Speaker Diarization

Ms. Apoorva Iyer , Ms. Deepika Kini , Mrs. Shanthi Therese (Associate Professor)
Dept. of Information Technology Thadomal Shahani Engineering College
Mumbai, India

Abstract *— Speaker Diarization is the task of determining 'who spoke when?'.Speaker Diarization uses unsupervised as well as supervised approaches to detect the change of speaker in the temporal dimension. This paper primarily describes the implementation of Speaker Diarization using Neural Networks (a supervised method). First a summary of the clustering algorithms is given. Then the three approaches using neural networks is specified. They are Speaker Diarization using Artificial Neural Networks, Recurrent Neural Networks and Adaptive Long Short Term Memory or Multiple LSTMs. Finally the accuracy is calculated and the results are compared.*

Keywords *— Artificial Neural Network, Recurrent Neural Networks, LSTM, MFCC*

I. INTRODUCTION

Speaker Diarization is the task of determining 'who spoke when?'. Speaker diarization is the process of partitioning an input audio stream into homogeneous segments according to the speaker identity. It can enhance the readability of an automatic speech transcription by structuring the audio stream into speaker turns and, when used together with speaker recognition systems, by providing the speaker's true identity.[1] With the rise of voice biometrics and speech recognition systems, the ability to process audio of multiple speakers is crucial. In many applications, we will want to identify multiple speakers in a conversation, for example when writing a protocol of a meeting. For such occasions, identifying the different speakers and connect different sentences under the same speaker is a critical task.[2]

Speaker diarization has utility in a majority of applications related to audio and/or video document processing, such as information retrieval for example. Indeed, it is often the case that audio and/or video recordings contain more than one active speaker. This is the case for telephone conversations (for example stemming from call centers), broadcast news, debates, shows, movies, meetings, domain-specific videos (such as surgery operations for instance) or even lecture or conference recordings including multiple speakers or questions/answers sessions.[2]

Speaker diarization is one of the tasks in the NIST Rich Transcription (RT) Meeting Recognition Evaluation. It is to automatically find the segments of time within a meeting in which each meeting participant is talking, a task to detect Who Spoke This requires for marking the start and end times of every speech segment with a speaker identity, from a continuous audio recording of a meeting. In recent years, there has been extensive research on the speaker diarization systems[3]

Audio diarization is defined as the task of marking and categorizing the different audio sources within an unmarked audio sequence. On the flip side, owing to its lack of search ability, working on audio data is a tedious task. [4]

II. THE PREVIOUS APPROACHES

The goal of a speaker diarization system is to analyse an audio stream and output a set of labels defining the moments when each individual speaker speaks. This can be cast as a classification task, if all the speakers along with their identities are known beforehand.[6] Speaker diarization is the process of detecting the turns in speech because of the changing of speaker and clustering the speech from the same speaker together, and thus provides useful information for the structuring and indexing of the audio document.[7] The first step in any automatic speech recognition system is to extract features i.e. identify the components of the audio signal that are good for identifying the linguistic content and discarding all the other stuff which carries information like background noise, emotion etc.[5] Most of present state-of-the-art speaker diarization systems fit into one of two categories: the bottom-up and the top-down approaches, as illustrated in Fig 1[2]

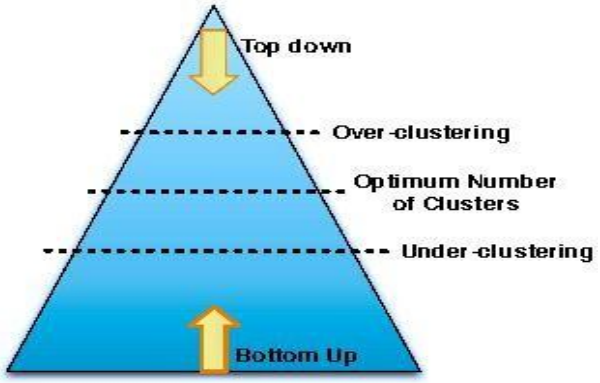


Fig. 1 Top down and bottom up Approach

A. The bottom-up Approach

The bottom-up approach is by far the most common in the literature. Also known as agglomerative hierarchical clustering (AHC or AGHC), the bottom-up approach trains a number of clusters or models and aims at successively merging and reducing the number of clusters until only one remains for each speaker. Clusters are generally modeled with a GMM and, upon merging, a single new GMM is trained on the data that was previously assigned to the two individual clusters.

This simple approach generally leads to good performance. In all cases the audio stream is initially over-segmented into a number of segments which exceeds the anticipated maximum number of speakers.

B. The Top-down Approach

In contrast with the previous approach, the top-down approach first models the entire audio stream with a single speaker model and successively adds new models to it until the full number of speakers are deemed to be accounted for. A single GMM model is trained on all the speech segments available, all of which are marked as unlabeled. Using some selection procedure to identify suitable training data from the non-labeled segments, new speaker models are iteratively added to the model one-by-one.

Top-down approaches however are far less popular than their bottom-up counterparts. Whilst they are generally outperformed by the best bottom-up systems, top-down approaches have performed consistently and respectably well against the broader field of other bottom-up entries.

Top-down approaches are also extremely computationally efficient and can be improved through cluster purification.

III. MAIN APPROACHES

The task of Speaker Diarization has been performed using Neural Networks. There are few assumptions for the audio files on which the algorithms are applied. They are as follows

- The number of speakers in the respective audio file is known.
- The audio file consists of no noise. Hence the algorithms do not tackle with noise removal.
- The speakers speak in a sequential manner that is one after the other. This implies the audio files have no overlapping speech.

Before the algorithms are applied the audio files must pre-processed. The data pre-processing step and the 3 major approaches using Neural Networks (which are Speaker diarization using the Artificial Neural Network, the Recurrent Neural Network and the Adaptive LSTMs.) are explained below.

A. Data Pre-processing

The dataset is imported as shown in figure 2. The dataset used are .wav audio files with a sampling rate of 16000 Hz. The dataset has 3 or 2 speakers. The speakers speak sequentially, that is there is no overlapping. The dataset is divided into training and testing files.

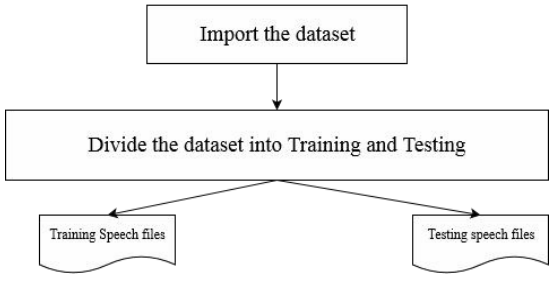


Fig. 2. Dataset divided

The audio files are further segmented into either 2 second or 0.1 second .wav files depending on the neural network used. Features are extracted from each audio segment of 2 second or 0.1 second. The feature used is the Mel frequency Cepsrtal Coefficients (MFCC)

Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC. They are derived from a type of cepstral representation of the audio clip (a nonlinear "spectrum-of-a-spectrum"). The difference between the cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum.[8] The approaches are described in the following sections. Figure 3 shows the steps carried out to extract the MFCC.

B. Artificial Neural Network

The steps are specified in the figure 4. First, the training data is prepared. It is segmented into 2 seconds segments using a python code. Every 2 seconds may have one or 2 speakers speaking sequentially. Then, the 13 mel frequency cepstrum coefficients (MFCC) are calculated for every 2 seconds. There are 199 rows of 13 mfccs. These 199 rows of mfccs are averaged, stored in a.csv file and used to train the network. Once the network is trained, the weights are stored in .hdfs file so that it can be reused when testing is done.The testing data is prepared in way similar to the training data. The coefficients are stored in a .csv file. The weights are extracted from the .hdfs file and the network is compiled.

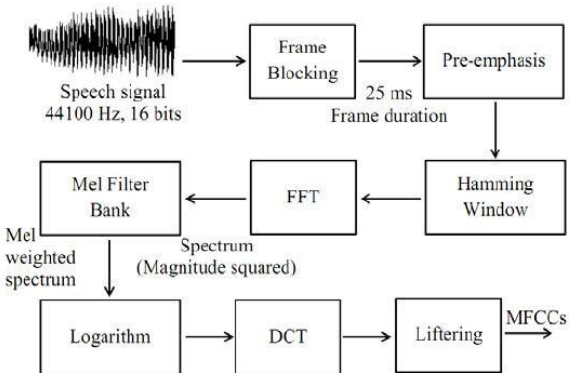


Fig. 3. Steps for MFCC Extraction

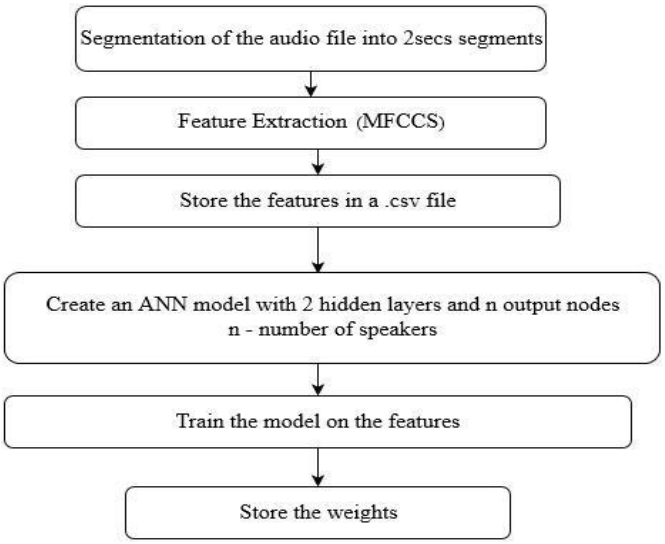


Fig. 4. Training for ANN

Now, the speakers are predicted for every 2 secs by the

network. These predicted values are stored and compared with the real test values. Finally the accuracy is calculated. The formula and process of calculating accuracy is explained further.

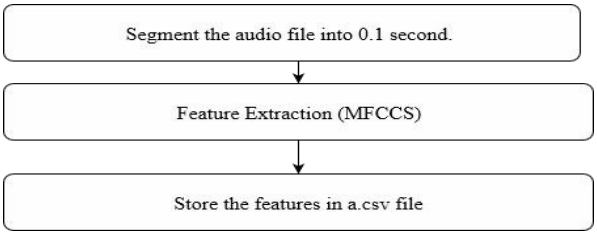


Fig. 5. Data Preprocessing for Single RNN (LSTM)

C. Recurrent Neural Network (LSTM)

Recurrent Neural Networks (RNN) are a powerful and robust type of neural networks and belong to the most promising algorithms out there at the moment because they are the only ones with an internal memory.

Figure 5 shows the data preprocessing steps of LSTM (Long Short Term Memory) networks. For the RNN network the data is divided into segments of 0.1 seconds. Then the features are extracted. These features being the 13 MFCC coefficients. Then they are stored in a .csv file along with their classification label

An LSTM network is created. This network will be trained on each 0.1 segment. In the LSTM, the mel cepstrum coefficients are given. That is the audio of 0.1 second will be trained on all the mfccs generated 9 frames each having 13 mfcc coefficients the result is predicted. Figure 6 shows the entire training process of the Single RNN LSTM network.

Figure 7 shows the testing process in which the data structure is created and loaded into the network for prediction later it is compared with real values to calculate the accuracy.

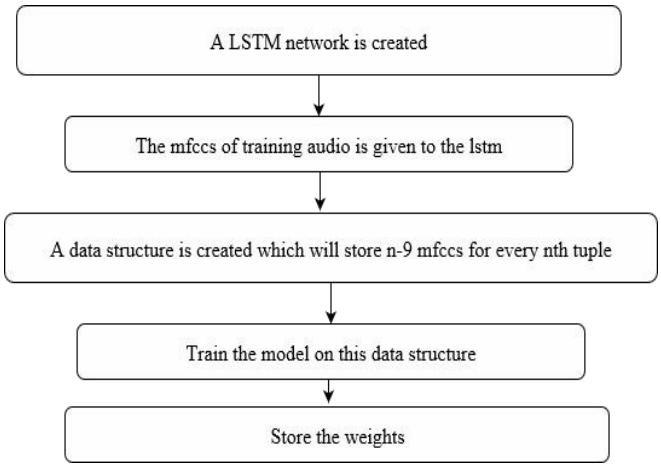
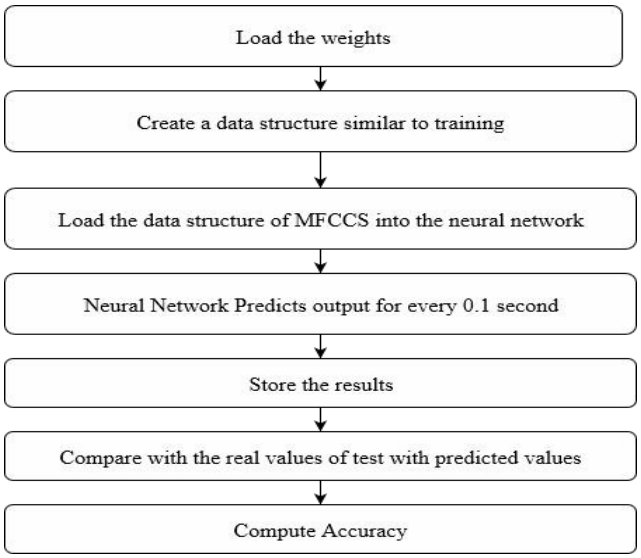


Fig. 6. Training the RNN LSTM model

D. Adaptive LSTMs

The Adaptive LSTM process uses one LSTM network for each speaker.The dataset will be segmented in 0.1 second segments. Then the features of this segment are extracted (MFCC) and stored in a.csv file. This is similar to the data preprocessing in the RNN method. A network is built for individual speakers and then a time step of 9 is used. It uses 2 hidden layers. Both the speakers are trained on positive and negative segments. Then, we store the weights for both the networks in an .hdfs file.

Fig. 7. Testing the RNN (LSTM) model



The testing data is the same for both the speakers. So we test the a particular segment on both the networks. Both the networks give a probability. The output probabilities of both the networks are compared.

The segment would belong to the network whose output node gives the higher probability. Figure 9 shows the Testing process of an Adaptive LSTM.

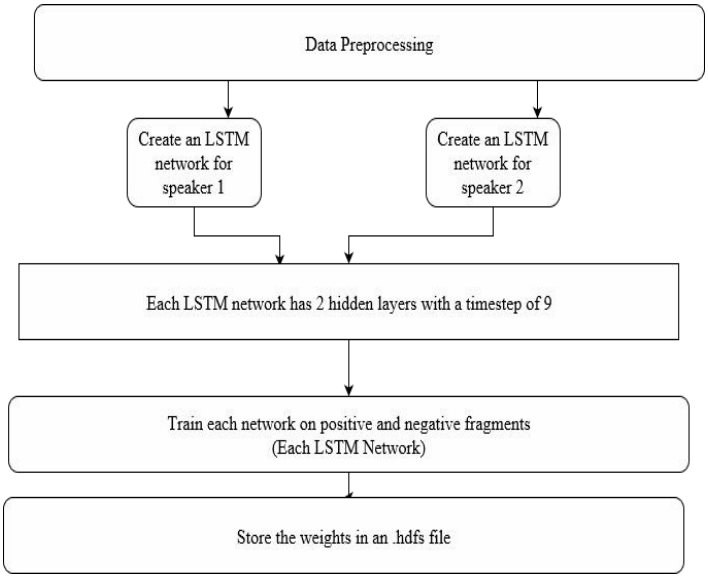


Fig. 8. shows the training process of the Adaptive LSTM

IV. RESULTS

The Accuracy of the approaches is calculated using the following formula.

Accuracy = Number of segments predicted correctly*100% Total Number of segments

A. ANN for 3 speakers

The data was segmented at 2 seconds. Every segment could have one or more speakers. Speakers spoke in a sequential manner that is one after the other in every 2 second segment. Here the features of the 2 second were averaged. Hence it results in a loss of accuracy.

Table 1 Accuracy of ANN Approach

Dataset 1	Dataset 2
96.12 %	96 %

B. RNN for 2 speakers

The data was segmented at every 0.1 second. Every 0.1 second captures a single phoeneme of the speaker. Here, features need not be averaged, rather the features are stored in the LSTM which increases the Accuracy.

So for every 0.1 second 9 rows of 13 mfccs are used to predict the result. Had ANN been used, we would have averaged the 9 rows of mfccs.

Dataset	Dataset 1		Dataset 2	
Dropout	0.2	0.5	0.2	0.5
Accuracy	98.113%	99.056 %	95.402 %	95.402 %

Table 2 Accuracy of RNN Approach

C. Adaptive LSTMs

Each speaker is assigned a network. This LSTM network is trained on both positive and negative segments of data. So every network has the task of only identifying its own speaker. Hence we see an increase in the accuracy.

The individual networks are trained in a way similar to the Single RNN. 0.1 second segments are created to train the each speaker's network. While testing the same file is fed into both the speaker's network. Whichever network gives a higher probability, the segment belongs to that speaker.

Dataset	Dataset 1		Dataset 2	
Dropout	0.2	0.5	0.2	0.5
Accuracy	99.056%	99.056 %	96.55 %	98.85 %

Table 3 Accuracy of Adaptive LSTM Approach

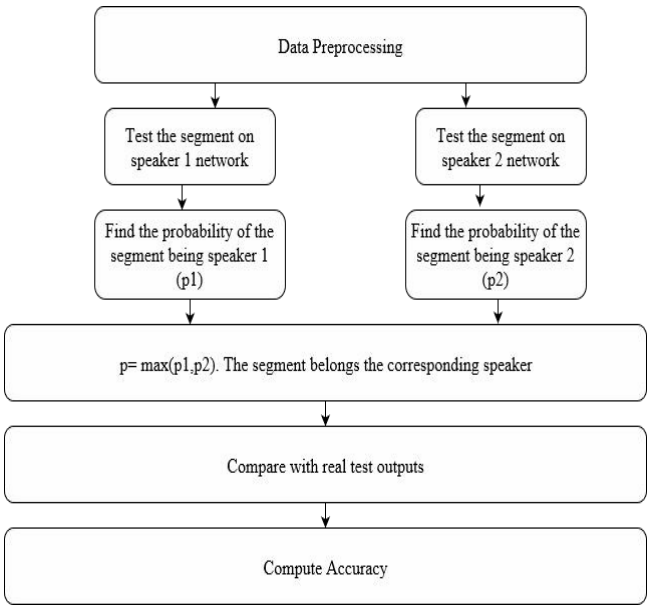


Fig. 9. Testing the Adaptive LSTM model

V. CONCLUSIONS

Speaker Diarization was implemented using various types of Neural Networks. It is evident from the above accuracy tables that ANN performs fairly but looses accuracy because the mfcc coefficients for a segment are averaged and given as input. ANN does not identify sequential patterns in the mfcc coefficients of an audio segment.

However, RNN approach for 2 speakers the mfcc coefficients are not averaged and the RNN model is able to identify the sequence of the mfcc coefficients of an audio segment. The third Approach Adaptive LSTM boosts the accuracy further because in this approach individual neural network are used for an individual speaker. Every Neural Network has to essentially focus on identifying it's respective speaker. Each neural network is trained on only on 2 categories. One is the positive sample (the respective speaker) and the other is the negative sample (Any other speaker). Hence, this paper provides an approach to Speaker Diarization.

REFERENCES

[1] https://towardsdatascience.com/speaker-diarization-with-kaldi-e30301b05cc8

[2] Xavier Anguera, Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland, Oriol Vinyals, Speaker Diarization: A Review of Recent Research, First draft submitted to the IEEE, 19th August, 2010.

[3] Speaker Diarization for Meeting Room Audio Hanwu Sun, Tin Lay Nwe, Bin Ma and Haizhou Li

[4] Arun Chandhandrasekhar, Shashankar Sudarsan "AUTOMATIC SPEAKER DIARIZATION USING MACHINE LEARNING TECHNIQUES"

[5] http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/

[6] Speaker Diarization using Deep Recurrent Convolutional Neural Networks for Speaker Embeddings Paweł Cyrta1 , Tomasz Trzcinski´ 1,2 , Wojciech Stokowiec 1,3 1 Tooploox, Poland, 2 Warsaw University of Technology, Poland, 3 Polish-Japanese Academy of Information Technology, Poland

[7] https://www.isca-speech.org/archive_open/archive_papers /iscslp2006/B11.pdf https://en.wikipedia.org/wiki/Mel-frequency_cepstrum