Zürcher Hochschule
für Angewandte Wissenschaften

zh
aw School of Engineering

Neural-network Based Speaker Diarization on Federal Assembly Sessions

Master's thesis to obtain the

Master of Advanced Studies in Data Science

ZHAW – School of Engineering

Supervisors
Mark Cieliebak
Thilo Stadelmann

Submission
October 23, 2020

Serge Brun
T WB-T-MAS-DASC-20-5

Mattackerstrasse 81
CH-8052 Zurich
+41 76 398 02 23
brunser1@students.zhaw.ch

Management Summary

Speaker diarization is the process of partitioning an audio stream with voices of multiple people into equal segments associated with each individual speaker. It has emerged as an important part of speech recognition systems by solving the problem of *who spoke when*. Speaker diarization has found application in scenarios ranging from broadcast news to business meetings up to parliamentary sessions and many more. It has seen rapid progress over the past decades, fuelled by the increasing demand for the applications it enables.

The thesis begins with an introduction of the first artificial neural network dating back to 1958, describing its evolution and development up to the present state of the art in deep learning. Then several experiments on speech systems belonging to SwissTXT are conducted, preceded by a comprehensive introduction into speaker diarization and subtitling. An important aspect of the thesis is to compare and assess these systems in order to understand how they support the process of speech transcription and subtitling of broadcast media. Thereby video recordings of Swiss parliamentary sessions have been used to evaluate speaker diarization and speaker change detection performance, providing a means to foster technical assistance for the working area of subtitling and respeaking.

Finally, a summary of the contributions is given concluding with an outlook to short- and long-term future work in the field of speech recognition.

Acknowledgements

This research work was carried out at the University of Applied Sciences Zurich (Switzerland).

I would hereby like to express my sincere gratitude to my supervisors Prof. Mark Cieliebak and Prof. Thilo Stadelmann who introduced me to speech technology and provided enormous support and guidance during my studies.

I wish also to thank Prof. Kurt Stockinger for the admission of this thesis to the consecutive master programme in Data Science.

Many thanks also go to my colleague at work, Tobias Matzat for his help and support in getting along with the European Media Laboratory (EML) Transcription Platform.

Finally, I am especially grateful to my friends and family who made this work possible.

Zurich, October 23, 2020
Serge Brun

Table of Contents

1. Speaker Diarization

1.1 Introduction

Speaker Diarization is the process of partitioning an audio stream into homogeneous segments and assigning them to the respective source without any prior knowledge of the speakers involved nor their exact number. These sources generally include speaker, music or background noise which is sometimes referred to as the "cocktail party problem" [1].

By trying to solve the problem of *who spoke when*, Speaker diarization has found application in many use cases related to audio and/or video document processing. This includes broadcast news, radio- and television programmes or conference recordings. For such scenarios, it can be helpful to automatically detect the intervals of speech activity and the number of involved speakers for further processing and evaluation.

Well-known examples of speaker diarization applications include speech and speaker indexing, speaker recognition (in the presence of multiple or competing speakers), video captioning, and many more. More formally, the process of speaker diarization can be considered as higher-level inference of audio data.

Initiated originally within the telephony domain, and later in broadcast news, speaker diarization has nowadays its main application in the field of conference meetings, media broadcast or parliamentary sessions. Speaker diarization has thus become an increasingly important area of speech processing technology [2].

Thanks to neural networks, in particular deep learning, the performance of state of the art speaker diarization systems has improved tremendously in recent years. The neural-based approaches have shown much better performance than other statistical methods [3]. This fact encouraged the original intention to write this thesis which was to measure the accuracy of detected speaker changes in recordings of parliamentary debates. The idea is to show if speech systems can already achieve human-level precision. Hence, the following chapters seek to describe and evaluate speaker diarization systems with the aid of statistical evaluation. The remainder of the thesis is organized as follows.

The section *Speaker Diarization* outlines the basic functionality of speaker diarization.

The section *Subtitling and Respeaking* describes the task of subtitling as seen in radio and broadcast programs.

The *Experiments* section elaborates on the experimental setup used to obtain the results based on the NIST Rich Transcription evaluations on meeting data [4], [5].

The final section concludes with a synopsis of recent voice research and an outlook to prospective areas for future speech investigation.

1.2 Definition

Diarization can be defined as the assignment of certain labels to temporal annotation regions of audio and video recordings. The labels can represent many audio types such as the background environment (silence, music, noise, lipsmack, etc.), the speaker's identity and gender as well as other characteristics present in the signal [6].
Speaker diarization is different when compared to other audio recognition systems, i.e. speaker identification and speech recognition. Speaker identification is the technique to find the speaker identity (who is speaking) and speech recognition strives to obtain the words spoken by the speaker (what spoken). The main goal of speaker diarization is however, to solve the problem of *who spoke when* by extracting segments of speech from an audio stream and associate them with the correct speaker [7]. It can also be seen as the combination of speaker segmentation and speaker clustering techniques. Speaker diarization enables to use the voice of a speaker to handle access to virtual assistants, verify customer identity in call centers or authenticate security control for confidential information areas [8].
Lastly, speaker diarization is the capability of a software or hardware to receive speech signal, identify the speaker present in the speech signal and assign it to people engaged in a dialogue [9].

1.3 History

The first model of a "Perceptron", a type of artificial neural network, yielding promising results in the ability to learn behavior seen in real neurons, was developed in the late 1950's by psychologist Frank Rosenblatt [10].
The book *Perceptrons: an introduction into computational geometry* released 1969 by Minksy and Papert, illuminated the limitations of the perceptron such as the lack of the exclusive-OR (XOR) function. Work on neural-networks was therefore said to be dissipated during the 70's and early 80's, leading to the so called AI winter when research efforts were largely concentrated on symbolic systems [10], [11].
In the mid 1980's a Neural Net Resurgence became apparent due to new discoveries in the field of machine learning. For example the backpropagation algorithm was first popularized in 1986 for training a three-layer neural network. Simply put, backpropagation computes how network parameters affect the final output by moving layer by layer from the final to the first one and adjusting the derivatives based on a loss function to a desired output. Because backpropagation did not generalize well to train

networks with more layers at that time, machine-learning moved again to other research fields such as graphical models and kernel methods [10].
From 2010 onward the so called deep learning revolution has taken off. Improved methods developed for training deep neural works, benefitting from cheap, powerful graphic card processors, have achieved results that hitherto have been unseen. Since then particular successes have been made with convolutional neural nets (CNN) for computer vision, with recurrent neural nets (RNN) for machine translation and speech recognition, followed by deep reinforcement learning (RL) for game playing and robotics. This progress promoted the use of deeper networks with more layers which led to the notorious vanishing or exploding gradient problem. The dilemma with backpropagated neural networks is that with more layers, the gradients of the loss function multiply at each layer by very small numbers which results in an exponential decline of long-term information, thus complicating the training of deep neural networks (DNN) over numerous time steps [10], [12]. In theory RNNs can make use of an information sequence of arbitrarily length but in practice they are limited to looking back only a few steps.
This shortcoming led to the rise of the *LSTM* (long-short term memory) network, introduced way back in 1997, by Juergen Schmidhuber and Sepp Hochreiter. In their paper "Long Short-Term Memory", a novel recurrent network architecture is presented that revolutionized speech recognition until only recently. The LSTM is specifically designed to overcome the above mentioned error back-flow problems and to retain state information over longer periods of time [13]. It has seen successful application in speech recognition for language and acoustic modeling, in sequence labeling such as part-of-speech tagging (POS) and named entity recognition (NER) as well in neural machine translation and video captioning [10].
In this context the *gated recurrent unit* (GRU) is to be listed as it combines and facilitates the inner working of the LSTM network, resulting in fewer parameters and faster training.
As an extension to traditional LSTMs, *bidirectional LSTMs (*BLSTM) are basically two independent RNNs stacked together in bidirectional order. This construction allows the network to have both backward and forward information about the sequence at every time step. The learning algorithm is fed with a given sequence of the input data from beginning to end and on a reversed copy from end to beginning. This can provide additional context and result in faster and even more thorough learning on the problem [14].
Transformers are somehow the new kid on the block when compared to other neural networks. The turning point was in 2017 when the transformer network was introduced in Google's *Attention is all you need* paper [15]. It has since then truly changed the way of working with text data. The Transformer in natural language processing is a novel ar-

chitecture that aims to solve sequence-to-sequence (seq2seq) tasks while handling long-range dependencies with ease [16].
It is based solely on attention mechanism which is the idea to handle dependencies between input and output with attention, omitting RNNs or convolution entirely. This is due to the fact that RNNs do not work well for longer sentences and cannot be parallelized because of their sequential nature. This extends training time and makes it difficult to run inference. While the RNN encoder-decoder generates the output sequence one element at a time, the attention mechanism passes all of them at once. The attention mechanism can be described as the weighted sum of the hidden states that are passed as the context vector from the encoder to the decoder. It allows the model to look at the other words in the input sequence to get a better understanding of a certain word in the sequence [17].
Among the many approaches and advances that have emerged since then, a major improvement in natural language processing is BERT [18]. Based on the Transformer architecture and an effective method to model language, it can be trained on large unlabeled text corpora, in an unsupervised or semi-supervised manner [19]. Recent work on pre-training Transformer with large scale corpus have demonstrated that BERT is capable of learning high-quality semantic representation [20]. It has reached new state of the art results on eleven natural language processing tasks, pushing the boundaries of what has only been seen in computer vision so far, referring to ImageNet classification and Generative Adversarial Networks [21], [22]. As a result, the pre-trained BERT model can be used for a wide spectrum of tasks ranging from question answering to sentiment analysis and hopefully soon for speaker diarization.

1.4 Essentials

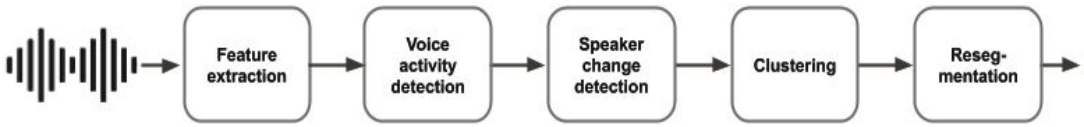

Figure 1: Speaker diarization pipeline

Typically, speaker diarization systems are built upon a combination of modular units as depicted in the figure above. This chapter reviews the mechanisms that are related to the speaker diarization task, starting with an introduction into feature extraction. Thereupon, the splitting of speech regions into equivalent segments is explained which is known as speaker change detection or speaker segmentation. The grouping of speakers according to their identity covers the embedding and clustering part. Resegmentation finally tries to refine the segments with the previously recognized speaker features [3].

1.4.1 Feature Extraction

Feature extraction is a method to remove non-speech regions like silence, music and background noise from audio data and is known as voice activity detection.
It is a dimensionality reduction process to detect and annotate audio features like energy, spectrum divergence between speech and background noise or pitch estimation.
The recognized features are converted into a sequence of acoustic feature vectors and grouped into speaker-homogeneous segments. Therefore each extracted feature should include the unbiased speaker characteristics in order to enable the system to properly identify and assign the individual speakers. An optimal feature extractor is expected to maintain both high inter-speaker and low intra-speaker discrimination at the same time [2], [3].
A commonly used feature extraction algorithm for speech signals are Mel-Frequency Cepstral Coefficients (MFCC). They were popularized in the 1980's by Damis and Mermelstein and have since then continued to be the state of art for automatic speech and speaker recognition [23]. MFCC computations are modeled after the human hearing system using the ear's working principle as foundation to retain phonetically vital properties of the speech signal. These typically include the human's ear critical frequency bandwiths which are linearly pitched below 1kHz and logarithmically above 1kHz using the Mel frequency scale [24]. MFCC have application in speech recognition systems, for example to distinguish numbers spoken into a telephone [25], for multimodal user identification in biometric identity management, as well as in music information retrieval for genre classification [26]. The computation of MFCCs is a rather complicated process that can be reduced to six simplified steps.

1. Split the signal into short frames of 20-40ms.
2. For each frame calculate the periodogram estimate of the power spectrum via Fast Fourier Transform (FFT).
3. Apply the mel filterbank to the power spectra and sum up the energy of each filter.
4. Take the logarithm of the periodogram values to get the filterbank energies.
5. Take the DCT (discrete cosine transform) of the log filterbank energies.
6. Keep DCT coefficients and retrieve the resulting list of Mel Frequency Cepstral Coefficients

However, with the advent of deep learning in speech systems, it has become questionable whether MFCCs are still the right choice given that deep neural networks are less susceptible to highly correlated input. The standard way of doing automatic speech recognition using the Gaussian mixture model and hidden Markov model with mel-frequency

cepstral coefficients required the decorrelation of filter bank coefficients with discrete cosine transformation. With deep neural networks this step has lost its significance and is no longer necessary.

1.4.2 Voice Activity Detection

Voice activity detection (VAD) is considered as the binary task of distinguishing speech from silence or non-speech segments in spoken utterances of a given audio stream or recording.
Accurate speech activity detection is significant as wrongly detected non-speech-segments as speech or vice versa may deteriorate the speaker model performance, leading to a decreased overall performance of the system. Consequently, a broad range of pattern recognition techniques have been adopted, varying from support vector machines to multilayer neural networks [27].
Voice activity detection is usually measured against prediction scores that are greater than a predefined threshold θ_{VAD}, eventually yielding the segments marked as speech [27], [28].

1.4.3 Speaker Change Detection

Speaker change detection (SCD) is an important part of a speaker diarization system and is sometimes referred to as speaker segmentation.

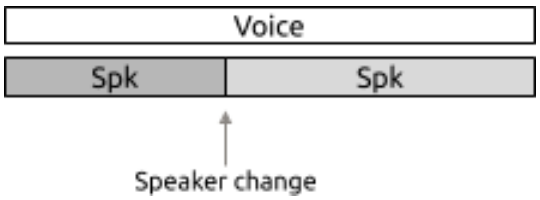

Figure 2: Speaker change detection

In the case of a detected speaker change, as seen in the picture above, the voice segment is subsequently split to contain a single speaker.
SCD aims to detect speaker change points in a given audio recording and is usually evaluated with prediction scores like precision and recall against a tunable threshold θ_{SCD} once the feature vectors from the voice activity detection have been extracted.
On a more general level, speaker change detection seeks to find the times when a change in the acoustics of a recording is recognized. It can detect boundaries within a speaker turn even when the background conditions change. Examples include speech/non-speech, music, background noise and other sounds.
Speaker diarization systems can obtain an output by means of first performing a speaker segmentation and then grouping respectively clustering the segments belonging to the same speaker. Among the many types of speaker segmentation methods in research literature, the following ones are looked at in greater detail.

Segmentation and classification

In segmentation and classification segmentation is achieved in two steps where the segment boundaries are first detected and then classified. This approach uses a distance-based method which is typically the Bayesian information criterion (BIC) [29]. From two given windows of audio stream data the algorithm computes three models $G_1(\mu_1,\sum_1)$, $G_2(\mu_2,\sum_2)$ and the combined $G(\mu,\sum)$. The ΔBIC distance metric is defined as

$$\Delta BIC = BIC\{G_1\} + BIC\{G_2\} - BIC\{G\} \tag{1}$$

It can be applied to generate language-dependent sequences or breakpoints of speaker turns and was first presented by Chen and Gopalakrishnan in 1998 [30]. Optimization efforts to speed up the processing of BIC have shown that its computation is more intensive than other statistic-based metrics when used with signals of high resolution. However, it appears the good overall performance has kept it as the algorithm of choice in many modern applications.

Metrics such as the generalized likelihood ratio (GLR), use a preset threshold to decide whether both segments belong to the same speaker or otherwise. As a simplified version of the Bayesian information criterion, GLR finds the distance between two windows of an audio stream using the aforementioned three Gaussian models. The GLR distance is, given two windows (w) of audio data $G_1(\mu_1,\sum_1)$ and $G_2(\mu_2,\sum_2)$, defined as

$$GLR = w\left(2\log\left|\sum\right| - \log\left|\sum\nolimits_1\right| - \log\left|\sum\nolimits_2\right|\right) \tag{2}$$

It has been tested with real-time audio and speaker segmentation, as well with transcription and indexing of audio data.

Another often cited algorithm in the context of segmentation research is the Kullback-Leibler divergence (KL) introduced by Solomon Kullback and Richard Leibler in 1951. It compares two distributions (P and Q) to measure the distance of its probabilities where one distribution typically represents empirical observations (P) and the other one (Q) a probability model or an approximation. It is often denoted as

$$D(P\|Q) = KL(P,Q) = \sum_{x\in X} P(x)\cdot\log\frac{P(x)}{Q(x)} \tag{3}$$

with x as the possible outcome respectively the speaker change point. The Kullback-Leibler divergence algorithm has proven to compute fast while still delivering acceptable results [31].

Segmentation by classification

This approach puts audio segments of fixed length consecutively together, enabling the classifier to produce labels as a sequence of decisions. Notable examples used for speaker segmentation are the Gaussian mixture model (GMM) with entropy classification, support vector machines (SVM) to separate speech from non-speech in audio stream-

ing, decision trees (DT) with the same objective of audio separation as the former and the factor analysis (FA) technique. They have been successfully adapted to the broadcast domain reporting relevant results.

Multi-pass segmentation

Another way to differentiate speaker segmentation found in older research literature [32] is the distinction of systems that perform a single processing pass of acoustic data to determine the speaker change-points versus systems that perform multiple passes to achieve optimal segmentation by refining the decision of change-point detection on successive iterations [33]. This second class of systems include two-pass algorithms where the first pass suggests various change-points. Actually, more than there are, creating temporarily a high false alarm error rate which is then reevaluated respectively discarded in the second pass.

Many of the algorithms applied to find change-points including the ones reviewed can either work alone or in a two-step system together with another technique. Being known as computationally expensive, BIC-based approaches are sometimes only applied in a second pass while the first pass uses another statistic-based distance method [32], [33].

Neural segmentation

Neural networks have, due to advances in hardware and greater availability of data, seen an exponential growth in recent years. Since 2010 they are increasingly applied to speech technologies including varied work on audio segmentation [34], shifting the focus of scientific research to acoustic modelling with neural network-based representations. Various efforts and progresses have been made with feed-forward networks, multilayer perceptrons, CNNs, and RNNs. For instance, convolutional neural networks commonly related to image recognition and image classification applications, are also being applied to audio segmentation and classification. These implementations usually rely on time-frequency representations of the audio signal that are treated as channels, in analogy with image processing systems [35].

Recurrent neural networks are often found in tasks that deal with temporal sequences of information because they are considerably more favourable to model temporal dependencies than ones without a feedback-loop between input and output.

The long short-term memory (LSTM) network is a special kind of RNN introducing the concept of a memory cell. This cell uses controlled states referred to as gated state or memory, enabling them to learn, retain and forget information in long dependencies. This capability has significantly improved machine translation with large numbers of docu-

ments and the processing of multilingual speech recognition tasks, making LSTM networks a powerful overall tool to carry out long and short-term sequence labeling simultaneously [36].
With regard to broadcast audio transcription the complexity of the diverse speech and audio signals, audio segmentation is still a challenging problem. As an essential module of a broadcast audio transcription system, it has greatly benefitted from the advancements of deep learning. However, the need of large amounts of labeled training data quickly becomes a bottleneck of deep learning-based audio segmentation methods. To tackle this problem, an adapted segmentation method is proposed to select speech/non-speech segments with high confidence from unlabeled training data as a complement to the labeled training data. The method relies on GMM-based speech/non-speech models trained on an utterance-by-utterance basis. The long-term information is used to choose reliable training data from the utterances at hand. Audited results show that this data selection method is a powerful audio segmentation algorithm of its own. Besides it has been observed that newer deep neural networks such as BERT are trained on unlabeled data and thereby superior to those trained with data chosen by two comparing methods. Moreover, better performance could be observed by combining the deep learning-based audio segmentation methods with ones using metric-based data selection [37].

Overlapped speech detection

Overlapped speech detection is the task of detecting regions where at least two speakers are speaking at the same time. Thereby long audio sequences are split into short fixed-length, adding the additional benefit of higher correlation amongst the detected audio fragments. Because the input sequences are overlapping, each frame can have multiple averaged candidate scores to produce the final frame-level score [38].

1.4.4 Clustering

Clustering is in some scenarios a prerequisite for speaker diarization. It is usually distinguished between offline and online clustering.

Offline Clustering

Offline clustering is often associated with hierarchical clustering approaches such as bottom-up and top-down clustering followed by more recent techniques like x-vectors.
Agglomerative or bottom-up clustering is a method that starts with many small clusters and merges them to get a bigger cluster [39]. It is based on unsupervised learning and attempts to find clusters in unlabeled data. It is different from the k-means algorithm in that it does not

require any prior knowledge of the number of clusters, a measure of similarity is sufficient [40].
Bottom-up clustering has been used for many years in pattern classification as seen in [40] and is generally computed using a matrix distance between the fitted clusters. The closest pair is then iteratively merged until a distance measure exceeds a predefined threshold.
Top-down or divisive clustering on the other hand starts with one or very few clusters. To obtain the optimum amount of clusters they are recursively split until each speaker point is in its own cluster.
In practice, top-down clustering is known as less popular than agglomerative clustering even though it remains unclear which one achieves better results under what circumstances.
The output of hierarchical clustering in the form of a dendrogram is meant to help visualizing the history of groupings and figuring out the optimal number of clusters. This is achieved by building binary trees of the data that iteratively merge similar clusters (agglomerative method) or split dissimilar groups (divisive method) [39]. In both procedures a distance metric such as the Euclidean Distance, the Kullback-Leibler divergence (KL2) or the Bayesian information criterion is used to determine acoustic similarity. Since these statistic-based methods are known to not work properly with speaker clustering, other attempts have been made such as using factor analysis, i- or d-vectors in order to overcome these limitations. The problem is however, that they depend on rather long adjacent sliding windows (two seconds or more) and therefore appear to be biased towards fast speaker interactions [3].
The strengths of hierarchical agglomerative clustering (HAC) are that there are no assumptions to be made up front of the number of clusters, making the HAC algorithm substantially different from k-means. As the name suggest the output is ranked and thereby offering more information than unstructured data. Known weaknesses are the immutability once a cluster assignment has been performed, the limitation to small datasets because it requires quadratic computation time [39], [40] and its sensitivity to statistical outliers such as background and speech noise.
Another widely used algorithm in speaker diarization is spectral clustering. It has been adopted to various domains, including speaker diarization due to its proven theoretical guarantee and lower computational complexity [42]. It surpasses k-means performance when the number of speaker is unknown beforehand, but fails to adapt to real audio recordings with overlapping segments.
Over the recent years, neural embeddings have been successfully applied to speaker recognition and verification tasks, outperforming what has previously been considered as state of art in speaker modelling. As pointed out in [38], today's best performing speaker diarization systems rely on x-vectors as clustering input. Hereby, probabilistic linear

discriminant analysis (PLDA) is applied to fixed-length sliding windows in order to obtain the pairwise similarity (affinity) matrix [41] with the resulting embeddings called x-vectors. These embeddings are used in the task of speaker diarization, to learn not only the audio embeddings, but also the required scoring function at the same time [43].
Clustering-based methods have a number of problems. First, they cannot be optimized to minimize diarization errors directly, because the clustering procedure is a type of unsupervised learning method. Secondly, they have trouble in handling speaker overlaps because the clustering algorithm implicitly assumes one speaker per segment. In other words, clustering-based methods fail to adapt to real audio recordings because the speaker embedding model has only been optimized with single-speaker non-overlapping segments. These problems hinder speaker diarization applications from working on real world audio recordings which are likely to contain overlapping segments.
As proposed in [17], a viable solution is the use of end-to-end neural clustering (EEND). Different from most of the proposed methods, it does not rely on clustering anymore. Instead, a self-attention based neural network simultaneously outputs the joint speech activities for all speakers at each time frame.

Online Clustering

Online clustering is supposed to process data in real-time. In other words, the system has only access to data recorded up to the current time. Once a segment is available it is compared with all existing clusters. If the minimum similarity is smaller than a given threshold, the speaker label is immediately emitted as a new cluster. Otherwise the segment is added to the most similar existing cluster [44]. Online clustering algorithms are known to achieve inferior results in real-time diarization due to the lack of contextual information available in the offline setting. Moreover, a final resegmentation step can only be applied in the offline setting. Nonetheless, the choice between online and offline depends primarily on the context of the application i.e. the domain the system is intended to be deployed. For instance, latency sensitive applications such as live video or speech analysis typically restrict the system to online clustering algorithms.
An alternative approach to online clustering is to restart clustering each time a new audio segment comes in. It has been discarded as this would be computationally too expensive, and also bring an issue of temporal discontinuity because the labels obtained from the current clustering step and the previous runs may not be related anymore. To overcome this obstacle, Zhu et al. propose the use of a greedy algorithm, where the clustering is run only once after a warm-up period, and then only the existing clusters are updated. However, the greedy

algorithm is significantly less accurate than re-clustering and is also sensitive to initial clustering conditions [45].
Another solution proposed in [46] is the reconciliation algorithm. It tries to solve two problems of online processing which include the varying number of speakers over time, i.e. when a speaker comes in or another leaves and the tradeoff between latency and accuracy of the results. It compares the sequences of labels obtained in previous and current cluster sets on the same portion of the audio, and examines all possible permutations of the current labels, then selects the permutation with the lowest Hamming distance between both sequences of labels. In other words, it permutes the current labels to make it similar to the previous ones. To reduce the computational complexity, [46] proposes to use active windowing to limit the history to the N latest segments.
The most promising approach to online clustering uses an unbounded interleaved-state recurrent neural network (UIS-RNN) [45]. Hereby the clustering step is treated as an online generative process of an entire utterance (X, Y), where X is an extracted observation sequence of embeddings and Y is the sequence of ground-truth speaker labels. Each speaker is modeled by a parameter-sharing instance of RNN, while the RNN states for different speakers are interleaved in the time domain.
To accommodate for an unknown number of speakers, the RNN is modeled by the distance-dependent Chinese restaurant process (ddCRP). The system is fully supervised and requires time-stamped speaker labels for training. The system achieved a diarization error rate of 7.6%, outperforming the state of the art spectral offline clustering algorithm on the NIST SRE 2000 CALLHOME benchmark, practically establishing an online diarization solution with offline quality.

1.4.5 Resegmentation

Similar to spech activity and speaker change detection, resegmentation can be seen as a sequence labeling task.
Resegmentation is usually achieved in a combination of GMM clustering and Viterbi decoding algorithm, using categorical cross entropy as loss function and softmax for the activation function of the output layer. This procedure intends to refine the audio segmentation boundaries from the clustering step and is done to prevent of sudden changes in segmentation labels.
Other resegmentation systems rely on hidden Markov models (HMM) to produce the final hypothesis [29].

1.4 Evaluation

Speaker diarization systems are usually evaluated using the diarization error rate (DER) which is defined as the the sum of errors received from multiple sources. Speaker change detection systems on the other hand, are traditionally evaluated with recall and precision [3]. Since the outcome of voice activity detection is directly related to speaker change detection, the following section introduces its integral functionality.

1.4.1 Voice Activity Detection

The two main components of voice activity detection (VAD) are false alarm error (E_{FA}) and miss detection error (E_{MD}). They are also used to measure the diarization error rate.
E_{FA} is the ratio of scored time that hypothesized speech is labeled as non-speech in the reference. S_{Hyp} and S_{Ref} indicate the hypothesized andreference speech part, and $|S_{Hyp} - S_{Ref}|$ denotes the duration of hypothesized speech not contained in reference speech.

$$E_{FA} = \frac{|S_{Hyp} - S_{Ref}|}{T_{total}} \tag{4}$$

E_{MD} inversely denotes the percentage of time that a hypothesized non-speech segment corresponds to a reference speech part.

$$E_{MD} = \frac{|S_{Ref} - S_{Hyp}|}{T_{total}} \tag{5}$$

The detection error (E_D) for VAD is the sum of E_{FA} and E_{MD}.

$$E_D = E_{FA} + E_{MD} \tag{6}$$

1.4.2 Speaker Change Detection

Since one of the main objectives of the thesis is to evaluate the ability of a speech system to accurately predict speaker changes in a live or real-time audio stream, the estimation of speaker change detection (SCD) has an enabling role in the results analysis of the performed experiments.
The evaluation results have been received using precision and recall which are defined as follows

$$Precision = \frac{TP}{(TP + FP)} \tag{7}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{8}$$

Speaker change detection results can be viewed as hypothesized change points that are correctly detected within the allowance of a reference interval. This produces sequences of 0 and 1 with 1 being the

change point or segment boundary. If a hypothesis change point is not found in the reference, it is considered as a false alarm (F_A) change point or false positive (FP).
If a reference change point is not detected by a model, it is a miss detection (M_D) point or false negative (FN).
For example, in a system for speaker change detection, of 12 reference speaker changes (relevant items) 8 changes were detected by the system (retrieved instances). Of 8 detected changes, 5 are actually real speaker turns yielding a precision of 5/8 and a recall of 5/12.
In recall and precision evaluation a hypothesized change point is counted as correct if it is within the temporal neighborhood of a reference change point. Both values are very sensitive to the actual size of this maximum distance between two segment boundaries (aka. tolerance) and quickly reach zero as the tolerance decreases. It also means that recall and precision are sensitive to the actual temporal precision of human annotators.
The two dual metrics purity and coverage on the other hand, do not depend on a tolerance parameter [5]. Introduced by pyannote.metrics, a toolkit for reproducible evaluation, this metric pair allows additional insight on the behavior of the underlying system therefore making them more relevant in the perspective of speaker diarization.
Purity and coverage were originally introduced to measure cluster quality but can also be adapted to speaker change point detection.
Given R the set of reference speech turns, and H the set of hypothesized segments, coverage is defined as follows

$$Coverage = \frac{\sum_{r \in R} \max_{h \in H} |r_i \cap h_j|}{\sum_{r \in R} |r_i|} \tag{9}$$

where $|r_i|$ is the duration of the reference segment and $|r_i \cap h_j|$ is the intersection of segments r_i and h_j.
Coverage is computed for each reference segment as the ratio of the maximum intersection duration with the most adjacent hypothesis segment and the duration of the reference segment.

Purity is the dual or inverse metric that indicates how *pure* hypothesis segments are. It is defined as follows

$$Purity = \frac{\sum_{h \in H} \max_{r \in R} |h_i \cap r_j|}{\sum_{h \in H} |h_i|} \tag{10}$$

where $|h_i|$ is the duration of the hypothesized segment and r_j the duration of the reference segment. To measure assignment correctness it is necessary to know which hypothesis segment h_i maps to which ground truth classification r_j. Purity is then calculated as the ratio of the maximum intersection of the hypothesized segment with the reference segment and

the duration of the hypothesized segment. In case of a segmentation, too many speakers (over segmentation) would for example result in high purity and low coverage, whereas missing out on many speaker change points will decrease purity.

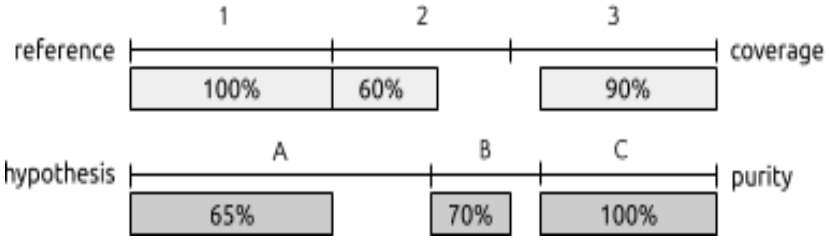

Figure 3: Segmentation example

In the above picture reference segment 1 is fully covered (100%) by the hypothesis segment A whereas hypothesis segment A is only 65% covered i.e. pure by reference segment 1 and 35% by reference segment 2. The reference segment 2 gets 60% coverage from hypothesis segment A and 40% from segment B applying equally for hypothesis segment B (70% and 30%) and C.
It then becomes clear that the intersection duration is obtained from the h_i and r_j mapping of the most adjacent hypothesis segment with the longest intersection.

1.4.3 Diarization Error Rate

The main metric used to evaluate speaker diarization as described and used by NIST in the rich transcription (RT) evaluations is the diarization error rate (DER) [47]. It is defined as the fraction of time that is not correctly attributed to a speaker or non-speech region and is computed as follows

$$DER = \frac{\sum_{s=1}^{S} dur(s) \cdot (max(N_{ref}(s), N_{hyp}(s)) - N_{correct}(s))}{T_{score}} \tag{11}$$

where T_{score} is the total scoring time in the denominator.

$$T_{score} = \sum_{s=1}^{S} dur(s) \cdot N_{ref}$$

In more simplistic terms the diarization error rate is defined as the sum of three sources of error.

$$DER = E_{FA} + E_{MD} + E_{Speaker} \tag{12}$$

$E_{Speaker}$ denotes the percentage of speech assigned to the wrong speaker while E_{FA} und E_{MD} point to the aforementioned false alarm and missed speech metrics.
Speaker error can be divided into incorrectly assigned speakers and speaker overlap. In the case of incorrectly assigned speakers the hypothesized speaker is not fitting with the reference (ground-truth) speaker. Conversely, speaker overlap error refers to the case when the incorrect

number of speakers speaking at the same time is assumed. The inclusion of overlapping speech error in the evaluation was restricted to a contrastive metric in the initial RT evaluations but since 2006 DER is the measure of choice. Overlap errors can be further classified as missed overlap (when fewer speakers than the real number are hypothesized) and false alarm overlap (when too many speakers are hypothesized) [5].
It is to be taken into account that DER is time-weighted and therefore attributes little attention to speakers whose overall speaking time is short. Usually a non-scoring collar of 250ms is applied at either side of a reference segment boundary to account for inevitable inconsistencies in the annotated start and end point labels.
The DER error can be decomposed into errors coming from different sources [32]. They are different from VAD and SCD metrics in that they do not account for overlap speech.
Speaker Error is defined as

$$E_{Spkr} = \frac{\sum_{s=1}^{S} dur(s) \cdot (min(N_{ref}(s), N_{hyp}(s)) - N_{correct}(s))}{T_{score}} \tag{13}$$

False alarm speech has the following notation

$$E_{FA} = \frac{\sum_{s=1}^{S} (N_{hyp}(s) - N_{ref}(s)) dur(s) \cdot (N_{hyp}(s) - N_{ref}(s))}{T_{score}} \tag{14}$$

and is only computed for segments where the reference speech is labeled as non-speech.

Missed speech is calculated for segments where the hypothesis segment is labeled as non-speech.

$$E_{MD} = \frac{\sum_{s=1}^{S} (N_{ref}(s) - N_{hyp}(s)) dur(s) \cdot (N_{ref}(s) - N_{hyp}(s))}{T_{score}} \tag{15}$$

For speech recognition API's like IBM Watson and Google Speech, a diarization error rate of 25% can be considered as about average for regular speech recognition. But in cases of more specific data, such as parliamentary session recordings, it becomes less likely to achieve such values, quickly reaching the 50% mark as seen in the experiments section.

2. Subtitling and Respeaking

2.1 Outline

An important reason to write this thesis stems from the fact that an essential part of SwissTXT's workforce is operating in the market for subtitling and respeaking. Thereby a software application is used to connect and coordinate the numerous language processing workflows. As the demand for subtitling has quickly grown over the years, the path of manual subtitling is no longer feasible, due to increasing production costs and reduced transcription times.
Assisted Subtitling is a technique that has seen great progress in backing respeaking processes during the last decade. In more technical terms assisted subtitling refers to the integration of automatic speech recognition (ASR) to automatically generate speech transcripts from different media sources, linked to the idea of supporting respeakers in their work.
The section will further report on the basics and origins of subtitling and respeaking, starting with an introduction to the task of subtitling.

2.2 Subtitling

Subtitling is the process of synchronously displaying audio transcriptions on a television, video screen or any other display device to provide additional or interpretive information [48]. It has been around since the early 20th century and has traditionally served as a means to make TV programmes and movies accessible to a broader audience.
As a result BBC came up with Ceefax in the 1970's, commonly known as teletext in many other european countries. It offers the possibilty to display subtitles when desired, creating so-called closed subtitling, a type of decodable and optional subtitling. It is different from open subtitling which the viewer cannot switch off [48].
It was quickly recognized that subtitling provided valuable benefit to meet the demands of media content for the deaf or hard of hearing, known by the acronym SDH (subtitles for the deaf and hard of hearing). Hereby, the term "caption" is used to refer to the method of transferring aural information into text for people with hearing impairment. In contrast to subtitles, captioning does not only contain dialog, but sound effects, onomatopoeias and other visual cues. For the remainder of the paragraph subtitling is referred to as a generic term for subtitle and caption, as they are generally considered equivalent.
The traditional task of subtitling is a process based on the manual production of time-aligned transcriptions of audiovisual content which is known to require considerable effort. It has been disclosed that the production of high-quality subtitles increases processing time of the

original content by a factor between eight and ten. Such circumstances prevented subtitlers to keep up with the increasing number of live tv-shows and dialogues. In 2001 BBC has therefore introduced the use of speech technology for live subtitling which has become the standard until today.

2.3 Live Subtitling

Live subtitling is different from pre-preparable subtitling only in using a pre-recorded media type but has obvious consequences for subtitling in practice.
In live subtitling a differentiation is usually made between block and scrolling/snake mode, referring to the way how subtitles are presented. Snake mode is associated with (near-)verbatim or word for word subtitling whereas edited subtitling relates to block or paraphrased mode.
A typical misconception about real-time subtitling and the use of speech recognition is that spoken television dialogues are input directly into a speech recognizer, which automatically transfers this speech into a subtitling program. This kind of speaker-independent speech recognition system, in which the software has not been trained to recognize the voice of a specific individual, is not yet accurate enough for subtitling purposes (Robson 2004).
Moreover, natural speech, redundant in nature, is not very suitable for direct use in subtitles without at least some degree of adaptation. Stop words, hesitations, false starts and the like, are not desirable in subtitling. Instead, a speaker-dependent system is used and actively trained by a respeaker talking repeatedly into the software to build up a personal lexicon. A well known ASR engine used also at SwissTXT is Nuance's Dragon NaturallySpeaking.

2.4 Towards Respeaking

In an average live subtitling session, one person watches and listens to the television programme as it is broadcast live. Wearing a headset, she or he simultaneously repeats or paraphrases what is being said, an act that is known as respeaking [48]. The respeaker speaks directly to a speech recognizer, which results into a concept subtitle. In case of errors made by the respeaker or the speech recognizer the subtitle is first corrected before put on air.
There are three major failure points in live subtitling referred to as error, delay and reduction. The occurrence of delay and errors in subtitles can be assigned to the speech system whereas the rendering is in greater part controlled by the respeaker. Although the written version of the spoken comment is nearly always a reduced form, this does not inevitably mean that the quality of the text is affected. Uncompleted thoughts, interjections, and repetitions are just a few examples of sen-

tence parts that carry no essential information, and can therefore be deleted or summarized. A different word choice or emphasis, however, can quickly result in a change of meaning. In other words, text reduction can be problematic as well. There are two common types of distinction in text reduction: partial and total reduction. Partial reduction is achieved through condensation, a more concise rendering of the source text. Total reduction is achieved through deletion or omission. Very often both processes are combined, leading to the rewriting that is so typical of subtitling.
During live subtitling, respeakers are required to simultaneously perform two major tasks, namely comprehending the audio comments and producing subtitles. Composing a subtitle thus requires shifting between – and concurrently activating – various memory processes, which puts a great strain on the working memory. It is therefore known that the average time of respeaking lies between 15 and 20 minutes.
It should now be obvious that some form of text reduction is required as respeakers have to decide what and how they are going to reduce the subtitle, reproducing only the essence of the information presented in the spoken comment. Respeakers use a wide variety of strategies to reduce the subtitles in such a way that the loss of information is limited as much as possible.
The analyses referred in [48] shed more light on the effect that quantitative reduction may have on comprehension, and show the delicate relation between the (reduced) subtitles and the original content. The descriptives show that it is almost impossible to subtitle literally when live subtitling. On average, less than half of the source text was subtitled by the respeakers even if told to do so.
The quantitative analysis, which allowed to explore the causes of text reduction, showed that reduction is not a random operation [48]. Text reduction is largely determined by a number of external factors. With regard to the reduction strategies, the results suggest that respeakers prefer to omit a comment rather than reducing it. Thus, the delay of the speech recognizer may be an important decisional factor for the choice between complete and partial reductions.

2.5 Evaluation

A common metric used for public televison broadcasts in several european countries like Italy and Switzerland is the NER model. It is used to determine the accuracy of live subtitles in television broadcasts and events that are produced using speech recognition.

$$NERvalue = \frac{N - E - R}{N} * 100 \tag{15}$$

The acronyms refer to N as the total number of words in the live subtitles, E as edition error and R as recognition error.

As opposed to the below described word error rate (WER), NER is a static model simply measuring the discrepancy between written and spoken text [49].

$$WER = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C} \tag{16}$$

WER is used to compare the accuracy of transcripts produced by speech recognition software where S is the number of substitutions, *D* number of deletions, *I* the number of insertions, *C the* number of correct words and *N* the number of words in the reference [50].
WER works on the word instead of the phoneme level and is derived from the Levenshtein distance which is a metric to measure the difference between two sequences. It is a valuable tool for comparing different systems as well as for evaluating improvements within one system but fails to include information about the nature of translation errors.

F-Measure
The F-score also known as F_1-score is the harmonic mean or weighted average of precision and recall and is calculated as follows:

$$F_1\text{-}score = \frac{2 * Recall * Precision}{Recall + Precision} \tag{17}$$

It takes true positives (TP) and false negatives (FN) into account, making it useful for unbalanced data distributions.
The F-score has also been applied in the context of natural language processing where a score closer to 1 designates better recall and precision of the translation.
Another method for evaluating the quality of machine translated text from natural/spoken language is BLEU. The central idea behind the BLEU (bilingual evaluation understudy) algorithm is to evaluate the quality of text which has been translated by a machine and that of a human. A number of 1 is considered a perfect output to the reference text which is even rarely attained by human translators [51].
The US National Institute of Standards and Technology (NIST) uses a BLEU based metric to assess the quality of machine-translated text by additionally applying a kind of inverse frequency for terms i.e. n-grams found in the document. Other well-known metrics are ROUGE, METEOR or LEPOR [52].

3. Experiments

3.1 Objectives

The main goal of this work was to explore and evaluate the prospectives of speaker change detection in a long running process of live video transmission, decoded in real-time by an end-to-end system, i.e. deep neural network (DNN). The basic question is whether today's speech systems achieve human-level accuracy in identifying speaker turns.
The experimental setup is based on recorded sessions of the Swiss parliament respectively of the National Council and the Council of State spanning from 2014 to 2019.

3.2 Datasets

Datasets suitable for speaker diarization tasks require to contain speech with multiple speakers. This can be, for example, spontaneous dialogue speech because one of the main challenges of speaker diarization is to determine the total number of speakers for each utterance. The following examples are some of the commonly used datasets for speaker diarization.

NIST Speaker Recognition Language: multiple Pricing: $2400.00 Transcripts: yes	NIST Rich Transcription Evaluation Language: en, ar, cn Pricing: $2000.00 Transcripts: yes, w/ audio
The ICSI Meeting Corpus Language: en Pricing: free Transcripts: yes, w/ audio	The AMI Meeting Corpus Language: multiple Pricing: $2400.00 Transcripts: yes
VoxCeleb Large scale audio-visual dataset of human speech Language: en Pricing: free Transcripts: unoffical	ETAPE Evaluation en Traitement Automatique de la Parole Language: fr Pricing: $5000.00 Transcripts: yes
CALLHOME American English Speech Language: en Pricing: $2500.00 Transcripts: yes	

Table 1: Diarization datasets

Augmentation noise sources are artificially noised examples used to harden speech models against background noise. Known examples are Musan, a corpus of music, speech, and noise recordings and Audio Set, a large-scale dataset of manually annotated audio events.

As indicated most of the datasets are associated with high aquisition costs, likewise valid for academic licenses, thus much work has gone into the location and assembling of freely available resources. German corpora such as VoxForge are available but lacking properly time-aligned transcripts, they failed to render assistance for the intended purpose. Namely, to use pyannote.audio to train a german based model for speaker change detection which could then be properly applied to the preferred diarization task.
The datasets used in the experiments were obtained from the website *parlament.ch* where the meetings and debates of the Swiss National Council and Council of States are listed and provided as video download. Consequently, the parlamentarians speak in one of the respective national languages french, italian, german or romansh. The type of data has been selected as it contains multiple speakers and characteristics matching more closely the aim of the thesis.
So far the evaluations on parliamentary debates have been made on recordings of the sessions

No.	Name
1	14.3668_36072 Motion «Wasserzinsregelung nach 2019»
2	17.479_47823 Initiative «Mehrwertsteuerpflicht generell ab 150 000 Franken Umsatz»
3	18.3170_48535 Motion «Asyl-Querulanten wirksam disziplinieren»
4	19.3759_48515 Postulat «Konsumkreditgesetz. Digital taugliche Formerfordernisse»
5	19.3960_48126 Motion «Gesetzliche Grundlage für die Bekanntgabe von Daten an die privaten Krankenversicherungseinrichtungen»

Table 2: Datasets used in the experiments

The data has been carefully selected to only contain german speakers but given the nature of Swiss parliamentary sessions, were nowhere to be found. The comprehensive search was also undertaken because the language can be passed as a parameter with the function call to improve transcription accuracy of the respective speech engine.

No.	Filename	Length	Size	Format
1	14.3668_36072	07:58	60.7 MB	mp3, mp4
2	17.479_47823.mp3 17.479_47823.mp4	13:44	108 MB	mp3, mp4
3	18.3170_48535	10:54	86.1 MB	mp3, mp4
4	19.3759_48515.mp3 19.3759_48515.mp4	12:11	95.6 MB	mp3, mp4
5	19.3960_48126	10:13	81.2 MB	mp3, mp4, wav, mov

Table 3: Properties of evaluated datasets

As seen in the above table the files are made available in mpeg-4 format. IBM Watson for instance, accepts only audio formats such as mp3 or flac to be uploaded. Therefore the data had first been converted according to the requirements of each speech system. For this task, the multimedia framework *ffmpeg* was of great benefit by converting the files into the required format.
Thereafter the resulting files were uploaded to each speech service using the cURL command line library or a file upload interface. The results were returned either in JSON (JavaScript Object Notation) format or in the case of EML as an XML document (Extensible Markup Language).
Note that the last file (5) was transcribed in two different file formats. One being wav and the other mov. Although the downloaded file did not show any signs of damage, verified by different software video applications such as Adobe Premiere and media savvy people, the poor score as shown below might be a result of the fact that the available debates are pre-processed i.e. reduced in quality and shortened in duration before being made public.
Due to the fact that no time-aligned reference annotations of the accessed datasets exist, a semi-manual annotation method has been applied using the IBM Watson transcriptions as a reference. The manually annotated speaker turns were compared with the time-stamped values of the speech engine output in order to correct and refine the manual reference annotations. The alignment was at best in the tenths of a second and served solely for the rectification of manual annotation inaccuracy.
Before evaluating the datasets with the scoring tools dscore [53] and pyannote.metrics, the results files from the transcription services had to be converted into the RTTM format according to the characteristic document structure of the respective speech system. The lack of available conversion scripts imposed quite a constraint for further processing which eventually resulted in writing own conversion implementations for each service. The reason for this procedure lies in the fact that dscore supports the calculation of different metric only when the reference and the hypothesis file are saved in the rich transcription time marked (RTTM) format [54]. RTTM is defined as follows

Type	Segment type, here SPEAKER
File ID	The waveform filename of the recording
Channel ID	The waveform channel Should always be 1
Beginning time	Onset of turn in seconds from beginning of recording
Duration	Duration of turn in seconds

Orthography Field	Should always be <NA>
Speaker Type	Should always be <NA>
Speaker Name	Name of speaker of turn Should be unique within scope of each file
Confidence Score	System confidence (probability) that information is correct Should always be <NA>
Signal Lookahead Time	Should always be <NA>

Table 4: RTTM document outline

Rich transcription time marked (RTTM) files are space-delimited text files containing one turn per line, each line contained of ten fields.

1	2	3	4	5	6	7	8	9	10

A typical sample RTTM file for speaker detection could look like this.

```
SPEAKER id 1  0.43  2.35 <NA> <NA> jay <NA> <NA>
SPEAKER id 1 07.21 19.11 <NA> <NA> lea <NA> <NA>
SPEAKER id 1 28.82  8.35 <NA> <NA> max <NA> <NA>
SPEAKER id 1 37.25  0.88 <NA> <NA> jay <NA> <NA>
SPEAKER id 1 42.43 15.05 <NA> <NA> sue <NA> <NA>
...
```

3.3 Speech Systems

Due to the constrained accessibility of the systems in question, their description is limited to publicly available information and user guides.

3.3.1 IBM Watson

Watson Speech-to-Text is a cloud-native solution that uses deep-learning AI algorithms to apply knowledge about grammar, language structure, and audio/voice signal composition to create customizable speech recognition for optimal text transcription [55]. It enables to get started quickly providing tutorials and examples, as well as 500 free minutes for transcription.

3.3.2 Speechmatics

Speechmatics is a highly scalable speech recognition engine using machine learning to transcribe and convert speech in any context [56]. According to the company's website they use cutting edge voice-to-text technology to achieve the most accurate transcription of any real-time or recorded media. With regard to usability, Speechmatics provides the possibilty to chose between a command line tool and a user-friendly inferface to initialize the transcription process.

3.3.3 EML

The EML Transcription Platform is a framework to automatically transform spoken language from various media sources (TV news, podcasts, lectures, etc.) into text. It can be configured to recognize different speakers and mark the points where a speaker turn has happened [57].

Architecture

The EML platform architecture consists of five different modules that are exposed as web services. Besides the core transcription technology and platform, speech components are required for each language. In order to reduce the needed skill level to develop speech components, the EML Transcription Platform provides the framework for such tools.

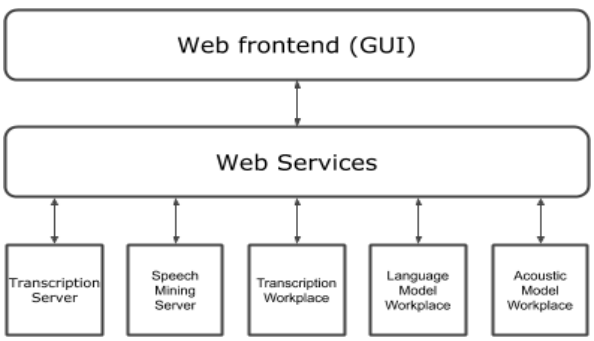

Figure 4: EML - Overall architecture

The remainder of this section focusses on some of the key aspects of the platform and is mainly intended for the technically interested reader.
The EML transcription server provides the core functionality for transcription. It is a highly scalable application running in a JEE application server (e.g. Apache Geronimo, JBoss Wildfly).
It handles the automatic deployment of audio files to the registered transcription decoders and the logging of the transcription job status. It is accessed through an XML-over-HTTP interface.
The EML transcription decoder is a highly efficient two-pass decoder with built-in capabilities for audio segmentation, speaker clustering (diarization) and online speaker adaptation. The transcription decoder implements sophisticated algorithms and techniques for the automatic conversion of audio into text. The recognition decoder is independent of the chosen language and the application scenario.
Figure 5 shows the typical components of the transcription decoder.

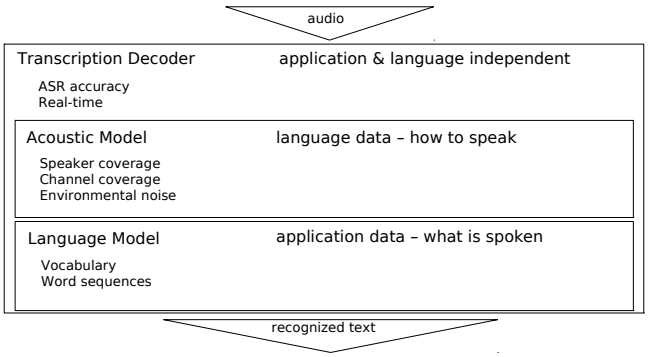

Figure 5: Transcription decoder components

In order to recognise speech the decoder needs a language component consisting of an acoustic model and a language model.
The acoustic model provides statistics on the spoken sounds of a certain language (e.g. german) – in other words, it models «how one should speak». The model is based on large amounts of manually transcribed spoken data, thus ensuring accurate, speaker-independent recognition under varying channel conditions or environmental noise. The acoustic model is largely application-independent and customising it for an application typically provides further recognition improvements but the customisation process is quite demanding and complex. It would typically be undertaken at long intervals and requires the availability of large amounts of training data.
The language model covers recognisable words and contains statistics from typical conversations in the application domain – it models «what is spoken». Accordingly, best results are achieved by deriving this information from real-life recorded utterances.
All transcription systems use hidden Markov model techniques for acoustic modelling. These statistical models are based on a multiplicity

of transcribed audio data. The performance is enhanced as more data is added to model utterances recorded in the respective applications. Transcription is by default done in real-time but can be adjusted to quality requirements and off-peak hours. There is no time limit for audio files with a minimum length of 200ms. Experiments have shown that the decoder is capable of transcribing audio signals of more than 240 minutes length in broadcast and university lectures scenarios.
Streamed decoding is a feature where the audio stream is directly dispatched to an idle transcription server which in return immediately starts with the decoding of the audio. Unfortunately this component has not yet been made available but would have been optimal for the intended use of real-time speaker change detection.

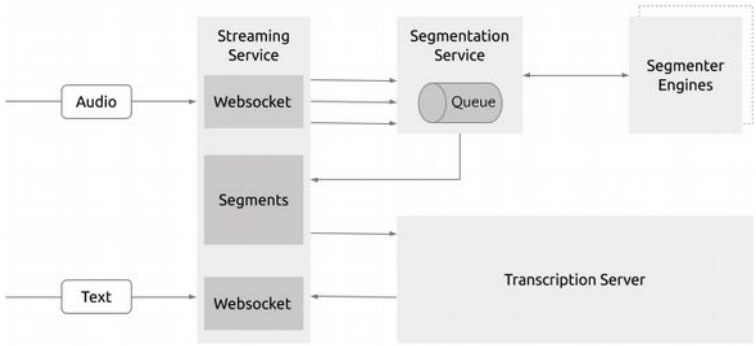

Figure 6: Use case for online diarization

The EML speech mining server provides a means of analysing the recognition results obtained by the EML transcription server. The architecture allows combining several analysis modules that enrich (annotate) the xml-encoded recognition result. The analysis modules can classify xml documents according to predefined keywords or keyword classes or – in a purely data driven way – according to the terms present in the documents. In addition to adding annotation information to individual documents, there are also modules available which aggregate this information and create trend statistics. This information can be used to alert users upon predefined keywords or topics. A quick interface is also provided for browsing and searching the large amounts of transcribed audio documents. The speech-to-text application may forward the transcribed result from the EML transcription server to the EML speech mining server where the XML documents are put into a Java messaging (JMS) queue. The analysis manager fetches these documents from the queue and puts them into a pipeline of individual analysis modules where they are annotated, indexed and statistically evaluated from metadata such as customer mood, customer requests, requested products, frequent terms and matched keywords.
The EML transcription workplace serves the purpose of the development and improvement of a speech component for a specific language. This requires a set of (manually) transcribed audio data. It provides a

web-based environment for manual transcription and correction of audio data. It is designed and implemented to facilitate the overall workflow of audio data collection, correction and pre-processing in order to minimise human transcription work, assure transcription quality, simplify data management, and to safeguard data security.
The EML language model workplace is available via a web interface and enables the user to develop a language model (LM) from scratch or to customize a given base LM by uploading text corpora or using data from the web. It aims to facilitate the build, customisation and improvement of language models through automation of the workflow and continuous model enhancement, supported by the ability to visualize and monitor the quality of the obtained values.

3.4 Results

The following section intends to give further insight and a more in-depth understanding of the results obtained during the process of data exploration and experimental design.
Some of the caveats encountered while conducting the experiments were the limited number of free minutes restricted by the respective speech service providers, the accessibility of quality parliamentary video content and the missing reference transcriptions of the relevant corpora. As a consequence the experiments were conducted on toy data, leading to a possibly inferior performance than otherwise. It can be assumed that the proposed systems will show better results on more high-quality data.

3.4.1 Speaker Diarization

The scores of the speaker diarization experiments have been achieved with the evaluation tool dscore whereas the ones of the speaker change detection have been evaluated with pyannote.metrics. dscore computes the diarization error rate using the NIST md-eval.pl tool without any default collar around speaker turns and a default frame step of 10ms explicitly including regions that contain overlapping speech in the reference diarization. This behavior can be altered using the collar, step and ignore_overlaps flags. All other metrics, not shown here, are computed off of frame-level labelings generated from the reference and system speakers.
The best diarization error rate results have been achieved by the EML speech engine with the exception of debate 5 as shown below.

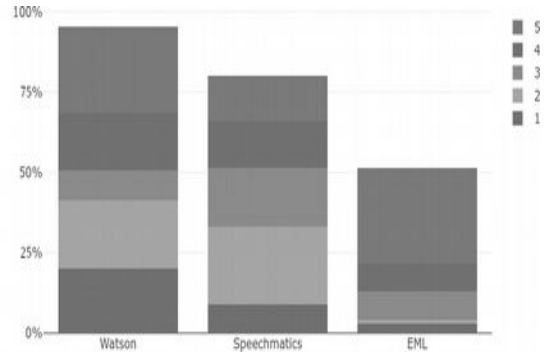

Figure 7: Diarization error rate

	Watson	Speechmatics	EML
1	40.34	18.07	5.65
2	42.41	48.20	2.40
3	18.90	36.58	18.01
4	35.09	28.87	17.17
5	54.36	28.31	59.32

Table 5: Diarization error rate

3.4.2 Speaker Change Detection

The speaker change detection results have been obtained using the pyannote.metrics open-source toolkit for speaker diarization.The diagram shows the averaged precision and recall scores computed from the reference and system annotations of the evaluated debates.

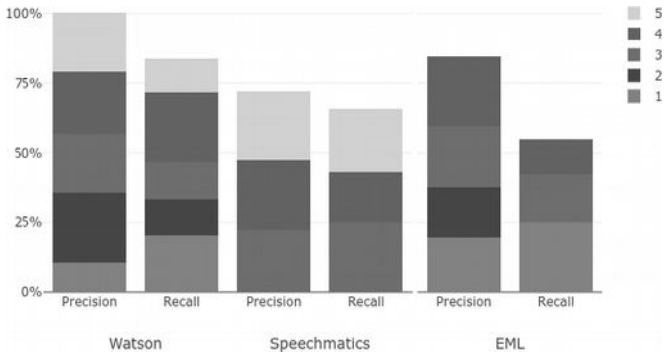

Figure 8: Recall and precision

	Watson		Speechmatics		EML	
	Precision	Recall	Precision	Recall	Precision	Recall
1	0.21	0.50	0.0	0.0	0.11	0.16
2	0.50	0.33	0.0	0.0	0.10	0.0
3	0.43	0.33	0.13	0.11	0.13	0.11
4	0.44	0.62	0.14	0.08	0.14	0.08
5	0.43	0.30	0.14	0.10	0.0	0.0

Table 6: Recall and precision

The following chart and table show the grouped coverage and purity values of the speech systems in consideration.

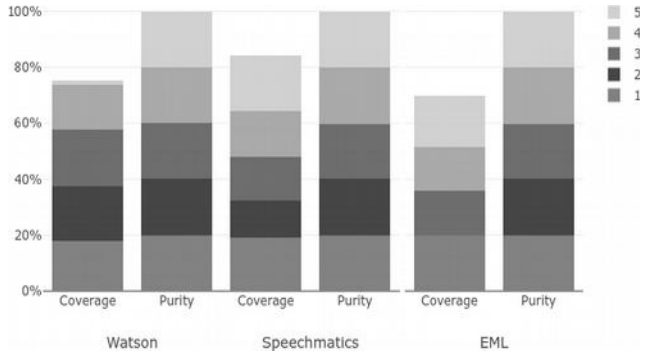

Figure 9: Coverage and purity

	Watson		Speechmatics		EML	
	Coverage	Purity	Coverage	Purity	Coverage	Purity
1	0.84	1.0	0.93	1.0	0.90	1.0
2	0.93	1.0	0.64	1.0	0.81	1.0
3	0.94	1.0	0.76	0.99	0.71	0.99
4	0.75	1.0	0.79	1.0	0.70	1.0
5	0.88	1.0	0.97	1.0	0.82	1.0

Table 7: Coverage and purity

The results suggest that the speech engines tend to oversegment speaker changes. In particular purity with an output value of almost 100% appears to confirm this assumption which has been further fortified by visualized speaker changes of the viewed parliamentary session samples.

3.4.3. Synopsis

In this thesis, the proposed approach to measure and evaluate speaker change detection in parliamentary sessions has been considered.
The results from the experiments disclose the Watson speech engine as the winner of the speaker change detection evaluation and EML as the system with the best scores for speaker diarization, in all but one file evaluations. The efforts to figure out the cause of the transcription score of debate 5 have so far failed.
The results indicate that a limited experimental setup already provides an opportunity to assess a professional speech recognition system. Speaker change detection and speaker diarization can therefore be evaluated on a real basis but on the premise of better quality of data and a less restricted access to the underlying system. A desirable option for a reproducible test procedure would imply fewer manual steps to avoid errors and to simplify the assessment of the measured values.
The experiments have shown that under the given assumptions, current speech systems do not reach human-level accuracy in detecting speaker changes and are therefore not yet capable to fully replace human interaction. Still, the results give a hint at the work of respeakers in the sense of how and where their daily work will carry on to continue.

4. Conclusion and Perspectives

In this thesis, different speech recognition system have been compared, especially with regard to the task of speaker change detection.
First, an overview of the topic has been given detailing the various aspects that led to today's speaker diarization systems. The intention of the subtitling and respeaking section was to raise awareness and understanding in how these professions are affected by the technical development. Finally, the experiments chapter presented the obtained results from the speech system evaluation at SwissTXT. It has been shown that speaker change detection is not yet accurately enough to fully substitute for human intervention.
Speaker recognition challenges such as real-time or online diarization, need to be addressed as a starting point for further development. Audio and video recordings of real environments with multiple speakers and overlapping speech will supposedly continue to be the sticking point of any future speaker diarization. Furthermore, broadcast audio transcription will remain a challenging problem due to its high variability of speech and audio signal sources. However, the last years have seen, apart from dedicated speech recognition tools, a rapid development of neural network-based applications entering the market, not least triggered by a substantial progress in deep learning. It has made significant impact on speech technology and is continuing to hold its promise of being a revolutionary technology. Recently, transformer networks with self-attention mechanism have been successfully applied to common natural language processing tasks such as machine translation. These networks could be for example further used to address the problem of *who spoke when* in speaker diarization.
So far the effects of the deep learning revolution have created a variety of application scenarios ranging from driverless cars to automated trading on the global stock markets. As a consequence an increased interest in real-world implementations of speech recognition systems has evolved, opening the field for tomorrow's achievements and inventions.

References

[1] A. Ephrat,I. Mosseri, O. Lang et al., Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation, arXiv:1804.03619 [cs.SD], Apr. 2018

[2] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, Speaker diarization: A review of recent research, IEEE Trans. on ASLP, vol. 20, no. 2, pp. 356–370, 2012.

[3] R. Yin, Steps towards end-to-neural-speaker-diarization, Artificial Intelligence [cs.AI], University Paris-Saclay, 2019, Accessed on: April 13, 2020. [Online]. Available: https://tel.archives-ouvertes.fr/tel-02332204/document

[4] NIST National Institute of Standards and Technology Accessed on: June 18, 2020. [Online]. Available: https://en.wikipedia.org/wiki/National_Institute_of_Standards_and_Technology

[5] "pyannote.metrics", A toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems, Accessed on June 4, 2020. [Online]. Available: https://pyannote.github.io/pyannote-metrics/reference.html

[6] K. Dabbai, S. Hajji, A. Cherif, Real-Time Implementation of Speaker Diarization System on Raspberry PI3 Using TLBO Clustering Algorithm, Circuits, Systems, and Signal Processing 39, 4094–4109 (2020). Accessed on: Mar. 27, 2020. [Online]. Available: https://link.springer.com/article/10.1007/s00034-020-01357-2

[7] H. Sulistyanto, Speaker-Diarization: Its Developments, Application, And-Challenges, University of Wollongong, Australia, 2011. Accessed on: Mai 2, 2020. [Online]. Available: http://eprints.undip.ac.id/36153

[8] Minh N. Do, An automatic Speaker Recognition System, Swiss Federal Institute of Technology, Lausanne, Switzerland, 2001. Accessed on: Mai 2, 2020. [Online]. Available: http://read.pudn.com/downloads60/sourcecode/multimedia/audio/209082/asr_project.pdf

[9] P. K. Kurzekar, R. R. Deshmukh, V. B. Waghmare, P. P. Shrishrimal, A comparative study of feature extraction techniques for speech recognition system, International Journal of Innovative Research in Science, Engineering and Technology, 2014. Accessed on: June 7, 2020. [Online]. Available: http://www.ijirset.com/upload/2014/december/34_A_Comparative.pdf

[10] R. J. Mooney, The-Deep-Learning-Revolution, 2017. Accessed on June 6, 2020. [Online]. Available: https://slideplayer.com/slide/17790687

[11] "Perceptrons (book)", Accessed on June 10, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Perceptrons_(book)

[12] C. F. Wang, The Vanishing Gradient Problem, Jan, 8, 2019. Accessed on June 12, 2020. [Online]. Available: https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484

[13] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Nov. 1997. Accessed on June 12, 2020. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[14] J. Brownlee, How to Develop a Bidirectional LSTM For Sequence Classification in Python with Keras, June 16, 2017. Accessed on June 20, 2020. [Online]. Available: https://machinelearningmastery.com/develop-bidirectional-lstm-sequence-classification-python-keras

[15] A. Vaswani, N. Shazeer et al., Attention is all you need, Dec. 6, 2017. Accessed on Apr. 1, 2020. arXiv:1706.03762 [cs.CL]

[16] P. Joshi, How do Transformers Work in NLP? A Guide to the Latest State-of-the-Art Models, Accessed on May 8, 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/06/understanding-transformers-nlp-state-of-the-art-models

[17] Y. Fujita, N. Kanda, Y. Xue et al., End-to-End Neural Speaker Diarization with Self-Attention, arXiv:1909.06247 [eess.AS], Sep. 13, 2019

[18] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv:1810.04805v2 [cs.CL], Oct. 11, 2018

[19] O. Petrova, Natural Language Processing: the age of Transformers, Aug. 22, 2019. Accessed on: June 2, 2020. [Online]. Available: https://blog.scaleway.com/2019/building-a-machine-reading-comprehension-system-using-the-latest-advances-in-deep-learning-for-nlp

[20] Y. Zihao et al., Transformer tutorial, 2018. Accessed on: Sept 10, 2020. [Online]. Available: https://docs.dgl.ai/en/0.4.x/tutorials/models/4_old_wines/7_transformer.html

[21] M. Rizvi, Demystifying BERT: A Comprehensive Guide to the Groundbreaking NLP Framework, Sep. 25, 2019. Accessed on: June 12, 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework

[22] E. Culurciello, The fall of RNN / LSTM, Apr. 13, 2018. Accessed on: May 24, 2020. [Online]. Available: https://towardsdatascience.com/the-fall-of-rnn-lstm-2d1594c74ce0

[22] Sabur Ajibola Alim and Nahrul Khair Alang Rashid, Some Commonly Used Speech Feature Extraction Algorithms, Dec. 12, 2018. Accessed on: May 24, 2020. [Online]. Available: https://bit.ly/3c99i7w

[23] "Mel Frequency Cepstral Coefficient (MFCC) tutorial", Accessed on: May 24, 2020. [Online]. Available: http://www.practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs

[24] S. Chakroborty, A. Roy,G. Saha, "Fusion of a complementary feature set with MFCC for improved closed set text-independent speaker identification". IEEE International Conference on Industrial Technology, 2006. ICIT 2006. pp. 387-390

[25] "Mel-frequency cepstrum", Accessed on June 1, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Mel-frequency_cepstrum#Applications

[26] S. Modi, Biometrics in Identity Management: Concepts to Applications, Artech House, 2011, p. 112

[27] A. Sholokov, Improving-ML-for-Speaker-Recognition Segmentation, 2018, p.46. Accessed on: May 30, 2020. [Online]. Available: https://epublications.uef.fi/pub/urn_isbn_978-952-61-2977-8/index_en.html

[28] H. Bredin, R. Yin, J. Manuel Coria, pyannote.audio: neural building blocks for speaker diarization, arXiv: 1911.01255 [eess.AS], Nov. 4, 2019

[29] P. Gimeno, I. Vinals, A. Ortega et al., Multiclass audio segmentation based on recurrent neural networks for broadcast domain data, Mar. 5, 2020. Accessed on: June 8, 2020. [Online]. Available: https://link.springer.com/article/10.1186/s13636-020-00172-6

[30] D. Wang, R. Vogt, M. Mason and S. Sridharan, Automatic audio segmentation using the Generalized Likelihood Ratio, *2008 2nd International Conference on Signal Processing and Communication Systems*, Gold Coast, QLD, 2008, pp. 1-5, doi: 10.1109/ICSPCS.2008.4813705.

[31] "Kullback-Leibler divergence", Accessed on June 1, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Kullback-Leibler_divergence

[32] X. Miro, Robust Speaker Diarization for Meetings, Barcelona, Oct. 2006. Accessed on June 2, 2020. [Online]. Available: http://www.xavieranguera.com/phdthesis/node11.html

[33] M. Huijbregts, D. V. Leeuwen, F. D. Jong, Speech overlap detection in a two-pass speaker diarization system, In Interspeech, pp. 1063-1066, 2009.

[34] X. Zhao, C. Wang, X. Xu, Sub-voice Detection and Recognition based on Hybrid Audio Segmentation and Deep Learning, Sep. 2019. Accessed on: June 2, 2020. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3366194.3366219

[35] D. Doukhan, J. Carrive, Investigating the use of semi-supervised convolutional neural network models for speech/music classification and segmentation, 9th International Conferences on Advances in Multimedia (MMEDIA), 2017.

[36] F. Eyben, F. Weninger, S. Squartini, B. Schuller, Real-life voice activity detection with LSTM recurrent neural networks and an application to Hollywood movies, IEEE International Conference O nAcoustics, Speech and Signal Processing (ICASSP)., 2013, pp. 483–487. Accessed on: May. 9, 2020. [Online]. Available: https://doi.org/10.1109/icassp.2013.6637694.

[37] X. Yang, D. Qu, W. Zhang et al., An adapted data selection for deep learning-based audio segmentation in multi-genre broadcast channel, Oct. 2018. Accessed on June 4, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1051200418300745

[38] R. Yin, H. Bredin, C. Barras, Speaker-change-detection-in-BroadcastTV-using-BiLSTM, Jan. 23, 2018. Interspeech 2017, Stockholm, Schweden. Accessed on June 4, 2020. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01690244/document

[39] C. Marklin, Hierarchical Agglomerative Clustering, Dec. 31, 2018. Accessed on June 21, 2020. [Online]. Available: https://bit.ly/3iVA9qt

[40] Nanni, Mirco. 2005. Speeding-Up Hierarchical Agglomerative Clustering in Presence of Expensive Metrics. 378-387. 10.1007/11430919_45.

[41] Ruiqing Yin, Herve Bredin, and Claude Barras, Neural Speech Turn Segmentation and Affinity Propagation for Speaker Diarization, in Proc. Interspeech 2018, 2018, pp. 1393–1397.

[42] A. Y. Ng, M. I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, Advances in neural information processing systems, pp. 849-856, 2002.

[43] N. Flemotomos, Spectral Clustering for Speaker Diarization, Dec. 3, 2018. [Online]. Available: https://nikosfl.github.io/work/classes/projects/proj_nf_ee546.pdf

[44] Q. Wang C. Downey L. Wan et al., Speaker Diarization with LSTM, Dec. 14, 2018. Accessed on June 1, 2020, arXiv:1710.10468 [eess.AS]

[45] A. Zhang, Q Wang, Z. Zhu et al., Fully Supervised Speaker Diarization, Feb. 19, 2019. Accessed on June 1, 2020, arXiv:1810.04719 [eess.AS]

[46] D. Dimitriadis and P. Fousek, Developing On-Line Speaker Diarization System, Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, pp. 2739-2743, August 2017.

[47] "Rich Transcription Evaluation", Accessed on June 12, 2020. [Online]. Available: https://www.nist.gov/itl/iad/mig/rich-transcription-evaluation

[48] C. Aliprandi, A. Del Pozo, R. Cassaca et al., Automatic Live Subtitling: state of the art, expectations and current trends, April 2014. Accessed on Apr. 23, 2020. [Online]. Available: https://www.academia.edu/17214759

[49] "NER", Accessed on June 20, 2020. [Online]. Available: https://en.wikipedia.org/wiki/NER_model

[50] "Word Error Rate", Accessed on June 20, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Word_error_rate

[51] "BLEU", Accessed on June 20, 2020. [Online]. Available: https://en.wikipedia.org/wiki/BLEU

[52] Evaluation of Machine Translation, Accessed on June 20, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Evaluation_of_machine_translation

[53] N. Ryant, dscore, Accessed on May 3, 2020. [Online]. Available: https://github.com/nryant/dscore

[54] NIST, The 2009 (RT-09) Rich Transcription Meeting Recognition Evaluation Plan. Accessed on May 3, 2020. [Online]. Available: https://bit.ly/35OEhVu

[55] "IBM Watson", Accessed on May 2, 2020. [Online]. Available: www.ibm.com/cloud/watson-speech-to-text

[56] "Speechmatics", Accessed on May 2, 2020. [Online]. Available: https://www.speechmatics.com/

[57] "EML", Accessed on Mar. 3, 2020. [Online]. Available: https://www.eml.org

List of figures

List of tables

Selbständigkeitserklärung

Mit der Abgabe dieser Arbeit versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat (Bei Teamarbeiten gelten die Leistungen der übrigen Teammitglieder nicht als fremde Hilfe).

Der/die unterzeichnende Studierende erklärt, dass alle zitierten Quellen (auch Internetseiten) im Text oder Anhang korrekt nachgewiesen sind, d.h. dass die vorliegende Arbeit keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

Zürich, 23. Oktober 2020

Unterschrift Studierende/r: