH 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, September 2-6, 2018*. ISCA, 2018, pp. 1393–1397.

[105] P. Broux, F. Desnous, A. Larcher, S. Petitrenaud, J. Carrive, and S. Meignier. "S4D: Speaker Diarization Toolkit in Python". In: *INTERSPEECH 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, September 2-6, 2018*. ISCA, 2018, pp. 1368–1372.

[106] A. Larcher, K. Lee, and S. Meignier. "An extensible speaker identification sidekit in Python". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*. IEEE, 2016, pp. 5095–5099.

[107] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur. "X-Vectors: Robust DNN Embeddings for Speaker Recognition". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, Alberta, Canada, April 15-20, 2018*. IEEE, 2018, pp. 5329–5333.

[108] 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: *2011 IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU 2011, Waikoloa, Hawaii, USA, December 11-15, 2011*. IEEE, 2011, pp. 1–4.

[109] G. E. Hinton, S. Osindero, and Y. W. Teh. "A Fast Learning Algorithm for Deep Belief Nets". In: *Neural Computation* 18.7 (2006), pp. 1527–1554.

[110] G. E. Dahl, D. Yu, L. Deng, and A. Acero. "Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition". In: *IEEE Transactions Audio, Speech & Language Processing* 20.1 (2012), pp. 30–42.

[111] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups". In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 82–97.

[112] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Y. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Y. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu. "Deep Speech 2: End-to-End Speech Recognition in English and Mandarin". In: *CoRR* abs/1512.02595 (2015).

[113] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012, Lake Tahoe, Nevada, USA, December 3-6, 2012*. Curran Associates, Inc., 2012, pp. 1106–1114.

[114] K. He, X. Zhang, S. Ren, and J. Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, Nevada, USA, June 27-30, 2016*. IEEE, 2016, pp. 770–778.

[115] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. “Distributed Representations of Words and Phrases and their Compositionality”. In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, Nevada, USA, December 5-8, 2013*. Curran Associates, Inc., 2013, pp. 3111–3119.

[116] I. Sutskever, O. Vinyals, and Q. V. Le. “Sequence to Sequence Learning with Neural Networks”. In: *Advances in Neural Information Processing Systems 27: 28th Annual Conference on Neural Information Processing Systems 2014, Montreal, Quebec, Canada, December 8-13, 2014*. Curran Associates, Inc., 2014, pp. 3104–3112.

[117] “Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear Prediction (CS-ACELP)”. In: *ITU-T Recommendation G.729* (1996), pp. 1–152.

[118] R. Salami, C. Laflamme, B. Bessette, and J. P. Adoul. “ITU-T G.729 Annex A: Reduced Complexity 8 kb/s CS-ACELP Codec for Digital Simultaneous Voice and Data”. In: *IEEE Communications Magazine* 35.9 (Sept. 1997), pp. 56–63.

[119] A. Benyassine, E. Shlomot, H. Y. Su, D. Massaloux, C. Lamblin, and J. P. Petit. “ITU-T Recommendation G.729 Annex B: a Silence Compression Scheme for Use with G.729 Optimized for V.70 Digital Simultaneous Voice and Data Applications”. In: *IEEE Communications Magazine* 35.9 (Sept. 1997), pp. 64–73.

[120] “Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms”. In: *ETSI ES 202 050 V1.1.5* (2007).

[121] J. Li, B. Liu, R. Wang, and L. Dai. “A Complexity Reduction of ETSI Advanced Front-End for DSR”. In: *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2004, Montreal, Quebec, Canada, May 17-21, 2004*. IEEE, 2004, pp. 61–64.

[122] J. Sohn and W. Sung. “A voice activity detector employing soft decision based noise spectrum adaptation”. In: *1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1998, Seattle, Washington, USA, May 12-15, 1998*. IEEE, 1998, pp. 365–368.

[123] J. Sohn, N. S. Kim, and W. Sung. “A Statistical Model-Based Voice Activity Detection”. In: *IEEE Signal Processing Letters* 6.1 (1999), pp. 1–3.

[124] J. Ramirez, J. C. Segura, M. C. Benitez, A. de la Torre, and A. J. Rubio. “Efficient Voice Activity Detection Algorithms Using Long-Term Speech Information”. In: *Speech Communication* 42.3-4 (2004), pp. 271–287.

[125] P. Mermelstein. “Distance Measures for Speech Recognition: Psychological and Instrumental”. In: *Pattern Recognition and Artificial Intelligence*. Academic Press, 1976, pp. 374–388.

[126] J. Wu and X. Zhang. “An efficient voice activity detection algorithm by combining statistical model and energy detection”. In: *EURASIP Journal on Advances in Signal Processing* 2011 (2011), pp. 18–27.

[127] J. Macqueen. “Some methods for classification and analysis of multivariate observations”. In: *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*. 1967, pp. 281–297.

[128] A. P. Dempster, N. M. Laird, and D. B. Rubin. “Maximum likelihood from incomplete data via the EM algorithm”. In: *Journal of the Royal Statistical Society, Series B* 39.1 (1977), pp. 1–38.

[129] S. Wisdom, G. Okopal, L. E. Atlas, and J. W. Pitton. “Voice Activity Detection Using Subband Noncircularity”. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*. IEEE, 2015, pp. 4505–4509.

[130] S. M. Strassel, A. Morris, J. G. Fiscus, C. Caruso, H. Lee, P. Over, J. Fiumara, B. Shaw, B. Antonishek, and M. Michel. “Creating HAVIC: Heterogeneous Audio Visual Internet Collection”. In: *Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*. ELRA, 2012, pp. 2573–2577.

[131] L. R. Rabiner. “Readings in Speech Recognition”. In: Morgan Kaufmann Publishers Inc., 1990. Chap. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pp. 267–296.

[132] D. Liu and F. Kubala. “Fast speaker change detection for broadcast news transcription and indexing”. In: *INTERSPEECH 1999 - Eurospeech, 6th European Conference on Speech Communication and Technology, Budapest, Hungary, September 5-9, 1999*. ISCA, 1999.

[133] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data”. In: *Eighteenth International Conference on Machine Learning (ICML 2001), Williamstown, Massachusetts, USA, June 28 - July 1, 2001*. 2001, pp. 282–289.

[134] A. Saito, Y. Nankaku, A. Lee, and K. Tokuda. “Voice activity detection based on conditional random fields using multiple features”. In: *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*. ISCA, 2010, pp. 2086–2089.

[135] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999. ISBN: 978-0-387-98793-4.

[136] S. Galliano, G. Gravier, and L. Chaubard. “The ester 2 evaluation campaign for the rich transcription of French radio broadcasts”. In: *INTERSPEECH 2009, 10th Annual Conference of the International Speech Communication Association, Brighton, United Kingdom, September 6-10, 2009*. ISCA, 2009, pp. 2583–2586.

[137] D. Wang, R. Vogt, M. Mason, and S. Sridharan. “Automatic audio segmentation using the Generalized Likelihood Ratio”. In: *2nd International Conference on Signal Processing and Communication Systems (ICSPCS), Gold Coast, Australia, December 15-17, 2008*. IEEE, 2008, pp. 1–5.

[138] L. Lu, H. Jiang, and H. Zhang. “A robust audio classification and segmentation method”. In: *9th ACM International Conference on Multimedia 2001, Ottawa, Ontario, Canada, September 30 - October 5, 2001*. ACM, 2001, pp. 203–211.

[139] L. Lu, H. Zhang, and S. Z. Li. “Content-based audio classification and segmentation by using support vector machines”. In: *Multimedia Systems* 8.6 (2003), pp. 482–492.

[140] F. Itakura. “Line spectrum representation of linear predictor coefficients of speech signals”. In: *The Journal of the Acoustical Society of America* 57.S1 (1975), S35.

[141] S. Kullback and R. A. Leibler. “On Information and Sufficiency”. In: *The Annals of Mathematical Statistics* 22.1 (Mar. 1951), pp. 79–86.

[142] J. P. Campbell. “Speaker recognition: a tutorial”. In: *Proceedings of the IEEE* 85.9 (Sept. 1997), pp. 1437–1462.

[143] A. Tritschler and R. A. Gopinath. “Improved speaker segmentation and segments clustering using the bayesian information criterion”. In: *INTERSPEECH 1999 - Eurospeech, 6th European Conference on Speech Communication and Technology, Budapest, Hungary, September 5-9, 1999*. ISCA, 1999, pp. 679–682.

[144] A. S. Malegaonkar, A. M. Ariyaeeinia, P. Sivakumaran, and J. Fortuna. “Unsupervised speaker change detection using probabilistic pattern matching”. In: *IEEE Signal Processing Letters* 13.8 (2006), pp. 509–512.

[145] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. “Front-End Factor Analysis for Speaker Verification”. In: *IEEE Transactions Audio, Speech & Language Processing* 19.4 (2011), pp. 788–798.

[146] A. Kanagasundaram, R. Vogt, D. Dean, S. Sridharan, and M. Mason. “i-vector Based Speaker Recognition on Short Utterances”. In: *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*. ISCA, 2011, pp. 2341–2344.

[147] S. Shum, N. Dehak, E. Chuangsuwanich, D. A. Reynolds, and J. R. Glass. "Exploiting Intra-Conversation Variability for Speaker Diarization". In: *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*. ISCA, 2011, pp. 945–948.

[148] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak. "Language Recognition via i-vectors and Dimensionality Reduction". In: *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*. ISCA, 2011, pp. 857–860.

[149] D. M. Gonzalez, O. Plchot, L. Burget, O. Glembek, and P. Matejka. "Language Recognition in iVectors Space". In: *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*. ISCA, 2011, pp. 861–864.

[150] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel. "Joint Factor Analysis Versus Eigenchannels in Speaker Recognition". In: *IEEE Transactions Audio, Speech & Language Processing* 15.4 (2007), pp. 1435–1447.

[151] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. "The AMI Meeting Corpus: A Pre-announcement". In: *Machine Learning for Multimodal Interaction, Second International Workshop, MLMI 2005, Edinburgh, United Kingdom, July 11-13, 2005, Revised Selected Papers*. Springer, 2005, pp. 28–39.

[152] B. Zhou and J. H. L. Hansen. "Efficient audio stream segmentation via the combined T^2 statistic and Bayesian information criterion". In: *IEEE Transactions Speech and Audio Processing* 13.4 (2005), pp. 467–474.

[153] W. M. Fisher, G. R. Doddington, and K. M. Goudie-Marshall. "The DARPA Speech Recognition Research Database: Specifications and Status". In: *Proceedings of DARPA Workshop on Speech Recognition*. 1986, pp. 93–99.

[154] E. Variani, X. Lei, E. McDermott, J. Gonzalez-Dominguez, and I. Lopez-Moreno. "Deep neural networks for small footprint text-dependent speaker verification". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*. IEEE, 2014, pp. 4052–4056.

[155] C. Cieri, D. Miller, and K. Walker. "The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text". In: *Fourth International Conference on Language Resources and Evaluation, LREC 2004, Lisbon, Portugal, May 26-28, 2004*. ELRA, 2004.

[156] L. Li, D. Wang, Z. Zhang, and T. F. Zheng. "Deep Speaker Vectors for Semi Text-independent Speaker Verification". In: *CoRR* abs/1505.06427 (2015).

[157] Z. Zajic, M. Kunesova, and V. Radova. “Investigation of Segmentation in i-Vector Based Speaker Diarization of Telephone Speech”. In: *Speech and Computer - 18th International Conference, SPECOM 2016, Budapest, Hungary, August 23-27, 2016*. Springer, 2016, pp. 411–418.

[158] O. J. Rasanen, U. K. Laine, and T. Altosaar. “An Improved Speech Segmentation Quality Measure: the R-value”. In: *INTERSPEECH 2009, 10th Annual Conference of the International Speech Communication Association, Brighton, United Kingdom, September 6-10, 2009*. ISCA, 2009, pp. 1851–1854.

[159] A. H. Waibel, T. Hanazawa, G. E. Hinton, K. Shikano, and K. J. Lang. “Phoneme recognition using time-delay neural networks”. In: *IEEE Transactions Acoustics, Speech, and Signal Processing* 37.3 (1989), pp. 328–339.

[160] L. Mateju, P. Cerva, and J. Zdansky. “Investigation into the Use of Deep Neural Networks for LVCSR of Czech”. In: *2015 IEEE International Workshop of Electronics, Control, Measurement, Signals and Their Application to Mechatronics, ECMSM, 2015, Liberec, Czech Republic, June 22-24, 2015*. IEEE, 2015, pp. 184–187.

[161] R. Kneser and H. Ney. “Improved Backing-off for M-gram Language Modeling”. In: *1995 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1995, Detroit, Michigan, USA, May 8-12, 1995*. IEEE, 1995, pp. 181–184.

[162] F. Richardson, D. A. Reynolds, and N. Dehak. “A Unified Deep Neural Network for Speaker and Language Recognition”. In: *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 1146–1150.

[163] L. Ferrer, Y. Lei, M. McLaren, and N. Scheffer. “Study of Senone-Based Deep Neural Network Approaches for Spoken Language Recognition”. In: *IEEE/ACM Transactions Audio, Speech & Language Processing* 24.1 (2016), pp. 105–116.

[164] M. McLaren, L. Ferrer, and A. Lawson. “Exploring the Role of Phonetic Bottleneck Features for Speaker and Language Recognition”. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*. IEEE, 2016, pp. 5575–5579.

[165] L. Mateju, P. Cerva, J. Zdansky, and R. Safarik. “Using Deep Neural Networks for Identification of Slavic Languages from Acoustic Signal”. In: *INTERSPEECH 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, September 2-6, 2018*. ISCA, 2018, pp. 1803–1807.

Author's Publications

2019:

1. L. Mateju, Z. Callejas, D. Griol, J. M. Molina, and A. Sanchis. "An Empirical Assessment of Deep Learning Approaches to Task-Oriented Dialog Management". Accepted to: *Neurocomputing (Q1)*. 2019.

2. L. Mateju, P. Cerva, and J. Zdansky. "An Approach to Online Speaker Change Point Detection Using DNNs and WFSTs". In: *INTERSPEECH 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, September 15-19, 2019*. ISCA, 2019, pp. 649–653.

2018:

3. R. Safarik, L. Mateju, and L. Weingartova. "The Influence of Errors in Phonetic Annotations on Performance of Speech Recognition System". In: *Text, Speech, and Dialogue - 21st International Conference, TSD 2018, Brno, Czech Republic, September 11-14, 2018*. Springer, 2018, pp. 419–427.

4. L. Mateju, P. Cerva, J. Zdansky, and R. Safarik. "Using Deep Neural Networks for Identification of Slavic Languages from Acoustic Signal". In: *INTERSPEECH 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, September 2-6, 2018*. ISCA, 2018, pp. 1803–1807.

5. R. Safarik and L. Mateju. "Automatic Development of ASR System for an Under-Resourced Language". In: *41st International Conference on Telecommunications and Signal Processing, TSP 2018, Athens, Greece, July 4-6, 2018*. IEEE, 2018, pp. 100–103.

2017:

6. L. Mateju, P. Cerva, and J. Zdansky. "Investigation into the Use of WFSTs and DNNs for Speech Activity Detection in Broadcast Data Transcription". In: *E-Business and Telecommunications - 13th International Joint Conference, ICETE 2016, Lisbon, Portugal, July 26-28, 2016, Revised Selected Papers*. Springer, 2017, pp. 341–358.

7. R. Safarik and L. Mateju. "The Impact of Inaccurate Phonetic Annotations on Speech Recognition Performance". In: *Text, Speech, and Dialogue - 20th International Conference, TSD 2017, Prague, Czech Republic, August 27-31, 2017*. Springer, 2017, pp. 402–410.

8. L. Mateju, P. Cerva, J. Zdansky, and J. Malek. "Speech Activity Detection in Online Broadcast Transcription Using Deep Neural Networks and Weighted Finite State Transducers". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, Louisiana, USA, March 5-9, 2017*. IEEE, 2017, pp. 5460–5464.

2016:

9. M. Bohac, L. Mateju, M. Rott, and R. Safarik. "Automatic Syllabification and Syllable Timing of Automatically Recognized Speech - for Czech". In: *Text, Speech, and Dialogue - 19th International Conference, TSD 2016, Brno, Czech Republic, September 12-16, 2016*. Springer, 2016, pp. 540–547.

10. L. Mateju, P. Cerva, and J. Zdansky. "Study on the Use of Deep Neural Networks for Speech Activity Detection in Broadcast Recordings". In: *13th International Joint Conference on e-Business and Telecommunications (ICETE 2016) - Volume 5: SIGMAP, Lisbon, Portugal, July 26-28, 2016*. SciTePress, 2016, pp. 45–51.

11. R. Safarik and L. Mateju. "Impact of Phonetic Annotation Precision on Automatic Speech Recognition Systems". In: *39th International Conference on Telecommunications and Signal Processing, TSP 2016, Vienna, Austria, June 27-29, 2016*. IEEE, 2016, pp. 311–314.

2015:

12. L. Mateju, P. Cerva, and J. Zdansky. "Investigation into the Use of Deep Neural Networks for LVCSR of Czech". In: *2015 IEEE International Workshop of Electronics, Control, Measurement, Signals and Their Application to Mechatronics, ECMSM, 2015, Liberec, Czech Republic, June 22-24, 2015*. IEEE, 2015, pp. 184–187.

A Additional Tables

Table A.1: Influence of the number of epochs on the performance of SAD.

epochs	FER [%]	MR [%]	FAR [%]	F [%]	$\delta_{2/3}$ [s]
5	2.6	0.5	8.0	**54.9**	0.29
10	**2.4**	0.5	**7.2**	52.7	**0.26**
15	2.7	0.4	8.7	51.2	0.29
20	2.7	0.4	8.8	49.7	0.31
25	2.7	0.4	8.4	50.3	0.29
30	2.6	0.4	8.3	50.5	0.29
35	3.1	**0.2**	10.4	41.9	0.34
40	2.9	0.3	9.4	44.5	0.32
45	2.9	0.3	9.4	46.3	0.32
50	2.9	0.3	9.4	47.1	0.33

Table A.2: A detailed overview of recordings of QUT-NOISE-TIMIT corpus.

target environment	group	recordings	hours	change points	speech
low noise	A	4,000	100	94,054	45.6%
	B	4,000	100	95,342	46.6%
	A & B	8,000	200	189,396	46.1%
medium noise	A	4,000	100	93,462	45.2%
	B	4,000	100	95,990	46.4%
	A & B	8,000	200	189,452	45.8%
high noise	A	4,000	100	93,670	45.3%
	B	4,000	100	96,394	46.9%
	A & B	8,000	200	189,452	46.1%
all	A	12,000	300	281,186	45.4%
	B	12,000	300	287,726	46.6%
	A & B	24,000	600	568,912	46.0%

Table A.3: Extended results of the proposed speech activity detection approach in each scenario of QUT-NOISE-TIMIT corpus across all target environments.

target e.	scenario	HTER [%]	FER [%]	MR [%]	FAR [%]	F [%]	$\delta_{2/3}$ [s]
low	cafe	2.2	2.3	1.5	2.9	81.8	0.05
	car	**1.3**	**1.4**	**0.9**	1.7	**85.2**	**0.03**
	home	4.2	4.5	**0.9**	7.6	76.7	0.05
	reverb	4.0	3.9	5.6	2.4	78.3	0.07
	street	1.5	1.5	1.3	**1.6**	84.8	**0.03**
	all	2.6	2.7	2.0	3.2	71.3	0.05
medium	cafe	8.1	8.2	7.1	9.1	67.4	0.16
	car	**2.1**	**2.1**	**1.5**	2.6	**82.1**	**0.05**
	home	7.1	7.5	2.2	12.0	69.1	0.10
	reverb	8.8	8.3	15.0	2.7	70.1	0.16
	street	2.9	2.8	3.2	**2.5**	80.2	0.06
	all	5.8	5.8	5.8	5.8	61.4	0.11
high	cafe	28.2	27.8	33.7	22.7	42.5	0.39
	car	**5.7**	**5.6**	**6.4**	5.0	**72.7**	**0.16**
	home	12.9	13.2	8.8	16.9	59.4	0.21
	reverb	27.8	25.7	53.8	**1.8**	47.3	0.30
	street	10.4	9.8	17.0	3.8	68.1	0.17
	all	17.0	16.4	24.0	10.0	41.0	0.22
all	cafe	12.8	12.7	14.2	11.5	63.6	0.20
	car	**3.0**	**3.0**	**2.9**	3.1	**80.1**	**0.08**
	home	8.1	8.4	4.0	12.2	68.4	0.12
	reverb	13.5	12.6	24.7	**2.3**	65.6	0.17
	street	4.9	4.7	7.1	2.7	77.8	0.09
	all	8.5	8.3	10.6	6.3	58.0	0.12

Table A.4: Extended results of online performance of the proposed speech activity detection approach in each scenario of QUT-NOISE-TIMIT corpus across all target environments.

target environment	scenario	RTF	latency [s]
low noise	cafe	**0.02**	1.7
	car		**1.5**
	home		1.6
	reverb		1.7
	street		1.6
	all		1.6
medium noise	cafe	**0.02**	1.9
	car		**1.7**
	home		1.8
	reverb		2.0
	street		**1.7**
	all		1.8
high noise	cafe	**0.02**	2.2
	car		**1.9**
	home		2.0
	reverb		2.1
	street		**1.9**
	all		2.0
all	cafe	**0.02**	1.9
	car		**1.7**
	home		1.8
	reverb		2.0
	street		**1.7**
	all		1.8

Table A.5: Summarized results comparing the proposed SCP detection approach (both online and offline configurations) with the reference system on all of the COST278 languages.

approach	language	P [%]	R [%]	F [%]	$\delta_{2/3}[s]$	RTF	L [s]
LIUM		37.7	**70.4**	49.1	0.39	**0.016**	-
online	Basque	57.6	62.2	59.8	**0.17**	0.064	**3.0**
offline		**61.7**	67.4	**64.4**	0.23	0.079	4.8
LIUM		69.7	**76.5**	**72.9**	0.25	**0.017**	-
online	Belgian Dutch	60.8	45.1	51.8	**0.22**	0.065	**2.9**
offline		**71.3**	56.8	63.2	0.24	0.080	4.8
LIUM		**89.9**	**80.0**	**84.6**	**0.13**	**0.017**	-
online	Czech	86.9	74.9	80.5	**0.13**	0.065	**2.9**
offline		87.9	78.9	83.2	0.15	0.080	4.8
LIUM		51.4	**83.3**	63.6	0.31	**0.017**	-
online	Spanish	**66.2**	78.8	**72.0**	**0.16**	0.065	**3.0**
offline		65.6	75.0	70.0	0.25	0.080	4.7
LIUM		58.4	**86.1**	69.6	0.33	**0.016**	-
online	Galician	65.6	79.8	72.0	**0.23**	0.064	**3.0**
offline		**68.2**	83.0	**74.8**	0.30	0.079	4.7
LIUM		67.6	**88.7**	**76.7**	0.14	**0.017**	-
online	Greek	64.8	76.4	70.1	**0.12**	0.065	**3.0**
offline		**70.6**	79.3	74.7	0.16	0.080	4.7
LIUM		60.8	**88.5**	72.1	0.23	**0.017**	-
online	Croatian	**73.0**	81.8	**77.1**	**0.20**	0.065	**2.9**
offline		71.6	79.4	75.3	**0.20**	0.080	4.7
LIUM		68.2	**76.5**	**72.1**	**0.23**	**0.017**	-
online	Hungarian	66.0	65.3	65.6	0.25	0.065	**2.9**
offline		**70.8**	69.4	70.1	0.25	0.080	4.7
LIUM		66.1	**79.6**	**72.2**	0.22	**0.016**	-
online	Portuguese	70.2	63.4	66.7	**0.11**	0.064	**3.0**
offline		**75.0**	67.7	71.2	0.19	0.079	4.7
LIUM		71.8	**89.2**	79.5	0.25	**0.016**	-
online	Slovenian	84.2	87.9	**86.0**	**0.09**	0.064	**3.0**
offline		**84.7**	83.6	84.1	0.14	0.079	4.7
LIUM		69.3	**91.0**	78.7	0.13	**0.016**	-
online	Slovak	81.1	64.2	71.7	**0.12**	0.064	**2.9**
offline		**86.0**	73.1	**79.0**	0.13	0.079	4.7

Table A.6: Summarized results exploring the influence of different training data on the performance of the proposed SCP detection approach (tuned for online use) on all languages of COST278 database.

training data	language	P [%]	R [%]	F [%]	$\delta_{2/3}[s]$	RTF	L [s]
enhanced data		35.2	45.9	39.8	0.27	**0.064**	**2.9**
COST278 train	Basque	**57.6**	**62.2**	**59.8**	**0.17**	**0.064**	3.0
combined data		54.8	58.2	56.4	0.22	**0.064**	**2.9**
enhanced data		64.1	30.9	41.7	0.21	**0.065**	**2.9**
COST278 train	Belgian Dutch	60.8	**45.1**	51.8	0.22	**0.065**	**2.9**
combined data		**77.4**	44.4	**56.5**	**0.16**	**0.065**	**2.9**
enhanced data		82.7	**81.8**	**82.2**	0.17	**0.065**	**2.9**
COST278 train	Czech	86.9	74.9	80.5	0.13	**0.065**	**2.9**
combined data		**87.1**	76.3	81.3	**0.11**	**0.065**	3.0
enhanced data		40.2	56.1	46.8	0.34	0.065	**3.0**
COST278 train	Spanish	**66.2**	**78.8**	**72.0**	**0.16**	0.065	**3.0**
combined data		62.1	72.0	66.7	**0.16**	**0.064**	**3.0**
enhanced data		44.4	67.4	53.5	0.34	**0.063**	**2.9**
COST278 train	Galician	**65.6**	**79.8**	**72.0**	0.23	0.064	3.0
combined data		**65.6**	**79.8**	**72.0**	**0.22**	**0.063**	**2.9**
enhanced data		56.3	67.9	61.5	0.17	**0.063**	**2.9**
COST278 train	Greek	**64.8**	**76.4**	**70.1**	**0.12**	0.065	3.0
combined data		52.4	73.6	61.2	0.13	0.064	**2.9**
enhanced data		55.6	63.6	59.3	0.20	0.065	3.0
COST278 train	Croatian	73.0	**81.8**	**77.1**	0.20	0.065	**2.9**
combined data		**74.1**	78.2	76.1	**0.19**	**0.064**	3.0
enhanced data		41.2	57.1	47.9	0.22	0.065	**2.9**
COST278 train	Hungarian	**66.0**	**65.3**	**65.6**	0.25	0.065	**2.9**
combined data		47.5	59.2	52.7	**0.21**	**0.064**	3.0
enhanced data		58.8	53.8	56.2	0.21	**0.064**	**2.9**
COST278 train	Portuguese	**70.2**	**63.4**	**66.7**	**0.11**	**0.064**	3.0
combined data		68.8	59.1	63.6	0.13	**0.064**	**2.9**
enhanced data		74.2	61.0	67.0	0.16	**0.064**	**2.9**
COST278 train	Slovenian	**84.2**	**87.9**	**86.0**	**0.09**	**0.064**	3.0
combined data		80.0	77.9	79.0	0.12	**0.064**	**2.9**
enhanced data		68.0	79.1	73.1	**0.07**	**0.064**	**2.9**
COST278 train	Slovak	**81.1**	64.2	71.7	0.12	**0.064**	**2.9**
combined data		79.7	**82.1**	**80.9**	0.08	**0.064**	**2.9**