I. J. Engineering and Manufacturing, **2024, 3, 27-35**
Published Online on June 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijem.2024.03.03

Speaker Diarization Using Bi-LSTM and Spectral Clustering

Trisiladevi C Nagavi*
S. J. College of Engineering, JSS Science and Technology University, Mysore, Karnataka, India
E-mail: trisiladevi@sjce.ac.in
ORCID iD: https://orcid.org/0000-0001-8789-3443
*Corresponding author

Samanvitha Sateesha
S. J. College of Engineering, JSS Science and Technology University, Mysore, Karnataka, India
E-mail: samanvitha.sateesha@gmail.com
ORCID iD: https://orcid.org/0009-0007-9687-662X

Shreya Sudhanva
S. J. College of Engineering, JSS Science and Technology University, Mysore, Karnataka, India
E-mail: shreya.sudhanva@gmail.com
ORCID iD: https://orcid.org/0009-0000-2970-3699

Sukirth Shivakumar
S. J. College of Engineering, JSS Science and Technology University, Mysore, Karnataka, India
E-mail: sukirth.shivakumar304@gmail.com
ORCID iD: https://orcid.org/0009-0006-4770-8921

Vibha Hullur
S. J. College of Engineering, JSS Science and Technology University, Mysore, Karnataka, India
E-mail: hullurvibha@gmail.com
ORCID iD: https://orcid.org/0009-0009-3691-4086

Received: 11 January, 2023; Revised: 20 February, 2023; Accepted: 17 March, 2023; Published: 08 June, 2024

Abstract: Speaker diarization is the ability to compare, recognize, comprehend and segregate different sound waves on the basis of the identity of the speaker. This work aims to accomplish this process by segmenting, embedding and clustering the extracted features from the speech sample. In this work, Mel-Frequency Cepstral Coefficients (MFCC) are extracted and fed into Bi-Directional Long Short-Term Memory (Bi-LSTM) model for segmentation. Then d-vectors are extracted using pre-trained models from pyannote libraries. Spectral Clustering is used to group and segregate the audio of one speaker from another. The experimentation is carried out on two speaker speech audio files and the results indicate that the diarization is successful. The diarization error rate of 9.4% for a 2-speaker audio file is the lowest DER achieved for the given data set. This indicates the efficiency of the system and also justifies the combination of methods chosen at each step. By considering such exciting technical trends, we believe the work presented in the paper represents a valuable contribution for the community by providing the recent developments using Bi-LSTM and spectral clustering methods, which enables the future development towards speaker diarization.

Index Terms: Speaker Diarization, Bi-LSTM, MFCC, Spectral Clustering, Diarization Error Rate

1. Introduction

Being able to distinguish between people proves to be an important task as a human being. Among the five sensory abilities, the prowess to listen to sounds and make sense of them is considered to be the best gift to humankind. The vibrations created to displace air molecules results in the genesis of sound waves. These sounds waves travel through the air medium with a speed of 332 meters per second until it is perceived by the human ear. Each sound wave is

different owing to the change in tone, frequency, amplitude, pitch and various other factors.

This research aims to diarize the audio files when 2-speakers are involved. There are numerous methods to achieve speaker diarization and each of these methods uses one or more of the steps detailed in section 3. The methods also use speaker diarization as a pre-processing step without solely focusing on diarizing the speaker files. This research focuses solely on achieving speaker diarization for a 2-speaker environment as efficiently as possible.

2. Literature Review

The detailed literature review is presented in this section on speaker diarization system. Daniel Garcia-Romero et al. assert the importance of speaker diarization and its role in the front-end for multiple speech technologies [1]. They report that speaker diarization is a necessary front-end framework for plethora of speech applications especially when multiple speakers are involved, but the existing methods that use i-vector clustering for short segments of speech are expensive and even tough to handle. In this work, the authors aim to solve this problem by eliminating the i-vector extraction process in the pipeline. They achieve this by using a deep neural network for learning representations, which helps to learn a fixed-dimensional embedding for acoustic segments of variable length. They also use a scoring function to measure the likelihood of the origin of segments from the speaker. Further they make use of the Deep Neural Network (DNN) clustered segments to initialize the VB re-segmentation to produce best results. While this approach provides a remarkable result, it is limited to supervised calibration technique as used by the authors.

Quan Wang et al. developed a novel d-vector based approach to speaker diarization [2]. They do this on the basis of d-vector based speaker verification systems. The authors developed a state-of-the-art speaker diarization system by combining LSTM based d-vector embeddings and non-parametric clustering. They have evaluated their model on NIST SRE 2000 CALLHOME datasets and have achieved a Diarization Error Rate (DER) of 12.0%. They have also obtained out-of-domain data from voice search logs and used it for training the model. The experiment was conducted on four clustering algorithms Naive online clustering, Links online clustering, K-Means offline clustering and Spectral offline. Further clustering algorithm is combined with both i-vectors and d-vectors Long Short-Term Memory (LSTM Networks), and reported the performance on three standard public datasets. The results reflected that all three datasets provided the best results with the d-vector based systems. The Voice Activity Detection (VAD) is done using a small Gaussian mixture model using two full covariance Gaussians.

The system proposed by Weiqing Wang et al. consists of 5 models for the task of diarization [3]. The first model is a VAD model followed by a speaker embedding model. The next models are for clustering, Overlapped Speech Detection (OSD) and Target-Speaker Voice Activity Detection (TS-VAD). The clustering model comprises of two systems based on clustering for speaker diarization with different similarity measures and OSD model is in turn composed of two different models. Their final submission contained 5 independent systems and achieved a DER of 5.07% on the test set. The method that obtained lowest DER was found to be Agglomerative Hierarchical Clustering (AHC) during the first submission. They further bettered their results by fusing multiple systems. While the authors achieve an impressive diarization error rate, the results can be improved by tuning the weights less aggressively and avoid over fitting.

Trisiladevi C Nagavi et al. performed content-based audio retrieval on the basis of acoustic similarity [4]. They employ Sort-Merge technique to realize their goal. They extracted MFCC frequency features of the audio streams. Statistical tools such as mean and Euclidean distance were used as well. The system developed by the authors consists of two steps. In the first step, all the clusters with identical high energy components are identified and merged. The audio files in all the merged clusters are then sorted according to their distances. The accuracy obtained is a little over 80%. However, the time complexity is observed to be significant. They provide a solution to encounter this problem. The solution provided helps to improve the time complexity however it is not immune to inaccuracies. Furthermore, the authors suggest the use of efficient indexing technique.

Federico Landini et al. have analyzed the diarization system developed by the BUT team for the Vox converse challenge [5]. Their diarization system comprised of 7 steps. They began their work by signal pre-processing, voice activity detection, embedding extraction and in clustering they followed three steps. Firstly, their system contained initial clustering of the embeddings using agglomerative hierarchical clustering followed by clustering with Bayes Hidden Markov Model. The sixth step was global speaker embedding reclustering followed by the identification of overlapped speech signals. They used two metrics - DER and Jaccard Error Rate (JER) to evaluate their results. They achieved DER of 4% and JER of 19.80 on development set and DER of 8.12% and JER of 18.35 on the evaluation set.

Huanru Henry Mao et al. proposed to combine Automatic Speech Recognition and Speaker Diarization models to perform diarization on long conversations and enhance word diarization error-rate [6]. They have introduced a striding attention decoding algorithm to handle lengthy conversations with unknown utterances. They collected hour long podcast episodes from the weekly "This American Life" and used it as their dataset. They concluded that when known utterances are absent, models that jointly learnt automatic speech recognition and speaker diarization performed better. Models that separately learnt speaker diarization performed better when known utterances were present.

Latane Bullock et al. have addressed the problem of overlapping speech in a diarization system [7]. They use LSTM based architecture for overlap detection. Following this, in the resegmentation step they use these overlap detections to assign two-speakers. They do this with the aid of speaker posterior matrix. They have performed their

experiment on AMI, ETAPE and DIHARD datasets. They were able to relatively reduce DER by 20%.

3. Methodology

The speaker diarization model comprises of numerous steps as shown in Fig 1. Following sections discusses each step-in detail.

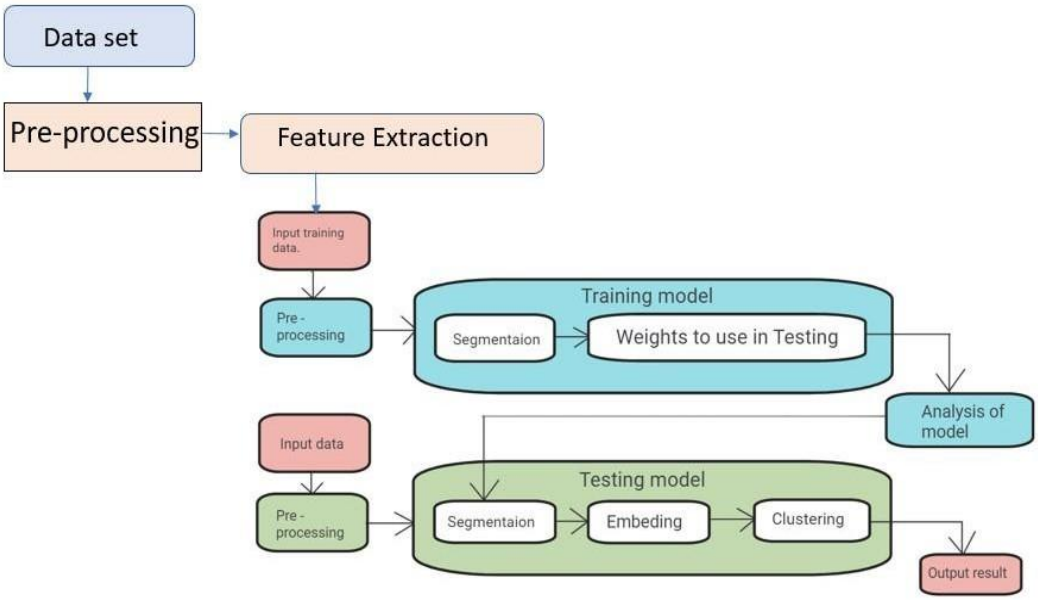

Fig. 1. Speaker Diarization Model

3.1 Data set

The data set used for this work is provided by VoxConverse [8]. It consists of multi-speaker clips of human speech extracted from YouTube. The data set contains two sets namely dev and test. There are 216 files on the dev set and 232 files on the test set. Each set is provided with annotations as Rich Transcription Time Marked (RTTM) files.

Each RTTM file contains multiple lines each containing ten fields. They are type, file ID, channel ID, turn on set, turn duration, orthography field, speaker type, speaker name, confidence score and signal lookahead time**.**

The research work aims to collect datasets from various sources that may be publicly available. After various experimentation and considering the factors such as a) number of speakers b) number of audio channels c) noise in the dataset d) labels available for the dataset, the authors decided to proceed with the dataset provided by VoxConverse.

3.2 Pre-processing

Data preprocessing is the process of processing the data to remove all redundant information before it is fed into any system. For textual data, it may be the removal of stop words where as for speech data, it is the removal of noise. The data for this work was pre-processed using webrtcvad. All white spaces and noise was trimmed out. The cleaned audio file contained only relevant information that is used for effective diarization. Another significant part of data preprocessing is normalizing of the audio files to establish a uniform sampling rate. Furthermore, there may be instances where the audio file may have multiple channels. In order to enhance the speech signal and normalize the direction of the speech signal, data preprocessing is vital.

3.3 Feature Extraction

Feature extraction is the process of acquiring various features such as power, pitch and vocal tract configuration from the audio signal. Front-end signal processing refers to transforming a 3 speech waveform to a form of parametric representation at a relatively lesser data rate for subsequent processing. Feature extraction approaches usually yield a multi-dimensional feature vector for every speech signal. There are a wide range of options to parametrically represent the speech signal. A few of them are - Perceptual Linear Prediction (PLP), MFCC, Linear Prediction Coefficients (LPC) and Linear Prediction Cepstral Coefficients (LPCC) [9].

MFCCs are one of the popular features [10] and their extraction process is a replication of the human hearing system as it intends to artificially implement the ears' working principle. This technique gives a discrete cosine transform of a real logarithm of the short-term energy displayed on the Mel Frequency Scale.

In the algorithm 1, a logarithmic transformation of a signal's frequency is the Mel Scale. The underlying notion behind this transformation is that sounds of equal distance on the Mel Scale are perceived by humans as being of same distance. Mel Spectrograms are spectrograms that illustrate how sounds on the Mel scale are represented. Procedure to

obtain MFCC features is shown in Fig 2. To develop the MFCC features, the audio is first converted from Hertz to Mel Scale, the logarithm of Mel representation of audio is taken, logarithmic magnitude is taken and Discrete Cosine Transformation is used. As a result, a spectrum is created across Mel frequencies rather than time, resulting in MFCCs.

Algorithm 1: Procedure for MFCC

Input: Sample Rate, Hop Length, Frame Size
Output: Matrix of MFCCs by frame
/***(Procedure for MFCC Extraction)***/
1. Compute Fourier Time Transform (FTT) power spectrum.
2. Compute Mel-frequency m-channel filter bank.
3. Convert to ceptra via Discrete Cosine Transform (DCT).
4. Return Matrix.

$$f(i) = floor\big((nfft + 1) * h(i)/sample\,rate\big) \tag{1}$$

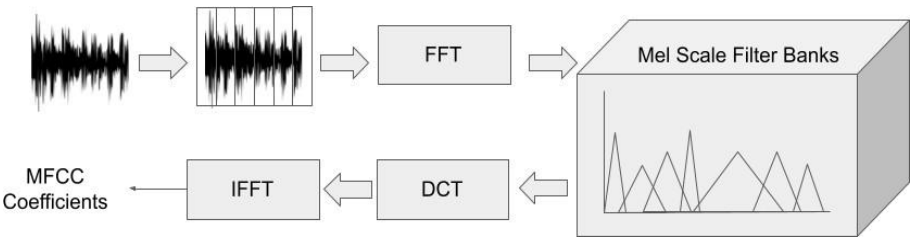

Fig. 2. Procedure to obtain MFCC Features.

3.4 Segmentation

Speaker segmentation is an important sub-problem of the field of speech recognition. Speaker segmentation is the process of identifying the boundaries between words, syllables, or phonemes in spoken natural languages. It aims to identify the exact location of the change of speaker. Therefore, this constitutes an important step in the process of diarization. Speaker segmentation is achieved by using numerous deep learning models. A sample output of speaker segmentation is shown in Fig 3.

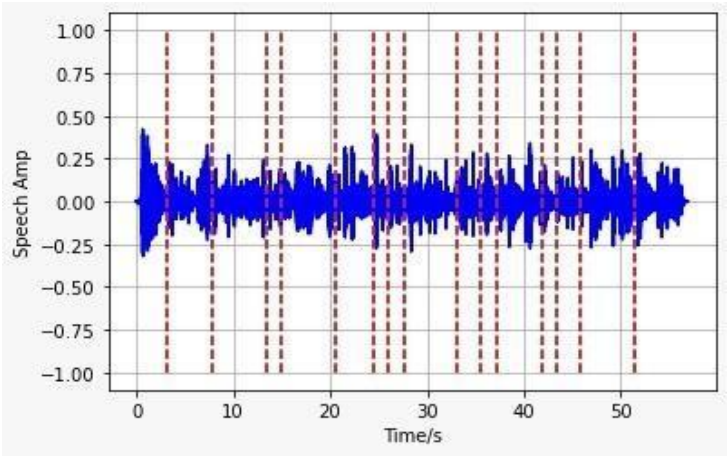

Fig. 3. Segmentation of Sample Speech File

3.5 Embedding Extraction

Embedding Extraction refers to the process of understanding and differentiating the identity of speakers. The extracted embeddings are represented in the form of a vector. The vector usually denotes a compact representation of speech data. There are different types of vectors. They are i-vectors, d-vectors and x-vectors depending on the algorithm used. There are multiple algorithms and pre-trained models available for embedding extraction. To extract d-vectors, the pre-trained models of the pyannote libraries are used.

3.6 Clustering

Exploratory data analysis techniques are often employed to understand the structure of the data. Clustering is one of the common techniques employed to obtain insights about the data. It can be defined as the process of grouping data into various subgroups such that all the data points belonging to one subgroup contains high degree of homogeneity. On the other hand, data points belonging to different clusters are very different. Each subgroup is called a cluster. In other words, homogeneous clusters are found within the data such that data points in each cluster are as similar as possible according to a similarity measure such as Euclidean-based distance or correlation-based distance. The decision of which similarity measure to use is application-specific.

$$Euclidean\ Distance\ d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{2}$$

3.7 Feature Extraction

The speech diarization system was trained to identify different speakers in an audio file as well as specify the timestamp and duration of the speech. This was achieved by extracting MFCCs of audio files and passing these features for segmentation. The segmentation of the audio files was completed using the Bi-LSTM model [11]. The segmented features were trained to extract embeddings using a pre-trained model and the output was fed as input to spectral clustering.

The MFCC features are calculated after pre-processing. Mel scale is used to map the frequency perceived by human ear to the actual frequency. Cepstrum is calculated using the formula,

$$\big(x(t)\big) = F^{-1}[log(F[x(t))] \tag{3}$$

Here, $x(t))$ represents the time domain signal.$F[x(t)$ is the spectrum. and $C[x(t)$ is regarded as the cepstrum.
The mapping to the Mel-Scale is done using the formula :

$$\text{mel (f)} = 1127ln\left(1 + \frac{1}{700}\right) \tag{4}$$

A Bi-LSTM model is shown in Fig 4.

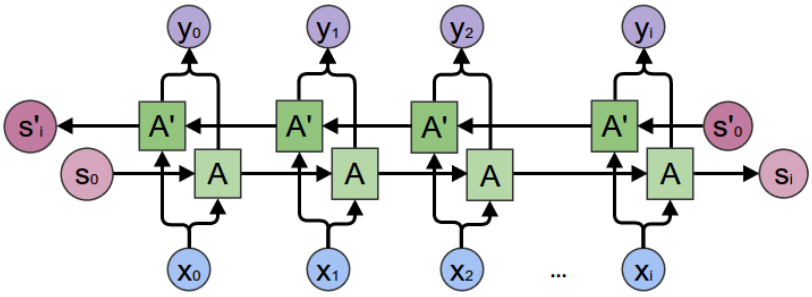

Fig. 4. Bi-directional LSTM

The architecture of a Bi-LSTM is as shown in the Fig 5. It consists of two independent LSTM cells stacked together so that there is forward as well as backward information at every step. A basic LSTM cell consists of 3 gates [12]. They are input gate, forward gate and output gate. The equations of these gates are shown below:

$$\begin{aligned} i_t &= \sigma w_i[h_{t-1}, x_t] + b_i \\ f_t &= \sigma w_f[h_{t-1}, x_t] + b_f \\ o_t &= \sigma w_o[h_{t-1}, x_t] + b_o \end{aligned}$$

i_t : represents input gate f_t : represents forget gateo_t: represents output gate
σ: represents sigmoid function
w_x : weight for the respective gate(x) neurons
h_{t-1}: output of the previous LSTM block (at timestamp t-1)x_t: input at current timestamp
b_x: biases for the respective gates(x)

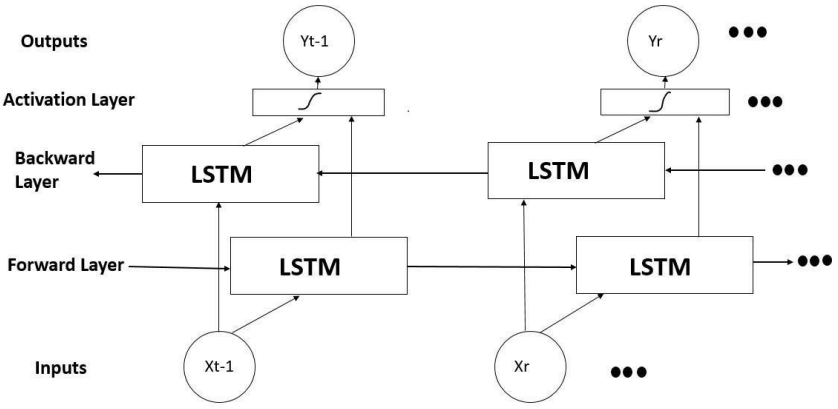

Fig. 5. Architecture of Bi-LSTM

To generate the new candidate value, the following formula is used :

$$C^{Nt} = tanh(W_c[a^{t-1}, x^t] + b_c) \tag{5}$$

The two outputs from the LSTM model are :

$$C^t = \Gamma * C^{Nt} + \Gamma f * C^{t-1} \tag{6}$$

$$a^t = \Gamma o * C^t \tag{7}$$

To extract embeddings pre-trained models from pyannote libraries were used and d-vectors were extracted [13, 14]. Spectral Clustering algorithm is a type of clustering algorithm where each data point is treated as a graph node [15, 16].

Algorithm 2: Spectral Clustering

Build the similarity graph in the form of adjacency matrix, A.
Project the data onto a lower dimension space by computing Graph - Laplacian Matrix, L.

$$d_i = \sum_{j=1,i,}^{n} E^{wi} \tag{8}$$

where, d_i is the degree of the ith node.

W_{ij} is the edge between the nodes i and j as defined in the adjacency matrix.

$$D_{ij} = i = j_0 , \; i \neq j \tag{9}$$

where, D_{ij} is the Degree Matrix.

Performing Clustering using K – Means clustering.

So, by using the algorithm firstly we create an undirected graph and n observations in the data. This can be represented by an adjacency matrix which has the similarity between each vertex as its elements. In the next step, the data is computed onto a lower dimension space by Laplacian Matrix. The whole purpose of computing the Laplacian Matrix is to find Eigen values and Eigen vectors in order to embed the data points into a low-dimensional space. In the last step weperform clustering using Eigen values and Eigenvectors using K – Means algorithm.

4. Experimenting Results

4.1 Evaluation Metrics

In this work, three evaluation metrics, namely diarization error rate, purity and coverage have been used [17].

Diarization Error Rate: It is a statistic used to assess speaker diarization systems' performance. It is calculated as the percentage of time that is incorrectly assigned to a speaker or non-speech.

Where, false alarms are the segments considered as speech in hypothesis and not in reference.

Missed detection are the segments considered as speech in reference and not in hypothesis. Confusion is the length of segments assigned to different speakers in reference and hypothesis.Total is the length of the reference.

$$DER = \frac{False\ Alarm + Missed\ Detection + Confusion}{Tatal} \tag{10}$$

Purity: The degree to which a set of records belongs to the same class is measured by purity. Here, S denotes Speaker and C denote Cluster.

$$Puritv = \frac{\sum C \max S|S\beta - C|}{\sum C|C|} \tag{11}$$

Coverage: It is the amount of data for which the model makes a prediction. Here, S denotes Speaker and C denote Cluster.

$$Coverage = \frac{\sum S \max C|S\beta - C|}{\sum S|S|} \tag{12}$$

4.2 Performance Evaluation

Table 1. Results In %

Metric	Value
Diarization Error Rate	9.4
Purity	93.0
Coverage	93.0

The audio files were subjected to pre-processing, feature extraction, segmentation, embedding and clustering for diarization. Methods such as Bi-LSTM were used for segmentation while spectral clustering was used for the clustering step. In this system, it is possible to diarize every single file and obtain it's annotation. After diarizing the files it can be observed from table 1 that the lowest diarization error rate for a VoxConverse audio files is found to be 9.4%, purity and coverage is both 93%. The output of the segmentation is shown in Fig 6. To visualize segmentation, the amplitude of speech waves was plotted against time to identify significant differences between the speakers. The dataset used in this work also contained ground truth in the form of RTTM files. Each file contained speaker id, turn off-set and the duration of the speech segment.

The ground truth was plotted for better visualization assigning a unique color to each speaker. After the completion of the experiment, an output file similar to RTTM file was generated to compare the resulting hypothesis with the ground truth. The resulting hypothesis was plotted in a similar fashion. It was found that the resulting hypothesis with the color blue and the ground truth with the color yellow had very minute differences as can be seen from Fig 7 and Fig 8.

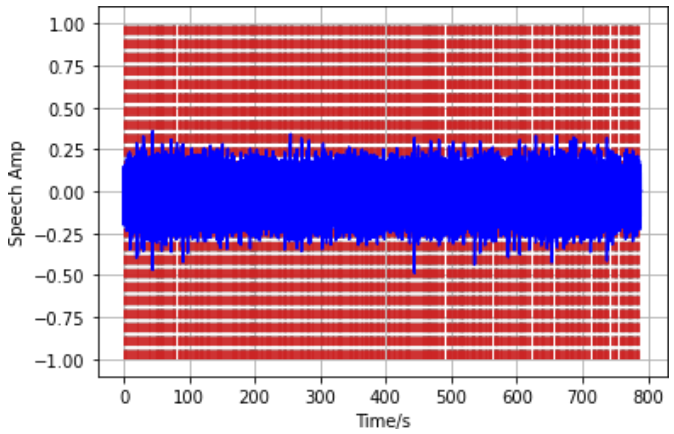

Fig. 6. Output of Segmentation

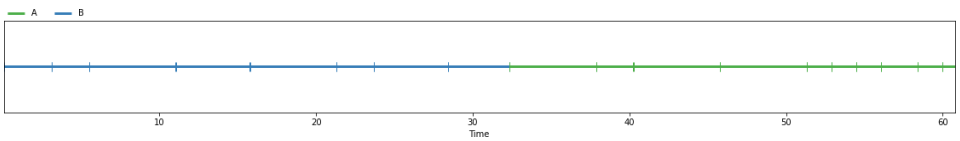

Fig .7. Resulting Hypothesis

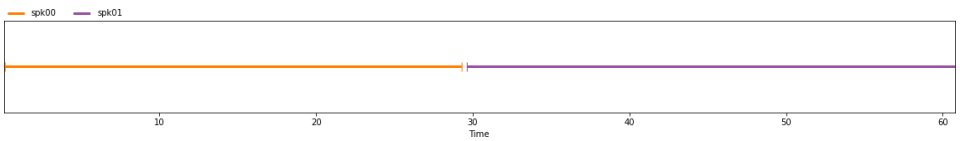

Fig. 8. Ground Truth

5. Conclusion

Speaker diarization is the process of addressing the question "who spoke when?" Speaker diarization has numerous applications in various fields of computer science and day to day life. Various techniques to pre-process the audio files, extract features, segment, embed and cluster the data were employed. The audio files were successfully been diarized with a diarization error rate of 9.4%. This work can be improved by using deep learning methods to classify and recognize the speakers.

Acknowledgment

The authors would like to extend sincere thanks to the management of Sri Jayachamarajendra College of Engineering, JSS Science and Technology University, Mysuru, India for their whole-hearted support throughout the course of the research work. Furthermore, the authors would like to thank VoxConverse for providing the dataset for this work. Finally, the authors would like to thank their family and friends for their constant support throughout the duration of thestudy.

References

[1] 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). IEEE, 2017, pp. 4930–4934.
[2] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, “Speaker diarization with LSTM,” in 2018 IEEE International conference on acoustics, speech and signal processing (ICASSP). IEEE, 2018, pp. 5239–5243.
[3] W. Wang, D. Cai, Q. Lin, L. Yang, J. Wang, J. Wang, and M. Li, “The dku-dukeece-lenovo system for the diarization task of the 2021 voxceleb speaker recognition challenge,” arXiv preprint arXiv: 2109.02002, 2021.
[4] T. Nagavi, S. Anusha, P. Monisha, and S. Poornima, “Content based audio retrieval with MFCC feature extraction, clustering and sort-merge techniques,” 07 2013, pp. 1–6.
[5] F. Landini, O. Glembek, P. Matˇejka, J. Rohdin, L. Burget, M. Diez, and A. Silnova, “Analysis of the but diarization system for voxconverse challenge,” in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 5819–5823.
[6] H. H. Mao, S. Li, J. McAuley, and G. Cottrell, “Speech recognition and multi-speaker diarization of long conversations,” arXiv preprint arXiv: 2005.08072, 2020.
[7] L. Bullock, H. Bredin, and L. P. Garcia-Perera, “Overlap-aware diarization: Resegmentation using neural end-to-end overlapped speech detection,” in ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 7114–7118.
[8] J. S. Chung, J. Huh, A. Nagrani, T. Afouras, and A. Zisserman, “Spot the conversation: speaker diarization in the wild,” arXiv preprint arXiv: 2007.01216, 2020.
[9] U. Shrawankar and V. M. Thakare, “Techniques for feature extraction in speech recognition system: A comparative study,” arXiv preprint arXiv: 1305.1145, 2013.
[10] T. Ganchev, N. Fakotakis, and G. Kokkinakis, “Comparative evaluation of various mfcc implementations on the speaker verification task,” in Proceedings of the SPECOM, vol. 1, no. 2005, 2005, pp. 191–194.
[11] M. Schuster and K. Paliwal, “Bidirectional recurrent neural networks,” Signal Processing, IEEE Transactions on, vol. 45, pp. 2673 – 2681, 12 1997.
[12] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[13] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, “pyannote.audio: neural building blocks for speaker diarization,” in ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing, 2020.
[14] H. Bredin and A. Laurent, “End-to-end speaker segmentation for overlap-aware resegmentation,” in Proc. Interspeech 2021, 2021.
[15] U. Von Luxburg, “A tutorial on spectral clustering,” Statistics and computing, vol. 17, no. 4, pp. 395–416, 2007.
[16] D. Verma and M. Meila, “A comparison of spectral clustering algorithms,” University of Washington Tech Rep UWCSE030501, vol. 1, pp. 1–18, 2003.
[17] O. Galibert, “Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech.” in INTERSPEECH, 2013, pp. 1131–1134.

Authors’ Profiles

Dr. Trisiladevi C. Nagavi is an assistant professor in the Department of Computer Science & Engineering, Sri Jayachamarajendra College of Engineering, JSS Science and Technology University, Mysuru. She graduated from Karnataka University, Dharwad. She obtained her Master’s and Doctoral degree from Visvesvaraya Technological University, Belgaum. Also, she has secured “SECOND RANK” in M.Tech Software Engineering from VTU Belgaum. She has expertise in the area of Audio, Music, Speech and Image Signal Processing, Information Retrieval and Machine Learning. Her research outcome resulted in an Android application to play favorite tunes. It is designed to retrieve songs by listening to user hum based on music melody representation models. Dr. Nagavi is an active member of IEEE India Special Interest Group on Communications Disability and

worked on Real Time Text to Speech Conversion and Translation System in Collaboration with All India Institute of Speech and Hearing (AIISH) Mysuru and IEEE Standards Association. Dr. Nagavi has published research papers at national and international journals, conference proceedings as well as chapters of books.

Samanvitha Sateesha is a graduate student pursuing her masters in Computer Science at the University of California, San Diego. She graduated from Sri Jayachamarajendra College of Engineering, JSS Science and Technology University, Mysuru in 2022. During her undergrad, she initially worked on full-stack development and designed websites for various clubs in her college. She also assisted a team to revamp a website during her first internship and managed a software development project. Her primary research is in areas related to artificial intelligence and machine learning. She has worked on text, image and speech data. Her major projects include identifying and classifying depression using textual data and MRI images, extracting information from resumes and matching them with a job description, etc. Ms. Samanvitha has co-authored a paper titled "Naive Bayes Classifier for depression detection using text data" at ICEECCOT-2021. She has interned at Blueyonder, Reap Benefit and Sitex Digitech. She aims to leverage technology to solve problems in society. She believes that true learning happens when one steps out of comfort zone and is willing to embrace the unknown. Ms. Samanvitha has also served as a student coordinator and SPOC of the Institution's Innovation Council, JSSSTU.

Shreya Sudhanva is a graduate student pursuing her masters in Computer Science at Northeastern University, Boston. She graduated from Sri Jayachamarajendra College of Engineering, JSS Science and Technology University, Mysuru in 2022. She has interned as a developer at companies like Toshiba Software India Pvt Ltd, Reap Benefit (NGO), Mysuru Consulting Group and ExcelSoft Technologies. She has experience in the areas of machine learning and artificial intelligence. She has worked on natural language processing and computer vision related projects like depression detection using textual social media data as well as MRI images, object detection using YOLOv5, twitter sentiment analysis, automatic resume filtering and matching system etc. She has also co-authored a paper based on her findings in the depression detection project at ICEECCOT-2021. Ms. Shreya likes taking up new ventures and building projects that she hopes will make a difference in the world, tackling one problem at a time. She has also served as the secretary at Linux Campus Club, Sri Jayachamarajendra College of Engineering, Mysuru for the year 2021-22. Furthermore, she had been elected as one of the placement secretaries of her class which allowed her to be an interface between her peers and the companies. She enjoys taking up responsibilityand keeping herself busy.

Sukirth Shivakumar is a graduate student pursuing his masters in Computer Science at Northeastern University, Boston. He graduated from Sri Jayachamarajendra College of Engineering, JSS Science and Technology University, Mysuru in 2022. He is interested in game design, virtual and augmented reality and application development. He has worked as a team lead at STEP-SJCE for a mobile application project. He has also worked as a developer in the field of virtual and augmented reality at Celestasi Technologies. His recent work was an internship at Hewlett Packard Enterprise where he was part of the UI development team of Networking product. An extrovert by heart is always good with working in teams. Mr. Sukirth was the anchor for the colleges annual fest JAYCIANA. He is also known to be adventurous and often likes to go on treks with his friends. He had been a part of Sahas, the adventure club of JSS Science and Technology University since 2018.

Vibha Hullur is a software engineer at Aris Global. She graduated from Sri Jayachamarajendra College of Engineering, JSS Science and Technology University, Mysuru in 2022. She has experience in the areas of Machine Learning and Web development. During undergrad, she worked on various projects like Automated Telephone Customer system, Twitter sentiment analysis, Digit recognition, and Driver drowsiness detection etc., Her recent work was an internship at SJCE-STEP where she was part of the front-end development team and built a website named "JEEVAN BINDU". She likes taking up new ventures and building projects that she hopes will make a difference in the world, tackling one problem at a time. Ms. Vibha had been a part of Make A Difference (MAD) and worked as a Fundraiser contributing to society. She has also served as a Creative Lead at Linux Campus Club, Sri Jayachamarajendra College of Engineering, Mysuru for the year 2021-22.

How to cite this paper: Trisiladevi C Nagavi, Samanvitha Sateesha, Shreya Sudhanva, Sukirth Shivakumar, Vibha Hullur, "Speaker Diarization Using Bi-LSTM and Spectral Clustering", International Journal of Engineering and Manufacturing (IJEM), Vol.14, No.3, pp. 27-35, 2024. DOI:10.5815/ijem.2024.03.03