International Journal of Computer Applications (0975 - 8887)
Volume 175 - No.31, November 2020

An Overview of Speaker Recognition and Implementation of Speaker Diarization with Transcription

Arthav Mane
Department of Electronics Engineering
Sardar Patel Institute of Technology
Mumbai, India

Janhavi Bhopale
Department of Electronics Engineering
Sardar Patel Institute of Technology
Mumbai, India

Ria Motghare
Department of Electronics Engineering
Sardar Patel Institute of Technology
Mumbai, India

Priya Chimurkar
Department of Electronics Engineering
Sardar Patel Institute of Technology
Mumbai, India

ABSTRACT

This paper presents an overview of the generic process of a speaker recognition system and an implementation of its usage in a speaker diarization process. The motivation behind this paper is to present a simple implementation of a speaker diarization system that inculcates the usage of speaker recognition, speech segmentation and speech transcription. On the basis of various speech features such as Mel Frequency Cepstral Coefficients (MFCCs), Joint Factor Analysis (JFA), i-vectors, Probabilistic Linear Discriminant Analysis (PLDA), etc., speaker modelling is done to train Gaussian Mixture Models (GMMs), Hidden Markov Models (HMMs) and to use clustering. Speaker diarization is then implemented to get speakers speech segments which are then converted into text for the user. The methods discussed, and thus implemented, emphasize on maximum identification rate and minimal error in order to develop the functionality of speaker diarization and audio transcription and are aimed at helping the user to create a manuscript of the conversations that take place between multiple people.

Keywords

Speaker Recognition, Speaker Modelling, Audio Segmentation, Speaker Diarization, Speech Processing, Speech Recognition, Transcription

1. INTRODUCTION

Advancements in speech technology has attracted great minds of researchers and experts from all over the world. One part of this technology is the field of speech processing and speaker recognition. The extent of usage and scope of application for a robust speaker recognition system is unparalleled. Accounting to this, numerous studies have been published on how to implement a speaker recognition system, however, due its shortcomings in robustness, its usage is limited for applications such as biometric verification, speaker authentication, forensic purposes, etc.[1]. Nevertheless, the prominent reason behind the development of a speaker recognition system is the barriers in speech processing for different speakers, each having a unique way of speaking, accent, pitch, rhythm, pronunciation, etc [2]. Research in this field has been going on for a long time, and consequently, studies have been published proposing the implementation of a speaker recognition system. For the development of an automatic speech recognition system, implementation of speaker identification modules followed by speech recognition modules have been proposed in previous studies [3].

The focus of this paper is to present an entire speaker recognition system as an application that can actually be used for automatically generating manuscripts of public meetings in groups, offices, classrooms, conferences, etc. Extending the usage of a speaker recognition system, use of speaker diarization [4] to fundamentally answer the question “which speaker spoke when?” in a conversation had in a meeting involving numerous speakers [5], is made. This is basically done by segmenting the conversation audio input into speech segments for different speakers based on the features extracted. This entire process of diarization [6] is implemented by basically applying the speaker recognition process coupled with unsupervised clustering. Furthermore, each of the segments obtained via diarization are transcribed to have the entire conversation in text format into a file for future reference.

2. BACKGROUND AND METHODS

A typical speaker recognition system is shown in Fig. 1. As seen in the figure, the entire system can be categorized into two types of processes, namely offline process and online process [1][2]. The offline process consists of creating a training algorithm for generating a background model based on the utterances of non-target speakers. The online process, on the other hand, involves the real-time processing of the speaker recognition system, using the trained models, for identifying the speaker in a new utterance. The first step involves training a base or background model, like Universal Background Model (UBM), after performing feature extraction on the utterances of non-target speakers. Once training

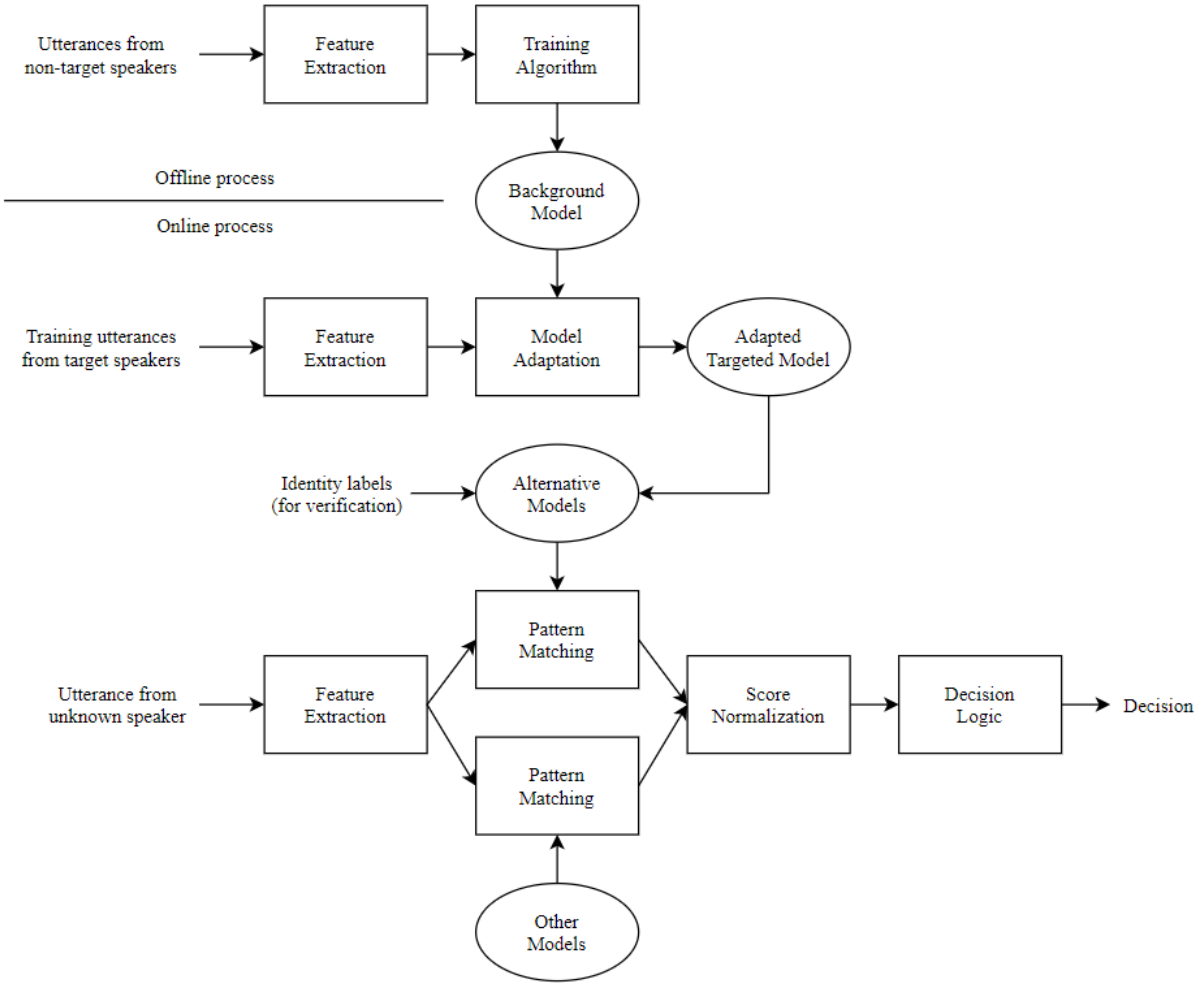

Fig. 1. A typical speaker recognition process

set is obtained of the utterances from the target speakers, it is fed into this model, after feature extraction, which results in the adapted targeted model. For testing this model on the utterance of an unknown speaker, pattern matching is used, which gives a likelihood score to the speaker. Later, this score is normalized and a decision is made on the identity of the speaker. This entire process along with speaker diarization can be broadly put into the following three steps.

2.1 Feature Extraction

Feature extraction is the first and the most fundamental step in a speaker recognition system. Numerous types of features can be extracted for a speech signal such as short-term spectral features, voice source features, spectro-temporal features, prosodic features and high-level features [7].

(1) Mel Frequency Cepstral Coefficients:
The Mel Frequency Cepstral Coefficients (MFCCs) feature extraction method is one of the very first approaches for speech feature extraction, and is still used extensively in speaker recognition based studies [8]. For each tone, having frequency f(in Hz), a subjective pitch is measured on a Mel scale [9].

$$fmel = 2595 \cdot log10(1 + \frac{f}{700}) \quad (1)$$

where, $f$mel in Mels is the subjective pitch that corresponds to a frequency in Hz. Thus, MFCCs form a baseline acoustic feature set for speakers as well as speech recognition applications [10][11]. MFCCs are basically a Discrete Cosine Transform (DCT) decorrelated parameter set computed via logarithmically compressed filter-output energies transformation. They are derived through a triangular filter bank that is perceptually spaced and processes the Discrete Fourier Transform (DFT) speech signal [12].

(2) Joint Factor Analysis:
Joint Factor Analysis (JFA), on the other hand, mainly incorporates the representation of a speaker utterance using a supervector (M) that comprises additive components from a speaker and channel/session subspace [13][14][15]. Specifically, the speaker-dependent supervector is defined as:

$$M = m + Vy + Ux + Dz \quad (2)$$

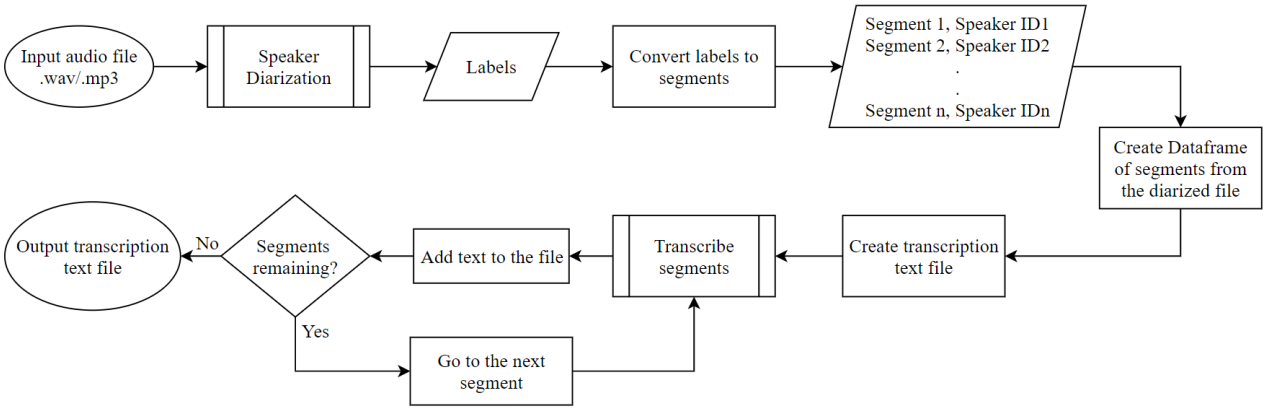

Fig. 2. Flowchart of speaker diarization and transcription process

where, *m* is the speaker- and session-independent supervector, *V* is the eigenvoice matrix and *D* is the diagonal residual for a speaker subspace, *U* is the eigenchannel matrix for session subspace, and *x, y, z* is the speaker- and session-dependent factors in respective subspaces. After computing the likelihood of test-utterance feature vectors, scoring is done against a session compensated speaker model possible using several JFA scoring methods [16]. Over the years, attempts have been made to implement speaker identification systems using MFCCs [17] and JFA.

(3) i-vectors:

In JFA, an assumption was made that the channel factors only handle the modeling of the channel effects, however Dehak [18] observed that the speaker features are also modeled by the channel dependent supervector. The usage of i-vectors was initially proposed by Dehak et al. [19]. Some improvements were made later on in its usage for speaker recognition [20]. For the total variability subspace training method makes the assumption that an utterance is represented by the GMM mean supervector given as:

$$M = m + Tw \tag{3}$$

where, *M* comprises speaker- and session- independent mean supervector *m* from a UBM and mean offset *Tw*. Furthermore, supervector *M* is taken to be normally distributed with the mean *m* and covariance TT^{t}, where *T* is the low-rank, total variability subspace. The low-rank vector *w* having a standard normal distribution N(0,1) is referred to as the i-vector [20]. Having extracted an i-vector from a speech sample, Linear Discriminant Analysis (LDA) is performed for inter-session compensation in an attempt to find a reduced set of axes that minimizes the within-speaker variance whilst maximizing the between-speaker variance observed in the i-vector space. For computing the classification score for a trial among i-vectors, cosine similarity scoring is used. Precisely, cosine kernel normalization [21] finds the cosine distance as a symmetric classification method and taking its advantage alleviates the need for common score-based normalization.

2.2 Speaker Modeling

Speaker recognition mainly encompasses the fundamental tasks of speech recognition, speaker identification and speaker verification [7]. A widely known model used extensively for speaker recognition is the Gaussian Mixture Model (GMM) [22][23] which is considered as an extension of one of the prior and simpler text-independent models, namely Vector Quantization (VQ) model [7][24]. For the Gaussian mixture speaker model, the Gaussian mixture density, defined as the weighted sum of M different component densities, is given by the equation

$$p(\vec{x}|\lambda) = \sum_{i=1}^{M} p_i b_i(\vec{x}) \tag{4}$$

where $\vec{x}$ is a D-dimensional random vector, bi$\vec{x}$ are component densities and pi are mixture weights. All component densities are a D-variate Gaussian function. In speaker identification using GMM, every speaker is represented its own GMM and is referred as λ. Here, a group of *S* speakers are represented by GMM's λ1, λ2,...,λS.

In text-independent speaker recognition, a UBM that represents all possible speakers is required along with the speaker model [25]. However, in text-dependent speaker recognition systems, GMM-UBM is the prominent approach.

For maximum efficiency and accuracy, a combination unsupervised HMM-UBM along with temporal GMM-UBM is used instead [26]. Hidden Markov Model (HMM) is a collection of stochastic automatons that on the basis of probabilistic rules follows transitions among states. HMM-UBM model is built having no knowledge of the speech transcriptions, and the parameters are then re-estimated with few iterations of Baum-Welch algorithm. Speaker HMM models are then derived from HMM-UBM with Maximum a Posteriori (MAP) adaptation [27]. For a given observation sequence, the speaker model with the maximum *a posteriori* probability is found.

2.3 Audio Segmentation

For diarization purposes, HMMs are used for recognizing sequential labels on the basis of respective sequence of audio feature vectors [6][28]. This is achieved by using the Viterbi

algorithm [29] for finding the sequence of states emitting a particular sequence of observations with the highest probability.
For the segmentation process involved in diarization, after feature extraction, the Fisher Linear semi-Discriminant Analysis (FLsD) [30] method is used for finding the near optimal feature subspace in terms of speaker discrimination. K-means clustering method is then performed for the number of speakers which in turn yields a sequence of cluster labels. If the number of speakers is not known, the entire clustering process is repeated for a range of numbers of speakers. The quality, efficiency and accuracy of this process is then decided using the Silhouette width criterion [31] and the optimal number of speakers is thus obtained.

3. IMPLEMENTATION DETAILS

The implementation of this project has been divided into two processes: speaker diarization and speech transcription. The flow of this entire process is depicted in Fig. 2. The detailed explanation for which has been given in this section.

3.1 Speaker Diarization

For the implementation of this process, the pyAudioAnalysis [28] library has been used.

—Input: The audio file format supported for diarization process is .wav or .mp3. However, .wav format is preferred for ease of use during transcription.

—Reading the audio file: The library includes the functionality of reading an audio input signal, provided the audio files are in the prescribed formats. On reading the audio input, the sampling rate and an array of signal values is obtained for the audio waveform signal. The length of the signal is thus the product of the duration of the input audio and its sampling rate.

—Diarization: Segmentation and clustering of the audio is also done using the functionalities provided by the pyAudioAnalysis library. For its usage, number of speakers in the audio is to be known, else, if not known, a range of numbers of speakers will be tried out the most probable number of speakers will be determined using the Silhouette width criterion [31]. An appropriate dimension value to be used for FLsD [30] also needs to be passed for finding near optimal feature subspace in terms of speaker discrimination. Feature extraction is then performed on the audio signal which is then wrapped with the classifiers after normalization. Clustering of speaker features is implemented using k-means clustering for the predetermined number of speakers. An HMM for the speakers is fitted using the pretrained Gaussian HMM model that comes included with the library. This HMM model is used to predict the segment labels for the audio file.

—Converting labels to segments: The labels obtained from the diarization function are converted to segments and the corresponding speaker IDs are also retrieved. The segments obtained are in the form of a multidimensional array wherein each row represents a segment with starting and ending time of the segment. The flags is an array of speaker IDs for the corresponding segments.

On performing speaker diarization, which speaker spoke when was found for 7 segments as shown in Fig. 3. The same information was tabulated, as shown in Fig. 4, to indicate the start and end time of a speaker for which that particular speaker was speaking, where ‘speaker0’ from Fig. 3 is mapped to speaker 2 and ‘speaker1’ to speaker 1.

3.2 Speech Transcription

For the implementation of this process, the SpeechRecogniton [32] library has been used. Using the Recognizer class provided by this library, transcription is done for each and every segment sequentially. The audio is handled for any noise and disturbances as well. For actually converting the speech in the audio segment into text, the APIs provided by Google is accessed via an internet connection in an attempt to minimize the error and save the time and effort required to train a separate speech recognition model for transcription. The text obtained is then sequentially added to a text file along with the corresponding speaker ID. For instance, the output transcription text file, as seen in Fig. 5, provides a speaker wise transcription of the conversation in the audio file. The audio file mix_3.wav consisted of a conversation between 2 speakers which is clearly depicted in the file. Following speaker diarization to find out which speaker spoke when, a sequential transcription of the sentences spoken by each speaker is written into the file. This helps to retain maximum information from the audio and make it available for future reference.

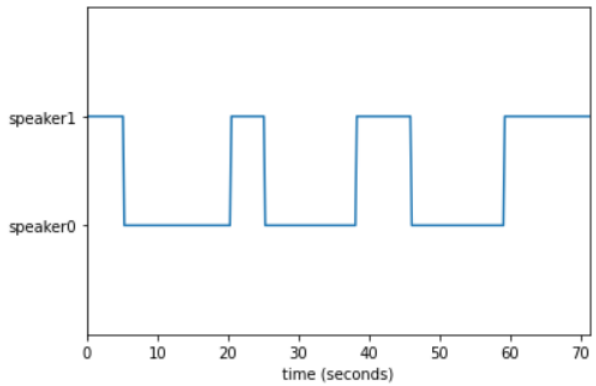

Fig. 3. Change in speaker with time in the audio file

	speaker	start	end
0	1	0.0	5.2
1	2	5.2	20.4
2	1	20.4	25.2
3	2	25.2	38.2
4	1	38.2	46.0
5	2	46.0	59.2
6	1	59.2	71.2

Fig. 4. Time (s) in the audio file for which each speaker was speaking

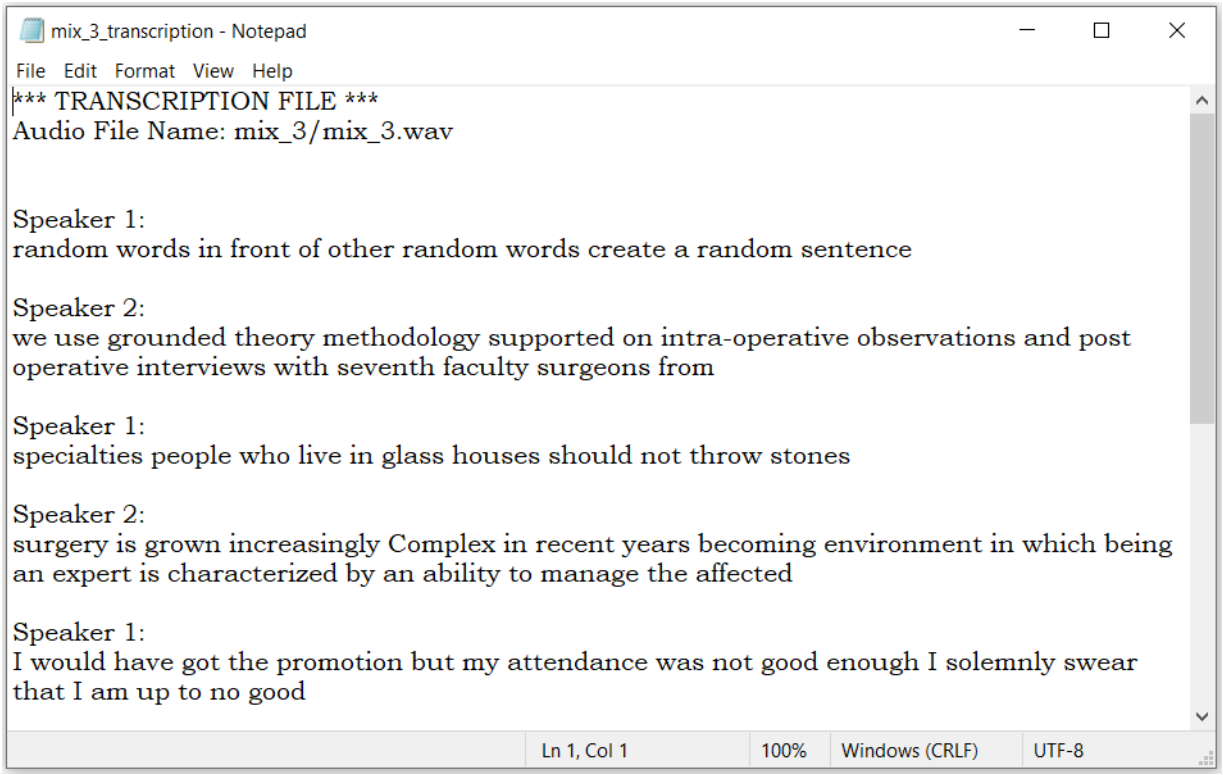

Fig. 5. Output transcription text file

4. RESULTS

For the purpose of experimentation, a total of 32 audio samples containing multiple speakers were tested. Of the 32 audio files, 27 were correctly diarized for an accuracy of 84.375%. The silent segments obtained due to unvoiced signals, that didn't include any speech utterances from a speaker, in the process of diarization are ignored during the transcription process. This has been done to prevent the further program from being compromised. However, the segments that were obtained, tend to show error in the number of speakers obtained through the unsupervised clustering process if the LDA dimension values are incorrectly passed. The transcribed text tends to show some error in the beginning and at the end, resulting due to the unclear transition from one speaker to another. This is caused due to the cross- fade of audio signals and overlap of speech. Additionally, since the SpeechRecognition library uses the internet for transcription, a disruption in the internet access causes the transcription process to fail. The transcription text is optimally spaced throughout keeping in mind the convenience of the user to read specific segments in the file and navigate through it whenever necessary. Additionally, all the files created in the process are saved in the same location as that of the input audio file to ensure ease of access and information retention to the user. Overall, the entire program provides a fairly accurate and efficient method to diarize and transcribe an audio file that contains multiple speakers.

5. CONCLUSION AND FUTURE SCOPE

The main objective of this paper was to give an overview of a typical speaker recognition system to develop a speaker diarization with transcription program that can be used in corporate group meetings for noting their minutes and ensure information retention in an efficient manner. It was also aimed at reducing the human effort for the same. The results were found to be quite satisfactory, with minimal error. Since, the proposed method can be used only if the entire audio file of the conversation or meeting is available, there is scope to develop an algorithm do so in real time. Emotion recognition can also be added in an attempt to make the transcription more insightful.

6. ACKNOWLEDGMENT

The authors would like to express their gratitude towards the Department of Electronics Engineering, Sardar Patel Institute of Technology, Mumbai. The authors are also immensely grateful to the entire teaching and non-teaching staff of the Electronics Department, S.P.I.T., for their support and guidance during the course of this research. The authors thank them for their assistance and comments that greatly improved this manuscript.

7. REFERENCES

[1] K. Selvan, A. Joseph and K. K. Anish Babu, "Speaker recognition system for security applications," *2013 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, Trivandrum, pp. 26-30, 2013.

[2] P. Verma, P. K. Das, "i-Vectors in speech processing applications: a survey," *Int J Speech Technol 18*, pp. 529546, 2015.

[3] S. Swamy, K. V. Ramakrishnan, "An Efficient Speech Recognition System," *Computer Science Engineering: An International Journal (CSEIJ)*, vol. 3, no. 4, 2013.

[4] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland and O. Vinyals, "Speaker Diarization: A Review of Recent Research," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356-370, February 2012.

[5] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1557-1565, September 2006.

[6] G. Sell and D. Garcia-Romero, "Speaker diarization with PLDA i-vector scoring and unsupervised calibration, *2014 IEEE Spoken Language Technology Workshop (SLT)*, South Lake Tahoe, NV, pp. 413-417, 2014.

[7] D. A. Reynolds, "An Overview of Automatic Speaker Recognition Technology, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Florida, pp. IV-4072-IV-4075, 2002.

[8] V. Tiwari, "MFCC and its applications in speaker recognition, 2010.

[9] S. Memon, M. Lech and L. He, "Using information theoretic vector quantization for inverted MFCC based speaker verification, *2009 2nd International Conference on Computer, Control and Communication*, Karachi, pp. 1-5, 2009.

[10] M. Sahidullah and G. Saha, "On the use of Distributed DCT in Speaker Identification, 978-1-4244-4589-3, 2009.

[11] S. Kim, T. Eriksson, Hong-Goo Kang and Dae Hee Youn, "A pitch synchronous feature extraction method for speaker recognition, *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Que., pp. I-405, 2004.

[12] M. A. Hossan, S. Memon and M. A. Gregory, "A novel approach for MFCC feature extraction, *2010 4th International Conference on Signal Processing and Communication Systems*, Gold Coast, QLD, pp. 1-5, 2010.

[13] P. Kenny, G. Boulianne, P. Ouellet and P. Dumouchel, "Joint Factor Analysis Versus Eigenchannels in Speaker Recognition, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435-1447, May 2007.

[14] P. Kenny, G. Boulianne, P. Ouellet and P. Dumouchel, "Speaker and Session Variability in GMM-Based Speaker Verification, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1448-1460, May 2007.

[15] P. Kenny, P. Ouellet, N. Dehak, V. Gupta and P. Dumouchel, "A Study of Interspeaker Variability in Speaker Verification, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 980-988, July 2008.

[16] O. Glembek, L. Burget, N. Dehak, N. Brummer and P. Kenny, "Comparison of scoring methods used in speaker recognition with Joint Factor Analysis, *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, pp. 4057-4060, 2009.

[17] F. Leu and G. Lin, "An MFCC-Based Speaker Identification System, *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, Taipei, pp. 1055-1062, 2017.

[18] N. Dehak, "Discriminative and generative approaches for long- and short-term speaker characteristics modeling: Application to speaker verification, PhD thesis, Ecole de Technologie Suprieure (Canada), 2009.

[19] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel and P. Ouellet, "Front-End Factor Analysis for Speaker Verification, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788-798, May 2011.

[20] M. McLaren and D. van Leeuwen, "Improved speaker recognition when using i-vectors from multiple speech sources, *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, pp. 5460-5463, 2011.

[21] N. Dehak, R. Dehak, J. Glass, D. Reynolds and P. Kenny, "Cosine Similarity Scoring without Score Normalization Techniques, *Proceedings Odyssey 2010 The speaker and language recognition workshop*, pp. 1519, 2010.

[22] D. A. Reynolds and R. C. Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models, *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 72-83, January 1995.

[23] D. A. Reynolds, T. F. Quatieri and R. B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models, *Digital Signal Processing*, vol. 10, issue 1-3, pp. 19-41, 2000.

[24] D. Burton, "Text-dependent speaker verification using vector quantization source coding, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 2, pp. 133-143, February 1987.

[25] W. Chen, Q. Hong and X. Li, "GMM-UBM for text-dependent speaker recognition, *2012 International Conference on Audio, Language and Image Processing*, Shanghai, pp. 432-435, 2012.

[26] A. Sarkar and Z. Tan, "Text Dependent Speaker Verification Using unsupervised HMM-UBM and Temporal GMM-UBM, *Interspeech*, San Francisco, 2016.

[27] J. Gauvain and Chin-Hui Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains, *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291-298, April 1994.

[28] T. Giannakopoulos, "pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis, *PLoS One 10*, 2015.

[29] M. Plumpe, A. Acero, H. Hon, X. Huang, "HMM-based Smoothing For Concatenative Speech Synthesis, *5th International Conference on Spoken Language Processing (ICSLP 98)*, Sydney, 1998.

[30] T. Giannakopoulos and S. Petridis, "Fisher Linear Semi-Discriminant Analysis for Speaker Diarization, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 7, pp. 1913-1922, September 2012.

[31] L. Vendramin, R. Campello and E. Hruschka, "On the Comparison of Relative Clustering Validity Criteria, *2009 SIAM International Conference on Data Mining*, 2009.

[32] Anthony Zhang (Uberi), "SpeechRecognition" Python library, https://github.com/Uberi/speech_recognition