SPEECH SEPARATION USING Bi - LSTM NETWORK

Poonam Dhankhar[1], Assistant Professor
[1]Maharaja Surajmal Institute of Technology, New Delhi

ABSTRACT

Audio separation is a fundamental problem in signal processing, and the most typical problem is called the "who spoke when" in a multi-speaker environment. We want to present a method of partitioning an audio stream with multiple people into homogeneous segments associated with each individual. In this paper, we implemented a Voice Activity Detector (VAD) module to separate out speech from non-speech. This is required to trim out silences and non-speech parts from the audio recording. A speech segmentation module removed the non-speech parts, and divided the input utterance into small segments. We build on the success of d-vector based speaker verification systems to develop a new d-vector based approach to speaker diarization. Specifically, we combined LSTM-based d-vector audio embeddings with recent work in nonparametric clustering to obtain a state-of-the-art speaker diarization system. Our system is evaluated on a standard public dataset - AMI Meeting Corpus Dataset.

This method will create a neural-network based embedding of the speech segments extracted by the speech segmentation process. The embeddings of the segments belonging to the same speakers were passed through the Clustering Algorithm which assigned the label of the speaker. This step was crucial as it assigned the labels to our embeddings, as well as the number of clusters, which indicates to us the number of speakers in the audio file. We achieved a 14.0% diarization error rate on the AMI Meeting Dataset.

Index Terms— Speaker diarization, deep learning, audio embedding, LSTM, spectral clustering

1. INTRODUCTION

Audio analysis is a field that includes automatic speech recognition (ASR), digital signal processing, and music classification, tagging, and generation. It is a growing subdomain of deep learning applications. Some of the most popular and widespread machine learning systems, virtual assistants Alexa, Siri and Google Home, are largely products built atop models that can extract information from audio signals.

Audio data analysis is about analyzing and understanding audio signals captured by digital devices, with numerous applications in the enterprise, healthcare, productivity, and smart cities. Applications include customer satisfaction analysis from customer support calls, media content analysis and retrieval, medical diagnostic aids and patient monitoring, assistive technologies for people with hearing impairments, and audio analysis for public safety. A typical speaker diarization system usually consists of four components: (1) Speech segmentation, where the input audio is segmented into short sections that are assumed to have a single speaker, and the non-speech sections are filtered out; (2) Audio embedding extraction, where specific features such as MFCCs [6], speaker factors [7], or i-vectors [8, 9, 10] are extracted from the segmented sections; (3) Clustering, where the number of speakers is determined, and the extracted audio embeddings are clustered into these speakers; and optionally (4) Resegmentation [11], where the clustering results are further refined to produce the final diarization results.

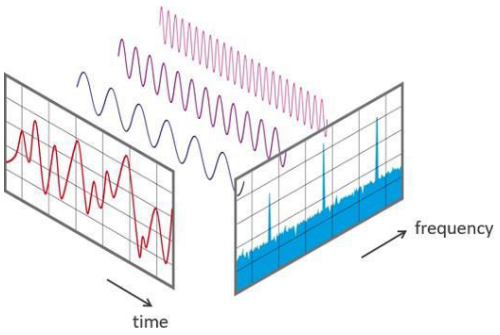

Fig 1. Audio signal in terms of time and frequency

There have been several attempts to apply spectral clustering [16] to the speaker diarization problem [17, 8]. However, to the authors' knowledge, our work is the first to combine LSTM-based d-vector embeddings with spectral clustering.

2. IMPLEMENTATION

The objective in this method is to present a method of partitioning an audio stream with multiple people into homogeneous segments associated with each individual.

Speech Detection

In this step, we use a Voice Activity Detector (VAD) module to separate out speech from non-speech. This is required to trim out silences and non-speech parts from the audio recording. Voice activity detection (VAD) is a technique in which the presence or absence of human speech is detected. This part has been completed using a module developed by Google called WebRTC. It's an open framework for the web that enables Real-Time Communications (RTC) capabilities in the browser. The voice activity detector is one of the specific modules

present in WebRTC. This basic working of WebRTC based VAD is as,

- WebRTC VAD is a Gaussian Mixture Model (GMM) based voice activity detector
- GMM model using PLP features
- Two full covariance Gaussians: One for speech, and one for Non-Speech is used

Speaker Segmentation

The idea is to exactly identify the location of speaker change point in the order to milliseconds which is achieved by segmenting the audio into windows with overlap. The size of the window determines the size of your segment. For this part we have tried to develop a BiLSTM network that is trained using a special SMORMS3 optimizer. SMORMS3 optimizer is a hybrid optimizer developed using RMSprop and Adam optimizers. SMORMS3 stands for "squared mean over root mean squared cubed". Given an audio recording, speaker change detection aims at finding the boundaries between speech turns of different speakers. The expected output of such a system would be the list of timestamps between spk1 & spk2, spk2 & spk1, and spk1 & spk4.

Combining VAD and Speaker Segmentation

Once the results from above modules were obtained, we combined them in a logical way such that we had obtained frames of arbitrary seconds depending on the voiced part and the speaker change part. This can be explained with an example, let us suppose we have first performed VAD and found that from 2 to 3 seconds there is some voice. In the next part of speaker segmentation, we found that at 2.5 seconds there is a speaker change point. So we split this audio frame of 1 seconds into two parts frame 1 from 2 to 2.5 seconds and then from 2.5 to 3. Similarly if we find that from 3 to 3.5 seconds there is some voice and then there is a silence of 1 seconds i.e. exactly at 4 sec some voice is coming into play. Now using the Speaker change part we found that at 4 seconds there is speaker change point again we combined it in such a way we defined there is new speaker at 4 seconds.

Embedding Extractions

This part now has to handle the idea of differentiating speakers. An embedding is a vector representation of data which could be used by the deep learning framework. First, we find the MFCC (Mel Frequency Cepstral Coefficient) of the audio segment. These are basically feature coefficients which capture the variations in the speech like pitch, quality, intonation etc of the voice in a much better way. They are obtained by doing a specialized Fourier Transform of the speech signal. In the next step, we use an LSTM based network which takes in the MFCCs and outputs a vector representation (embedding) which they call a d-vector. As mentioned in previous parts the frames extracted will go through the process of feature extraction. To extract d-vectors we use the pyannote libraries pretrained models.

Clustering using k-means algorithm

Clustering is one of the most common exploratory data analysis techniques used to get an intuition about the structure of the data. It can be defined as the task of identifying subgroups in the data such that data points in the same subgroup (cluster) are very similar while data points in different clusters are very different. K-Means algorithm is an iterative algorithm that tries to partition the dataset into 'K' pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous the data points are within the same cluster. The way k-means algorithm works is as follows:

- Specify number of clusters K.
- Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement.
- Keep iterating until there is no change to the centroids. i.e. assignment of data points to clusters isn't changing.
- Compute the sum of the squared distance between data points and all centroids.
- Assign each data point to the closest cluster (centroid).
- Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

Applying Loss Metric

The final part is to now evaluate how true we are. For this, we have used the PyAnnote library's metric module which contains the DER (Diarization Error rate) function that helps us to see how wrong we are in determining who spoke when.

- Hypothesis: It shows who spoke when in an audio.
- Ground Truth: It is the visualization of manually annotated audio files.

Diarization error rate (DER) is the standard metric for evaluating and comparing speaker diarization systems. It is defined as follows

$$\text{DER} = \frac{\text{false alarm} + \text{missed detection} + \text{confusion}}{\text{total}}$$

where false alarm is the duration of non-speech incorrectly classified as speech, missed detection is the duration of speech incorrectly classified as non-speech, confusion is the duration of speaker confusion, and total is the total duration of speech in the reference.

3. EXPERIMENTS

Model

The actual architecture of the network f is depicted in Figure below. It is composed of two Bi-LSTM (Bi-LSTM 1 and 2) and a multi-layer perceptron (MLP) whose weights are shared across the sequence. Bi-LSTMs allow processing sequences in forward and backward directions, making use of both past and future contexts. The output of both forward and backward LSTMs are concatenated and fed forward to the next layer. The shared MLP is made of three fully connected feed forward layers, using tanh activation function for the first two layers, and a sigmoid activation function for the last layer, in order to output a score between 0 and 1.

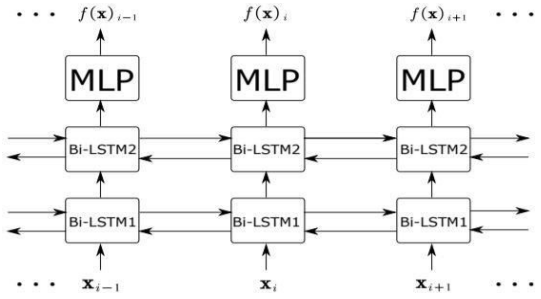

Fig 2. Model Architecture

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional (Bidirectional (None, 137, 256)          167936
_________________________________________________________________
bidirectional_1 (Bidirection (None, 137, 256)          394240
_________________________________________________________________
time_distributed (TimeDistri (None, 137, 32)           8224
_________________________________________________________________
time_distributed_1 (TimeDist (None, 137, 32)           1056
_________________________________________________________________
time_distributed_2 (TimeDist (None, 137, 1)            33
=================================================================
Total params: 571,489
Trainable params: 571,489
Non-trainable params: 0
_________________________________________________________________
```

Fig 3. Network Specifications

Dataset

The model is applied on the AMI Meeting Corpus Dataset. This dataset is a multi-modal data set consisting of 100 hours of meeting recordings. The recordings use a range of signals synchronized to a common timeline. These include close-talking and far-field microphones, individual and room-view video cameras, and output from a slide projector and an electronic whiteboard. The meetings were recorded in English using three different rooms with different acoustic properties, and included mostly non-native speakers. There are a total of four files namely ('ES2003a', 'ES2003b', 'ES2003c', 'ES2003d') which have been used.

The AMI Meeting Corpus includes high quality, manually produced orthographic transcription for each individual speaker, including word-level timings that have been derived by using a speech recognizer in forced alignment mode. It also contains a wide range of other annotations, not just for linguistic phenomena but also detailing behaviors in other modalities. These include dialogue acts; topic segmentation; extractive and abstractive summaries; named entities; the types of head gesture, hand gesture, and gaze direction that are most related to communicative intention; movement around the room; emotional state; and where heads are located on the video frames. The linguistically motivated annotations have been applied the most widely, and cover all of the scenario-based recordings.

Result

Diarization error rate (DER) is the standard metric for evaluating and comparing speaker diarization systems. For this, we have used PyAnnote library's metric module which contains the DER function.

The evaluated DER obtained for the 4 audio files namely - ‘ES2003a’, ‘ES2003b’, ‘ES2003c’ and ‘ES2003d’ of the AMI Corpus Dataset were 30.8%, 36.7%, 14.5% and 29.3% respectively.

By comparing the DER metrics, we can say that the evaluation metrics show that K-means Clustering algorithm gave the highest performance for ‘ES2003c’ audio file where the number of speakers in the Ground Truth and Hypothesis predicted were the same and also achieved the lowest error rate of 14.5%.

‘ES2003d’ audio file had the second best loss metric whereas ‘ES2003b’ and ‘ES2003a’ had close values of the loss metric.

	Speaker_id	Offset	end
0	B	0.280000	3.194354
1	B	3.204354	5.300000
2	B	7.480000	8.024104
3	B	8.034104	10.380000
4	B	11.860000	15.872449
..	...	...	...
488	C	2236.960000	2240.270000
489	C	2241.490000	2243.850000
490	D	2245.630000	2248.390000
491	B	2252.710000	2252.910000
492	B	2254.920000	2256.530000

Fig 5. Speaker Identification

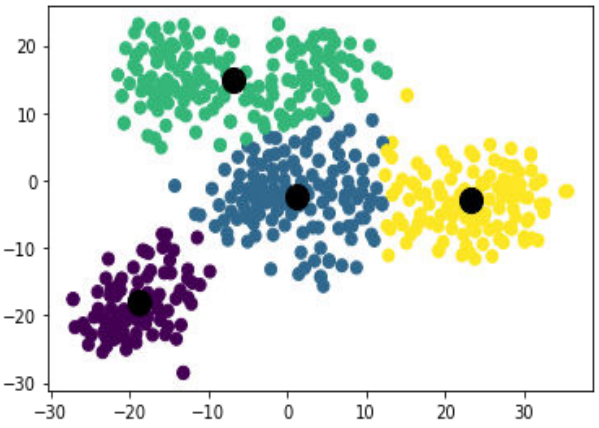

Fig 6. Clustering output using k-means algorithm

4. CONCLUSION

We have developed a speaker change detection approach using bidirectional long short-term memory networks. We took 4 audio files from the AMI Corpus Dataset which were over 2 hours long. We were able to achieve a Diarization Error rate of 14.5% on one of the dataset (ES2003c). Finally, despite major improvements of the speaker change detection module, its impact on the overall speaker diarization system is minor but we were able to achieve an accuracy of 85.5% using our model. We plan to investigate LSTM-based speech turn embeddings like TristouNet to fully benefit from this improved segmentation.

AMI Dataset Audio File	DER
ES2003a	30.8%
ES2003b	36.7%
ES2003c	14.5%
ES2003d	29.3%

Table 1. DER values for AMI Corpus Dataset

5. FUTURE WORK

As the rise of deep learning technology, more and more advancements have been made for speaker diarization, from a method that replaces a single module into a deep-learning-based one, to a fully end-to-end neural diarization. Furthermore, as the speech recognition technology becomes more accessible, a trend to tightly integrate speaker diarization and ASR systems has emerged, such as benefiting from the ASR output to improve speaker diarization accuracy. According to [18], a few remaining challenges for speaker diarization towards future research and development are listed here:

Online processing of speaker diarization. Most speaker diarization methods assume that an entire recording can be observed to execute speaker diarization. However, many applications such as meeting transcription systems or smart agents require only short latency for assigning the speaker.

Speaker overlap. Overlap of multi-talker speech is the inevitable nature of conversation. Nevertheless, many conventional speaker diarization systems, especially clustering-based systems, treated only non overlapped regions of recordings sometimes even for the evaluation metric.

Audio visual modeling. Visual information contains a strong clue to identify speakers. While these studies showed the effectiveness of visual information, the audio-visual speaker diarization has yet been rarely investigated compared with audio-only speaker diarization, and there will be many rooms for improvement.

6. REFERENCES

[1] Quan Wang, Carlton Downey, Li Wan, Philip Andrew, Mansfield Ignacio, Lopez Moreno SPEAKER DIARIZATION WITH LSTM, Dec 2018.

[2] Aonan Zhang, Quan Wang, Zhenyao Zhu, John Paisley, Chong Wang FULLY SUPERVISED SPEAKER DIARIZATION, Feb 2019.

[3] Scott Seyfarth, Sundararajan Srinivasan, Katrin Kirchhoff Speaker-conversation factorial designs for diarization error analysis, Jun 2021.

[4] Tae Jin Parka, Naoyuki Kandab, Dimitrios Dimitriadisb, Kyu J. Hanc, Shinji Watanabed , Shrikanth Narayanan A Review of Speaker Diarization: Recent Advances with Deep Learning , Jan 2021.

[5] Srikanth Raj Chetupalli, Sriram Ganapathy Speaker diarization assisted ASR for multi-speaker conversations, Apr 2021.

[6] Patrick Kenny, Douglas Reynolds, and Fabio Castaldo, "Diarization of telephone conversations using factor analysis," IEEE Journal of Selected Topics in Signal Processing, vol. 4, no. 6, pp. 1059–1070, 2010.

[7] Fabio Castaldo, Daniele Colibro, Emanuele

Dalmasso, Pietro Laface, and Claudio Vair, "Stream-based speaker segmentation using speaker factors and eigenvoices," in Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on. IEEE, 2008, pp. 4133–4136.

[8] Stephen H Shum, Najim Dehak, Reda Dehak, and James R ´ Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 10, pp. 2015– 2028, 2013.

[9] Mohammed Senoussaoui, Patrick Kenny, Themos Stafylakis, and Pierre Dumouchel, "A study of the cosine distance based mean shift for telephone speech diarization," IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), vol. 22, no. 1, pp. 217–227, 2014.

[10] Gregory Sell and Daniel Garcia-Romero, "Speaker diarization with plda i-vector scoring and unsupervised calibration," in Spoken Language Technology Workshop (SLT), 2014 IEEE. IEEE, 2014, pp. 413–417.

[11] Gregory Sell and Daniel Garcia-Romero, "Diarization resegmentation in the factor analysis subspace," in Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. IEEE, 2015, pp. 4794–4798.

[12] Kiran Karra and Alan McCree Speaker Diarization using Two-pass Leave-One-Out Gaussian PLDA, Apr 2021.

[13] Jee-weon Jung, Hee-Soo Heo, Youngki Kwon, Joon Son Chung, Bong-Jin Lee Three-class Overlapped Speech Detection using a Convolutional Recurrent Neural Network, Apr 2021.

[14] Nauman Dawalatabad, Mirco Ravanelli, Francois Grondin, Jenthe Thienpondt, Brecht Desplanques, Hwidong Na ECAPA-TDNN Embeddings for Speaker Diarization, Apr 2021.

[15] Shota Horiguchi, Paola Garc´ıa, Yusuke Fujita, Shinji Watanabe, Kenji Nagamatsu END-TO-END SPEAKER DIARIZATION AS POST-PROCESSING, Dec 2021.

[16] Ulrike Von Luxburg, "A tutorial on spectral clustering," Statistics and computing, vol. 17, no. 4, pp. 395–416, 2007.

[17] Huazhong Ning, Ming Liu, Hao Tang, and Thomas S Huang, "A spectral clustering approach to speaker diarization.," in INTERSPEECH, 2006.

[18] Park, Tae Jin et al. "A Review of Speaker Diarization: Recent Advances with Deep Learning." ArXiv abs/2101.09624 (2021): n. pag.