Speaker Diarization

A BRIEF OVERVIEW

PARIJAT GHOSHAL

TABLE OF CONTENTS

1 Introduction

An increasing number of audio data such as televised broadcasts, telephone calls, and even meeting recording, are generated every day. Speaker diarization could provide some assistance in labelling these audio data with some information regarding the participants i.e. who spoke and when.

This review shall provide an overview about the domain of speaker diarization. Focussing on the architecture of a theoretical diarization framework, first the different parts of a diarization framework and its underlying processes will be explained. Then the evaluation metric to measure the accuracy of such a system will be introduced. Next the different state-of-the-art systems and their key features will be discussed, followed by a quick overview of competitions and their focus. Afterwards, the practical aspects of training diarization system will be discussed, namely the types of available corpora, frameworks, and tools.

2 Speaker Diarization

In an audio source, speaker diarization is the process of marking speaker changes between a detected portion of speech and other segments of speech (which are either non-speech or speaker change events) of the same speaker, (a.k.a. “who spoke when”). The research in speaker diarization focusses primarily on audio from broadcast news, recorded meetings, and telephone conversations [1]. Table 1 demonstrates the key differences between of these types of audio data [2].

Telephony conversations	Broadcast news	Meeting speech
Number of speakers limited to 2 or at most 3 persons	Number of speakers could be 10 persons or more	The number of speakers is limited to the capacity of the meeting room
Music and other audio contents do not exist	Some parts of the file may contain music or commercials	music or commercials does not exist
Recording channel and environment do not usually change	The recording condition of each speaker may vary	Variations in recording quality, including impulse noises, reverberation and variable speech levels may exist
The recording channel and environment are different for each speaker	The recording channel and environment may be different for each speaker	All the conversations take place in one place
Average speaker change duration is usually so short	Average speaker change duration is longer	Average speaker change duration may be short
Normal existence of overlapping regions where two or more speakers speak simultaneously	Normally there is no overlapping regions between speaker utterances	Normally there are overlapping regions between the speech of two speakers
Only one recording microphones is applied	the recording is performed using several microphones	The recordings may be performed with one or two channels

Table 1 Differences between speaker diarization of meetings, broadcast news and telephony conversations from [2]

3 Diarization System Architecture

A speaker diarization system should, given an audio input, return information about speaker changes as its output. The challenges of a diarization task depend on the type of audio being analysed, and the algorithms being implemented in each step of the diarization process. Consequently, a plethora of architectures exist for diarization systems. Figure 1 shows the architecture of diarization network, which is a mostly conflated version of all the multiple frameworks. Depending on the focus of the diarization task, and input data, some of the building blocks may be not required (e.g. pre-processing), optional (e.g. resegmentation), merged (e.g. segmentation and clustering), or even not applicable (e.g. embedding extraction). Nonetheless, I will be using this

framework as an exemplary guide to describe the different processes and methods used for speaker diarization.

Figure 1 Theoretical diarization framework

4 Diarization methods and algorithms

The following sections provides a brief overview of systems and approaches used for speaker diarization. The following subsections correspond to the different parts of the system architecture mentioned beforehand (see 3). For each subsection, I summarised the findings from multiple papers, and organised them into the appropriate part of the theoretical architecture framework (see Figure 1). These are only a selection of possible approaches, which are mentioned in significant detail in [1]–[3].

4.1 Pre-processing measures

The data pre-processing methods are domain specific (e.g. noise reduction using Wiener filtering). In the meeting setting, multiple microphones of different qualities located at different position can be used, which results in audio data with multiple channels. A common approach to multi-channel speaker diarization includes acoustic beamforming [4].There are many proposed methods to extend the method used in mono-channel diarization to tackle multiple channel audio. These include doing separate speaker diarization for each channel and then merging the outputs, using the longest detected speaker segments[5]. Other methods entail performing the diarization step for the channel with the best signal-to-noise ratio (SNR) [6]. Another approach is to combine the all channels into a single channel and then perform mono-channel diarization without considering the time difference of arrival of the audio to microphone between channels, which could lead to low quality results [7].

4.2 Feature extraction

In part of a diarization framework, the acoustic features (such as mel-frequency cepstral coefficients (MFCC), etc.) are extracted from speech data. For multi-stream audio data, the time-delay-of-arrival (TDOA) of speech to each microphone can also be considered as a feature, with the assumption that the speakers do not move when they speak.

4.3 Speech Detection

The goal of this step is to recognize sections of speech in the audio data. Common methods for extracting these segments include models which are trained on labelled training data, e.g. maximum-likelihood classification with Gaussian mixture models (GMM) or even multistate hidden Markov models (HMM). A very simple system using a speech/non-speech detection model can be used to eliminate non-speech frames [8]. Removing silence from the audio data can also be done either in the beginning, using a phone recognizer [9], or at the end using a word recognizer [10]. Other methods for removing silence also exist.

If the speech detection process is earlier in the diarization pipeline the errors incurred at this state cannot be rectified in later stages. For finding non-speech segments, energy-

based methods are not always recommended. These methods perform poorly on audio data from meetings[1] (especially for recordings made with a distant microphone). It is recommended to use pretrained speech/non-speech GMM[11].

4.3.1 Speech activity detection

Speech activity detection (SAD) consists of methods to label speech and non-speech segments in audio data. If the SAD is inaccurate, then the diarization error rate (DER) increases. Hence, the labelling non-speech segment should be given importance to achieve better performance of the system[8].

Models that haven been trained on speech and non-speech data with discriminant classifiers (e.g. linear discriminant analysis (LDA) with MFCC or support vector machines (SVM)) tend to perform better. However, their dependence on training with external data makes them susceptible to changes in acoustic conditions. Hybrid approaches are a potential solution to this problem, where one implements an energy-based detection to identify speech and non-speech segments in the audio stream[3].

4.4 Segmentation

Speaker segmentation is a fundamental part of the speaker diarization process. During this process the audio stream is split into either single speaker segments, or alternatively changes in speakers (speaker turns) are detected. The segmentation approaches can be silence based, model based, and distance based methods, but other methods also exist[2].

4.4.1 Silence based methods

The goal of this step is to recognize points in the audio data which are likely to be change points. If the audio input did not underdo a segmentation process (e.g. SAD was not done), then the change detection system searches the audio stream for both speech/non-speech change points, as well as speaker change points. If the audio stream has undergone some form a segmentation process (e.g. SAD was performed) then the change detection system searches speech segments for speaker change points.

4.4.2 Model based methods

A classification model is trained on a set of training data containing features from different speaker classes. Then the model is run on an audio stream, and the stream is classified according to the model. Boundaries found by the classification model indicate speaker change. The drawback of this approach is that prior knowledge about the input data is required.

A speaker independent approach is the universal background model (UBM). It is comparable to the model of a generic speaker (containing the average features of the training population), but such a model can be too nonspecific, and may not classify correctly [2].

4.4.3 Distance based methods

These approaches for change detection, which involve looking at adjacent windows of data and calculating a distance metric between the two, then deciding whether the windows originate from the same or a different source. The differences between them lie in the choice of distance metric and thresholding decisions.

The most common method used in speaker diarization is the Bayesian information criterion (BIC) and ΔBIC metric. However, this method requires a penalty term, which is

[1] paper shuffling, coughing, laughing etc.

hard to estimate, and model performance is linked to this penalty term. Furthermore, BIC based methods are computationally expensive.
The generalized likelihood ratio (GLR) is an alternative to BIC, which is a likelihood-based metric and uses the ratio between two hypotheses (H0 : current audio segment is spoken by the same speaker as the past audio segment, H1: current audio segment is not spoken by the same speaker as the past audio segment).
Kullback–Leibler (KL) divergence is another alternative metric, where one that estimates the distance between two random samples. However, KL may not always be suitable as it is asymmetric; hence the KL2 metric which is symmetric is popular for finding differences between audio segments.
The information change rate (ICR), can also be used to find differences between neighbouring audio segments. The ICR calculates the change in information that occurs by merging any two speech segments by using the distance between audio segments in a space of relevance variables, with maximum mutual information or minimum entropy. This measure is also computationally efficient and more robust than BIC based methods[12].

4.5 Embedding extraction

In this step embeddings are calculated from the previously extracted segments or even set of features. Usually, these are either i-vectors (see [13]) or d-vectors (see [14]). Then these embeddings become the basis for the upcoming steps in the diarization system, which could be clustering, or segmentation (see 0, and [15]–[17]).

4.6 Clustering

The goal of this step is to cluster segments belonging to the same speaker. This process should under ideal circumstances produces a single cluster per speaker present in the audio data, where all the segment within the cluster belong to one speaker. The prevalent method for this process entails doing AHC with BIC based stopping criterion[1].
There are two **main methods** for clustering, namely the top-down and the bottom-up approaches. The top-down approach consists of starting with a few clusters and for bottom-up approach one starts with many clusters. The aim of both methods is to reach the optimal number of clusters in an iterative manner. Both approaches are based on HMM, where a single state is a GMM which represents the speaker. The speaker turns are the transitions between states [3].
A comprehensive insight into the topic of top-down and bottom up approaches can be found in [18]. Their results demonstrate that none of the approaches on NIST RT evaluation datasets (average DERs of 21% and 22%) are superior. However, combining both approaches takes advantage of the complementary nature of both methods and delivers an average DER of 17%.

4.6.1 Bottom-up approach

Also known as agglomerative hierarchical clustering (AHC), the aim is to train a large number of clusters. Then reduce the number of clusters until only one cluster per speaker remains. For audio data, the audio stream is divided into more segments than the expected number of speakers.
Every cluster is modelled using a GMM. Then iteratively, the clusters which are close to each other are merged. A new GMM is trained on the new cluster, which is made out of two merged clusters. The closeness between the clusters are calculated using distance metrics (see 4.4.3). The reassignment of frames to clusters are done using the Viterbi

realignment. This cluster merging and distance calculation process is iterated until a stopping criterion[2] has been reached after which only one cluster per speaker should remain [3].

4.6.2 Top-down approach

Unlike the bottom-up approach, the process starts with a single GMM which is trained on all the available speech segments of the audio stream. All of the speech segments are marked as unlabelled. Using some selection criteria to select training items from the unlabelled data, a new model for the speakers are added to the overall model iteratively. Each iteration also contains a Viterbi realignment and adaptation step. When a speech segment is attributed to a speaker model, they are marked as labelled. Either use a stopping criterion can be used to decide when to stop the process, or the process continues until there are no unlabelled segments left[3].

4.6.3 Stopping criterion

The stopping criterion is central to the performance of the model, as it determines when the optimum number of clusters has been reached. Over-clustering results in creating corrupted clusters containing speech segments from different speakers. However, the corrupted clusters can be of use if the aim is to group similar speakers or acoustic environments. Under-clustering creates many clusters with speech segments from the same speaker, which can be useful if a speaker is in different acoustic environments [1].

4.7 Simultaneous Segmentation and Clustering

In most systems, the segmentation and clustering methods are done in a single step. The process entails representing the acoustic classes using either HMM, GMM, or mixed models. Then an expectation maximisation training or MAP adaptation is implemented to get the closest model. Afterwards the Viterbi algorithm is used for reassigning the data to new closest models. This process is run multiple times to achieve model stabilisation. This iterative process helps the class distribution to be able to adapt to the data, when a class is created or eliminated.

Despite being slower, simultaneous segmentation and clustering has advantages over the single pass methods. These advantages include overcoming issues with segmentation errors, which can be corrected in the re-segmentation process. Instead of using local information to locate speaker change, the Viterbi algorithm takes all of the data into consideration. Whilst doing the frame assignment with the Viterbi algorithm, a minimum duration for the speaker turn is used to avoid unrealistically short speaker turns. The duration period is based on the minimum length of a speaker turn [3].

4.8 Resegmentation

This is the final stage of the clustering diarization process. It is usually done with Viterbi decoding on the final cluster models and non-speech models. The aim of this process is to improve the quality of the segmentation and retrieve small segments that may have been dropped during the clustering process. Hence, this process reduces the false alarm component of the error rate [19].

---

[2] Bayesian Information Criterion (BIC) , Kullback-Leibler (KL)-based metrics ,Generalized Likelihood Ratio (GLR)etc.

5 Diarization error rate

For diarization tasks, the standard performance metric is the diarization error rate (DER) formulated by NIST Rich Transcription evaluation metric. The DER is the percentage of speech segment which are incorrectly assigned. It is defined as follows [20]:

$$Error_{SpkrSeg} = \frac{\sum_{allsegs} \left\{dur(seg) * \left(\max\left(N_{Real}(seg), N_{Out}(seg)\right) - N_{Correct}(seg)\right)\right\}}{\sum_{allsegs} \{dur(seg).N_{Ref}(seg))\}}$$

Dur(*seg*): speech segment duration
Num_{Real}(*seg*): number of real speakers in the speech segment
Num_{Out}(*seg*): number of output speakers in the speech segment
$Num_{Correct}$(*seg*): number of correctly recognised real speakers in the speech segment
N_{Ref}(*seg*): number of reference speakers in the segment,
NSys(*seg*): number of system speakers in the segment

speaker error time: dur(*seg*) * {min(N_{Ref}(*seg*), N_{Sys}(*seg*)) – $N_{Correct}$(*seg*)}.
missed speaker time: dur(*seg*) * (N_{Ref}(*seg*) – N_{Sys}(*seg*))
false alarm speaker: dur(*seg*) * (N_{Sys}(*seg*) – N_{Ref}(*seg*))

The numerator is the speaker diarization error time, and consists of speaker error time, missed speaker time, and false alarm speaker time. The DER is time-weighted, for speakers with short speaker turns, the impact on the final score is small. An output label generated by the diarization system cannot always be the precise time labels according to the ground truth labels. Hence, as a way to reduce DER, in most cases a non-scoring collar of 250ms is implemented at every ground truth segment boundary.

6 State-of-the-art speaker diarization systems

In this section, I describe the key methods used in a set of comparable (see 6.4) speaker diarization systems. The systems are grouped according to the methods used by them.

6.1 i-vectors

There are systems which use i-vectors to improve the diarization process. In [15], the researchers extracted i-vectors from short segments of multi-speaker conversations, which are then ascribed to speaker clusters based on their cosine score or probabilistic linear discriminant analysis (PLDA). They used an unsupervised method to calibrate the PLDA scores (which controls the stopping criterion of the clustering process). Their findings show that when the i-vectors are generated using dense sampling from overlapping temporal segments, then the performance of the diarization system improves. Furthermore, for each conversation they applied a principal component analysis (conversation-dependent PCA). The conversation-dependent PCA not only caused a dimensionality reduction of the i-vectors, but also contributed to a DER reduction. Finally, the researchers demonstrated that using PLDA results in better accuracy than the cosine measure for the clustering process. They achieved the best DER of 13.7% for the entire dataset.

Yet a different approach rendering better results is shown in [16]. A variation of the mean shift algorithm, which can work in conjuncture with the cosine distance, was implemented for the clustering process with i-vectors. Moreover, a Viterbi based resegmentation was implemented to further refine the cluster quality. This system achieved a global DER of 12.4 %.

6.2 Resegmentation with Variational Bayes

An improvement of the model introduced in [15] is achieved by focusing on the resegmentation process of the diarization pipeline [17]. The researchers' methods were inspired by Variational Bayes (VB)[21], as they adapted and extended VB to match the data in the diarization pipeline. The subspace of VB (originally defined as the speaker factors) was substituted with i-vectors. Moreover, they extended VB with an HMM to take speaker turns into consideration. This VB based approach improved the diarization process and resulted in a DER of 11.5%.

6.3 Deep learning approaches

Deep learning methods for speaker diarization exhibit promising results (see 6.4). For the speaker diarization task, deep learning algorithms are used as part(s) of the framework to reduce the DER. The specific characteristics[3] of different types of neural networks are also used at stages of the framework, where the traditional approaches (see 4) may be at their limit.

6.3.1 Embeddings

In a partial departure from the methods used in [15], [17], the proposed architecture in [23] removes the i-vector extraction process. This new approach entails learning fixed-dimensional embeddings for audio segments of different lengths. Then the system pairwise calculates the likelihood of a given embedding to belong to the same speaker or a different speaker. A deep neural network (DNN) is used for the simultaneous calculation of the embedding and its scoring metric. Then a conversation-dependent PCA is applied to every output, followed by an AHC of all the outputs. Finally, a VB based resegmentation [17] is done to improve the diarization results. This approach yields a DER of 9.9%.

6.3.2 Convolutional neural networks

Using supervised learning, Convolutional neural network (CNN) is trained on the spectrograms of the voice stream for speaker change detection (SCD). From the training data, which contains the absolute speaker change information (labels of the ground truth data), fuzzy labels are created with the help of a labelling function. These fuzzy labels reflect the probability of a speaker change occurring (number between 0 and 1) at a given time. The CNN is trained using these fuzzy labels. Thus, when presented with a spectrogram, the CNN tries to replicate the labelling function of the training data. Consequently, the output of the CNN is the probability of a speaker change occurring for a specific time segment[24]. For a diarization task, the aforementioned probabilities can be used to segment the audio data and subsequently create i-vectors of the segments. The i-vectors can be used for the clustering, and then the resegmentation step to further improve the results [25]. On the Callhome English 2-speaker subset, this approach yields a DER of 7.84%.

[3] translation invariance for CNN and internal state (memory) for RNN

6.3.3 Recurrent neural networks

Long short-term memory (LSTM) networks are one type of recurrent neural networks (RNN). In an LSTM based model for speaker diarization [26], acoustic features are extracted from the audio stream by using a fixed length sliding window, which becomes the input of the LSTM. The last frame output of the LSTM is a d-vector of a single sliding window. The audio stream is divided into speech segments, which are non-overlapping. Then all of the d-vectors corresponding to a specific speech segment are aggregated to create segment embeddings. Then the segment embeddings are used for the clustering process. Four different kinds of clustering methods namely naïve, k-means, links, and spectral clustering were used. The spectral clustering led to the best results. The model was trained on anonymized voice searches and tested on an out of domain dataset, with a minimum DER of 12%.

An improvement of the previous approach[26], is achieved by replacing the clustering step with an unbounded interleaved-state recurrent neural network (UIS-RNN)[27]. In a UIS-RNN, every speaker is represented by an instance of the RNN, with parameter sharing amongst the instances. The UIS-RNN can generate an unlimited number of RNN instances. These RNN instances within an UIS-RNN can have different states which are interleaved in the time domain.

In the domain of speaker diarization, it can be quite difficult to determine for any given utterance the number of speakers that are involved. Hence, the distance dependent Chinese restaurant process(ddCRP)[28][4] is used to model the speaker turn behaviour. An advantage of this method is that instead of analysing a single utterance, the UIS-RNN structure takes advantage of the labelled data and is able to learn some high-level structure between the speakers and their utterances. The DER achieved by this method is state-of-the-art, namely 7.6%.

6.4 Comparison

Comparing diarization systems can be difficult if the models are not evaluated on the same dataset. Therefore, I chose a set of diarization systems which are validated on the same dataset (Callhome corpus) but implement different methods. Table 2 shows DER and methods implemented in diarization systems described beforehand (see 0-6.3.3).

The type of algorithm used in the clustering method is important and the resegmentation improves the cluster quality, and consequently reduces the DER [15], [16].

An evolution of the methodology by the same group of researchers[5] can be observed in systems in [15], [17], and [23]. A gradual decline of the DER was achieved with every improvement to the system, from using i-vectors and PLDA, to refining the system output with resegmentation strategies, to finally discarding i-vectors, and using DNN. The DNN was used to create embedding, and this stage of the diarization pipeline corresponds to the embedding extraction process(see 4.5).

The implementation of CNN in [25], shows how it was used as a model based segmentation method (see 4.4.2). The success of segmentation strategy may be due to the CNNs being translation invariant[6].

---

[4] For details on how ddCRP work see [28]

[5] With the exception of [23], where more researcher were involved

[6] https://en.wikipedia.org/wiki/Convolutional_neural_network

Finally, the best system to date [27] on this evaluation set shows a 4.4% difference in DER when compared to its immediate predecessor with a similar approach [26]. This improvement of the overall system accuracy can be ascribed to the choice of using the UIS-RNN. RNNs are particularly suited for sequential data (e.g. a conversation, audio) because they have an internal state (memory). The significant innovation of this work is the introduction of the interleaved-state of the RNN in the time domain. This aspect is able to emulate conversations, as speakers join a discussion at different points in time and stop participating at different (speaker dependant) times as well. Hence, using the ddCRP to model the speaker participation works also in favour of this approach. In this case neural networks replace the embedding extraction as well as clustering aspect of the of diarization framework.

Ref.	Title	Authors	Methods	Year	DER
[27]	Fully Supervised Speaker Diarization	Zhang et al.	UIS-RNN, ddCRP	2018	7.6
[23]	Speaker diarization using deep neural network embeddings	Garcia-Romero et al.	DNN, embeddings, VB resegmentation	2017	9.9
[17]	Diarization resegmentation in the factor analysis subspace	Sell and Garcia-Romero	i-vector scoring with PLDA, VB resegmentation	2015	11.5
[26]	Speaker Diarization with LSTM	Wang et al.	LSTM, d-vectors, spectral clustering	2017	12 (5.971[1])
[25]	Speaker diarization using convolutional neural network for statistics accumulation refinement	Zajíc et al.	CNN, i-vectors, GMM	2017	7.841[1]
[16]	A Study of the Cosine Distance-Based Mean Shift for Telephone Speech Diarization,	Senoussaoui et al.	i-vector, mean shift with cosine distance	2014	12.4
[15]	Speaker diarization with PLDA i-vector scoring and unsupervised calibration	Sell and Garcia-Romero	i-vector scoring with PLDA	2014	13.7

Table 2 Speaker diarization systems evaluated on the Callhome corpus ([1]only 2 speaker English subset), with key methods and DER(%)

7 Diarization Competitions

7.1 NIST

From 2004 to 2009, NIST organized a series of benchmark evaluations within the Rich Transcription (RT) campaigns. One of the tasks involved speaker diarization of different sets of data. A common characteristic of these evaluations was that only a set of basic characteristics about the recording, such as recording scenario, language, and file format, were given to the participants. Information about the speakers were not provided. It was allowed to use external data to build the model[3]. The datasets, benchmarks, and evaluation metrics (see 5) set by NIST are still being used by researchers.

7.2 DIHARD

DIHARD Speech Diarization Challenge[7], which started in 2018, is moving away from creating systems for three main types of audio data typically used in diarization research (see 0). The DIHARD Challenge focusses on types of diarization problems on which the state-of-the-art systems will perform poorly, such as clinical interviews, YouTube videos,

[7] DIHARD 2018 overview: https://coml.lscp.ens.fr/dihard/2018/overview.html

or recordings in restaurants. The best systems of the 2 tasks proposed by the challenge have DER of 23.73%, and 35.51%[8]. This shows that there is a lot of room for improvement for the new types of audio data.

8 Corpora

Many evaluation datasets (including the ones used by NIST RT evaluations) are from the catalogue of the Linguistic Data Consortium (LDC). Table 3 shows a list of diarization datasets, including a brief description and their pricing and availability.

However, many of the datasets mentioned in the older NIST challenges are difficult to locate or are only partially available. Diarization datasets take a significant amount of effort to create. Furthermore, they also contain background noise, which emulates a realistic recording situation, but may not be of use to someone who is not interested in this feature. The Synthetic Diarization Corpus is synthetically generated from the LibriSpeech corpus, which contains audio book recordings [29]. The use of such a corpus could serve as a starting point for a diarization project. However, it is not recommended to use this corpus to train systems which should be able to work in realistic (non-synthetic) contexts.

ICSI Meeting Corpus, and AMI Meeting Corpus are two corpora containing audio from meetings. OpenSLR[9] also provides a some freely available speech corpora.

Name	Description	Pricing	Source	Usage
2000 NIST Speaker Recognition Evaluation	148.9 hours of conversational telephone speech	$2400	LDC[10]	[27] [23] [17] [16] [15]
CALLHOME American English Speech	20 unscripted 30-minute telephone conversations between native speakers of English	$1500	LDC[11]	[25] [26]
ICSI Meeting Corpus	70 hours of meeting recordings	Free	ICSI[12]	
AMI Meeting Corpus	100 hours of meeting recordings	Free	AMI[13]	
Synthetic Diarization Corpus	90 + hours of synthetic dialogs from LibriSpeech corpus	Free	emr.ai[14]	

Table 3 List of datasets for training and evaluating diarization systems, brief description, pricing, and usage in section 6

[8] DIHARD 2018 results: https://coml.lscp.ens.fr/dihard/2018/results.php
[9] http://www.openslr.org/resources.php
[10] https://catalog.ldc.upenn.edu/LDC2001S97
[11] https://catalog.ldc.upenn.edu/LDC97S42
[12] http://groups.inf.ed.ac.uk/ami/icsi/
[13] http://groups.inf.ed.ac.uk/ami/corpus/
[14] https://github.com/EMRAI/emrai-synthetic-diarization-corpus

9 Tools

Tools play an important role in building diarization models. An important task concerns converting an audio signal to a set of parameters. Thus, tools that can perform acoustic feature extraction are crucial. Libraries for clustering and calculating different metrics are also required for building diarization models. In this section, I look at a subset of open source tools that help with speaker diarization.

9.1 Complete Frameworks

These are tools that can do many parts of a speaker diarization framework, such as feature extraction, speech detection, segmentation, and clustering. My attempt at showing the features of a few selected diarization tools [30]–[33] and what they do at a specific stage of the diarization process can be seen in Table 4. The columns of the table are based on the diarization system architecture (see 3). Generating the table proved to be difficult, as the documentation of these frameworks are somewhat vague about the algorithms uses at specific stages.

Tool	Year	Language	Input format	Feature extraction	Speech detection	Segmentation	Embedding extraction	Clustering	Resegmentation
LIUM SpkDiarization[33]	2013	Java	audio: sphere, wave, mp3	MFCC	GLR, BIC	Viterbi decoding	i-vector	AHC, BIC	Viterbi decoding
SIDEKIT[30] , **S4D**[31]	20162018	Python	acoustic feature: spro4, htk, audio: raw, wav, sphere	LFCC, MFCC	Energy-based Voice Activity Detection (VAD)	BIC	i-vector	AHC, BIC,CLR	
pyAudioAnalysis [32]	2015	Python	Audio: WAV	MFCC	SVM	HMM		k-means, FLsD[15]	

Table 4 Overview of diarization Tools and their features

pyannote-audio[16] [34]–[36] performs implicitly only few of the tasks, but it easily allows to train LSTM based speech activity detection, speaker change detection, speaker embedding model, and diarization models. Moreover, one can connect the pipeline to a pre-existing database. During the implementation stage the individual diarization steps are not transparent.

9.1.1 Ease of use

The ease of use of these tools depends on one's knowledge of the programming language, the availability and quality of the documentation.
LIUM SpkDiarization [33] has a pretty steep learning curve, but it could be due to my proficiency in Java. S4D[31] and SIDEKIT[30] were quite easy to use, and tutorials were helpful. However, many parts of the S4D tutorial contain errors, or may not have been updated to match the latest version of the tools, which cause an initial setback when using these Python based frameworks.

[15] Fisher Linear Semi-Discriminant analysis
[16] https://github.com/pyannote/pyannote-audio

Finally, pyAudioAnalysis [32] is one of the easiest tools to use, and requires very few lines of to get started. It has a well-structured documentation with relevant examples and is suitable for beginners.
I also wanted to try ALIZÉ[37], but the documentation on their website is non-existent. I encountered issues in the feature extraction state, as the framework did not perform the task, even after installing all required tools and libraries.

9.2 Other tools

The following list of tools could be useful for feature extraction, evaluation and other specialized calculations and clustering.

BeamformIt[17] : This tool performs acoustic beamforming on multi-channel audio [4], [38]. It could be of use for working with meeting data.

pyannote.metrics[18]: A tool that allows the calculation of many diarization related evaluation metrics (e.g. DER), but it also contains many tools for the visual analysis of errors. [39] This could allow the user to visualize the part of the audio segment where the segmentation was inadequate, or a diarization error occurred.

uis-rnn[19]**:** A tool that allows to train the UIS-RNN[27] (see 6.3.3). The input to the system is a numpy.savez (.npz)[20] file.

SpectralCluster[21] : This tool does the spectral clustering mentioned in [26], and takes numpy arrays as input.

10 Conclusion

Speaker diarization is a versatile field of research. To approach the task of providing a review of this domain I created a theoretical framework for a diarization model. This theoretical framework contained all the important building blocks of a diarization system. Whit this theoretical framework as a guide I summarised the findings from multiple papers and organised them into the appropriate part of the theoretical architecture framework.
In the second block of this review, I looked at how diarization systems can be compared and chose papers which implemented different diarization methods but evaluated their findings on the same data set. I also observed that in deep learning approaches specific neural networks are used to replace one of the building blocks observed before. The current trend in diarization competition is moving towards more difficult and versatile types of audio data. For diarization tools it is absolutely indispensable to have a tool which

[17] https://github.com/xanguera/BeamformIt
[18] https://github.com/pyannote/pyannote-metrics
[19] https://github.com/google/uis-rnn
[20] https://docs.scipy.org/doc/numpy-1.15.0/reference/generated/numpy.savez.html#numpy.savez
[21] https://github.com/wq2012/SpectralCluster

preforms an acoustic feature extraction. There are many frameworks for doing speaker diarization but not all of them are transparent regarding the individual steps.
My review barely scratches the surface of what already has been done in this domain of research. Research in the field of speaker diarization has been an ongoing process for decades, hence this review merely tries to provide a synopsis for orientation. However, I strived to provide extensive references for all the material that was mentioned just briefly in order to facilitate a deep dive into this topic.

11 References

[1] S. E. Tranter and D. A. Reynolds, “An overview of automatic speaker diarization systems,” *IEEE Trans. Audio. Speech. Lang. Processing*, vol. 14, no. 5, pp. 1557–1565, 2006.

[2] M. H. Moattar and M. M. Homayounpour, “A review on speaker diarization systems and approaches,” *Speech Commun.*, vol. 54, no. 10, pp. 1065–1103, Dec. 2012.

[3] X. Anguera *et al.*, “Speaker diarization: A review of recent research,” in *IEEE Transactions on Audio, Speech & Language Processing*, 2012, pp. 356–370.

[4] X. Anguera, C. Wooters, and J. Hernando, “Acoustic Beamforming for Speaker Diarization of Meetings,” *IEEE Trans. Audio. Speech. Lang. Processing*, vol. 15, no. 7, pp. 2011–2022, 2007.

[5] C. Fredouille, D. Moraru, S. Meignier, L. Besacier, and J.-F. Bonastre, “The NIST 2004 spring rich transcription evaluation: two-axis merging strategy in the context of multiple distance microphone based meeting speaker segmentation,” in *RT2004 Spring Meeting Recognition Workshop*, 2004.

[6] Q. Jin and T. Schultz, “Speaker segmentation and clustering in meetings,” in *Eighth International Conference on Spoken Language Processing*, 2004.

[7] D. Istrate, C. Fredouille, S. Meignier, L. Besacier, and J. F. Bonastre, “RT05S evaluation: Pre-processing techniques and speaker diarization on multiple microphone meetings,” in *in NIST 2005 Spring Rich Transcrition Evaluation Workshop*, 2005.

[8] C. Wooters, J. Fung, B. Peskin, and X. Anguera, “Towards robust speaker segmentation: The ICSI-SRI fall 2004 diarization system,” in *In RT-04F Workshop*, 2004.

[9] R. Sinha, S. E. Tranter, M. J. F. Gales, and P. C. Woodland, “The cambridge university march 2005 speaker diarisation system,” in *Ninth European Conference on Speech Communication and Technology*, 2005.

[10] X. Zhu, C. Barras, S. Meignier, and J.-L. Gauvain, “Combining speaker identification and BIC for speaker diarization,” in *Ninth European Conference on Speech Communication and Technology*, 2005.

[11] X. Anguera, C. Wooters, B. Peskin, and M. Aguiló, “Robust speaker segmentation for meetings: The ICSI-SRI spring 2005 diarization system,” in *International Workshop on Machine Learning for Multimodal Interaction*, 2005, pp. 402–414.

[12] K. J. Han and S. S. Narayanan, “Novel inter-cluster distance measure combining GLR and ICR for improved agglomerative hierarchical speaker clustering,” in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 2008, pp. 4373–4376.

[13] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-End Factor Analysis for Speaker Verification,” *IEEE Trans. Audio. Speech. Lang. Processing*, vol. 19, no. 4, pp. 788–798, May 2011.

[14] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, “Deep neural networks for small footprint text-dependent speaker verification,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4052–4056.

[15] G. Sell and D. Garcia-Romero, “Speaker diarization with plda i-vector scoring and unsupervised calibration,” in *2014 IEEE Spoken Language Technology Workshop*

(SLT), 2014, pp. 413–417.
[16] M. Senoussaoui, P. Kenny, T. Stafylakis, and P. Dumouchel, “A Study of the Cosine Distance-Based Mean Shift for Telephone Speech Diarization,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 1, pp. 217–227, Jan. 2014.
[17] G. Sell and D. Garcia-Romero, “Diarization resegmentation in the factor analysis subspace,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4794–4798.
[18] N. Evans, S. Bozonnet, D. Wang, C. Fredouille, and R. Troncy, “A Comparative Study of Bottom-Up and Top-Down Approaches to Speaker Diarization,” *IEEE Trans. Audio. Speech. Lang. Processing*, vol. 20, no. 2, pp. 382–392, Feb. 2012.
[19] C. Barras, X. Zhu, S. Meignier, and J.-L. Gauvain, “Improving Speaker Diarization.” p. 5, Nov-2004.
[20] NIST, “Fall 2004 Rich Transcription (RT-04F) Evaluation Plan,” *Online www. nist. gov/speech/tests/rt/rt2004/fall/docs/rto4feval-plan-v14. pdf*, pp. 1–27, 2004.
[21] P. Kenny, “Bayesian analysis of speaker diarization with eigenvoice priors,” *CRIM, Montr. Tech. Rep.*, 2008.
[22] S. H. Yella, A. Stolcke, and M. Slaney, “Artificial neural network features for speaker diarization,” in *2014 IEEE Spoken Language Technology Workshop (SLT)*, 2014, pp. 402–406.
[23] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, “Speaker diarization using deep neural network embeddings,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 4930–4934.
[24] M. Hrúz and Z. Zajíc, “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, pp. 4945–4949.
[25] Z. Zajíc, M. Hrúz, and L. Müller, “Speaker diarization using convolutional neural network for statistics accumulation refinement,” in *Proceedings Interspeech (2017, in press)*, 2017, pp. 3562--3566.
[26] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. Lopez Moreno, “Speaker Diarization with LSTM,” *arXiv e-prints*, p. arXiv:1710.10468, Oct. 2017.
[27] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, “Fully Supervised Speaker Diarization,” *arXiv e-prints*, p. arXiv:1810.04719, Oct. 2018.
[28] D. M. Blei and P. I. Frazier, “Distance dependent Chinese restaurant processes,” *J. Mach. Learn. Res.*, vol. 12, no. Aug, pp. 2461–2488, 2011.
[29] E. Edwards *et al.*, “A Free Synthetic Corpus for Speaker Diarization Research,” in *International Conference on Speech and Computer*, 2018, pp. 113–122.
[30] A. Larcher, K. A. Lee, and S. Meignier, “An extensible speaker identification sidekit in Python,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5095–5099.
[31] P.-A. Broux, F. Desnous, A. Larcher, S. Petitrenaud, J. Carrive, and S. Meignier, “S4D: Speaker Diarization Toolkit in Python,” in *Interspeech 2018*, 2018.
[32] T. Giannakopoulos, “PyAudioAnalysis: An open-source python library for audio signal analysis,” *PLoS One*, 2015.
[33] M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin, and S. Meignier, “An Open-source State-of-the-art Toolbox for Broadcast News Diarization,” 2013.

[34] R. Yin, H. Bredin, and C. Barras, “Speaker Change Detection in Broadcast TV Using Bidirectional Long Short-Term Memory Networks.,” in *INTERSPEECH 2017*, 2017.
[35] R. Yin, H. Bredin, and C. Barras, “Neural Speech Turn Segmentation and Affinity Propagation for Speaker Diarization,” in *19th Annual Conference of the International Speech Communication Association, Interspeech 2018*, 2018.
[36] H. Bredin, “TristouNet: Triplet Loss for Speaker Turn Embedding,” in *42nd IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017*, 2017.
[37] J.-. Bonastre, F. Wils, and S. Meignier, “ALIZE, a free toolkit for speaker recognition,” in *Proceedings. (ICASSP ’05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, 2005, vol. 1, p. I/737-I/740 Vol. 1.
[38] X. Anguera Miró, *Robust speaker diarization for meetings*. Universitat Politècnica de Catalunya, 2006.
[39] H. Bredin, “pyannote.metrics: a toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems,” in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*, 2017.