The University of Hong Kong

Faculty of Engineering

Department of Computer Science

COMP7704
Dissertation Title
Real-time Speaker Recognizer

Submitted in partial fulfillment of the requirements for the admission to the degree of Master of Science in Computer Science

By
Pan Hao
3035349015

Supervisor's title and name: Dr. Beta C.L. Yip
Date of submission: 01/07/2019

Abstract

Speaker diarization has become more important in many speech processing tasks recently. Most state-of-the-art speaker diarization system decodes in an offline fashion and requires intensive computation and long processing time, which leads to the handicap for real-time applications. In this paper, we implemented the binary key speaker modelling approaches and built a fast offline speaker diarization system that can label the speakers in recorded audio, with visualization and audio player panel. An advanced version of the system is also developed, which can process the audio data in real-time with acceptable delay and correct earlier outputs when necessary.

Keywords: Speaker diarization, binary key speaker modelling, speaker clustering, MFCC

Declaration

The work contained in this dissertation has not been submitted to any publications, awards or education institutions. To the best of my knowledge and belief, this dissertation contains no material from other writers except where references are made.

Acknowledgement

Firstly, I would like to thank my supervisor, Dr. Beta C.L. Yip, for his great advice and assistance during the course of this dissertation.

Secondly, I would also like to thank Jose Patino, one of the researchers in the area of speaker diarization and binary key speaker modelling, who replied my email to answer my questions clearly and thoroughly.

I am also grateful to my boss Abby Ye for her support and understanding. She allows me to take the annual leave several times for this project despite the busy work in the company.

Finally, I would like to thank my family. Nothing would be possible without them.

Table of Contents

1. Introduction

1.1. Overview

With the boom of broadcast radio, TV channels and Internet, large volumes of audio or spoken documents are created and archived every day. Because of the difficulties and complexities of accessing and analyzing audio documents manually compared to text document, there is a growing need of using automatic audio processing technologies to efficiently index, search, access and analyze the information from audio data. The development of audio streaming also demand the real-time application of these technologies.

In many scenarios in presence of multiple speakers including conversations, meetings, conference and broadcast news, there are multiple audio sources or multiple speakers speaking within one audio channel. Speaker diarization is the process used in these cases to segment an input audio stream into speaker-homogeneous segments. Therefore, it is often summarized as "who spoke when" question [1]. The main difference between the speaker diarization and speaker recognition or speaker verification is that there is no speaker enrollment in the former so speaker identities are completely unknown. Another difference is that the temporal information is more important for speaker diarization than other speaker processing tasks.

Speaker diarization is a vital area in the community of speech processing because it provide the metadata in the audio of multiple speakers including the speaker segment labels, position of speaker turns and number of speakers, which can provide more context of the speech and be used for information retrieval. In the scenario of two speakers, for instance, doctors and patients in medical recording or customer and customer service in telephone conversation, speaker diarization can be used as source separation tool so that the further analysis on the speech of each side can be performed more easily. Another important application of speaker diarization is to find the boundary of sentences in conversation and the corresponding speakers of the sentences, to improve the readability and overall accuracy of the automatic speaker recognition (ASR) system. Generally, speaker diarization is an important front-end tool such that the audio information output can be more efficiently used as input in multiple speech processing tasks including spoken document indexing and retrieval, speaker recognition and speech-to-text transcription [2].

The traditional three primary application domains of speaker diarization is telephone conversation, broadcast news and conference meetings [3]. The audio streams from these domains are different in style of the speech, style of the noise source, numbers and locations of microphones, configuration of environment and therefore present unique challenges. [4] makes detailed comparison between broadcast news and conference meetings. The majority of the literature in speaker diarization will only focus on one of the three cases and some propose specific techniques to tackle some unique problems. For instance, [5] propose the acoustic beamforming technology to take advantage of the multiple microphones available in the meeting room domain to facilitate the speaker diarization process. Therefore, the speaker diarization system that has advantageous performance in one domain may not have comparable performance in other domains, and this domain-specific problem negatively affects the usability and extensibility of some systems.

Speaker diarization can also be referred as speaker segmentation and clustering, as the majority of diarization approach consist two main steps of segmentation and clustering [6]. However, there is a recent work that excludes the clustering step to perform fully supervised diarization [7].

1.2. Offline, online and real-time diarization

While the majority of the past works aim at improving the accuracy of the speaker diarization system on recorded audio, there is limited work aims at improving the speed of the diarization system and the possibility of real-time application.

Speaker diarization system can be differentiated as offline and online system. The offline system can access the whole audio recording before processing, and the clustering step is performed only when complete audio stream has been segmented. This means it is hard to adopt an offline diarization technique in real-time applications where the audio processing has to be conducted simultaneously or with acceptable latency when the audio is input.

Online diarization, on the other hand, only have access to the audio data up to the point that is been recorded, which means the diarization have to perform in a “left-to-right” fashion [8] that process and assign the segments once they are created and detected in the audio stream. Therefore, online speaker diarization is more suitable for real-time applications. However, offline speaker diarization is still the main focus in the field of speaker diarization [4] and there is limited work on online speaker diarization. A real-time speaker diarization system for the meeting environment is proposed in [9]. However, the system relies on the speaker seat locations and has the limitation of detecting only one speaker in one frame even if there are multiple speakers speaking. The online speaker diarization based on Gaussian mixture models (GMMs) and male, female and noise models, is tested with broadcast news data [10]. However, this system has difficulties dealing with speech overlapped by music. The novel Maximum a Posteriori (MAP) adapted transform within the i-vector speaker diarization framework proposed in [8], have a preferable diarization result for two-person telephone conversation audio, but still have diarization error rate (DER) 50% worse than offline system.

Generally, the performance of the online system is much worse than that of the offline system, but a state-of-art online system proposed recently that used Unbounded Interleaved-State Recurrent Neural Network (UIS-RNN) algorithm is demonstrated to have performance that is comparable with offline diarization [7]. The UIS-RNN model that used to predict speakers labels, is learned in a supervised manner, given the speaker label with temporal information and speaker-discriminative embedding (d-vectors) extracted from training data. Consequently, this process require large training data that incorporate temporal data and the diarization results will be significantly affected by the quality of the speaker embedding and not robust for different domains if the training data of UIS-RNN is domain-specific. Another obstacles for this method in real-time application includes the complexity and large computational resources required for the system.

1.3. Objectives

Given the limited research on the speed of the speaker diarization system and the gap of the real-time speaker diarization system, the general aim of this dissertation is to build a fast offline speaker diarization system that can label the speakers in recorded audio, and develop an advanced version of the system that can perform diarization in the real-time context.

More specifically, the proposed speaker recognizer system in this dissertation should fulfills the following requirements:

1) The system can perform speaker diarization for recorded audios like radio talk or phone conversation;
2) The system can perform speaker diarization in a real-time fashion that can process the live speech audio and generate output as the input is analyzed. The system can correct the earlier output when necessary. Reasonable buffers of data or short delays are acceptable;

3) The system should be language-independent and operating-system independent; It is preferable if the system is domain-robust
4) The system should not require the number of the speakers, identity of the speakers or voice samples of the speakers for the training.
5) The diarization results should be presented clearly in text form and being visualized. The audio playback should be available for result examination

1.4. Scope of the dissertation

As presented in Section 1.1, different application domains present unique diarization challenges. This dissertation will not aim at solving any domain specific challenges and aims at building a system that is domain-robust for different domains. Broadcast news audio will be firstly considered for training or testing purpose in this dissertation, because of the availability of the audio data and the higher difficulties to analyzed, due to the fact that there are usually more people speaking comparing with the telephone conversation audio and there are more noise and interruption comparing with the conference meeting audio.

The scope of this dissertation is restricted to the audio signal processing techniques in speaker diarization context, so no information other than input audio signal itself, can be used in the proposed system. The speaker diarization techniques that incorporate information that is not from the audio signal such as environment configuration including seat and microphone location [9], and visual activity sensing including face tracking [11] and focus of attention tracking [12], to assist speaker diarization, are all out of the scope of this dissertation.

Speaker diarization assumes no prior information of speaker identities and number of speakers for the input audio available. In this dissertation, these informations will also be unknown and no speaker enrollment process should be designed in the system. The speaker recognition or speaker detection tasks that have the access to the voiceprint of the speakers within the audio, is out of the scope of this dissertation.

This dissertation will only consider the design of the software system without special requirement on the hardware architecture. The specialization framework proposed in [13] to perform parallel implementation of GMM training on GPU to speed up the diarization system falls outside the scope of this dissertation.

The scope of this dissertation will only focus on speaker diarization system, without considering the combination of the speaker diarization system with other speech or audio processing system. The online diarization that incorporate ASR decoder in the system design proposed in [14] are outside of the scope.

Because speaker diarization system commonly consists of multiple components, there are many literatures that aims at improving the performance of one specific component in the system, for instance, speaker activity detection (SAD), speaker modelling or clustering. However, in this dissertation, the emphasis will be put on the overall system design for both offline and real-time context, and the investigation on the enhancement of the existing algorithm of specific components is subordinate

1.5. Organization of the report

The remaining sections of this dissertation will be organized as follows

Section 2 analyses the research problem and provide an overview of existing speaker diarization system. Section 3 provide the design idea and adopted algorithm of the proposed speaker diarization system. Section 4 outline the parameter, system configuration and the results of the experiment on our system. Section 5 concludes the dissertation with the discussion and summary of contribution and suggests the future direction of research

2. Analysis of problem

The aim of the section is to give an introduction of the existing speaker diarization problem, outlines the generic speaker diarization system, explain the role and functions of each common components in a complete speaker diarization system and discuss the problem involved in our system design and possible solutions.

2.1. General speaker diarization architecture

There are several previous works provide a comprehensive review on the existing diarization techniques and systems [2, 3, 15, 16]. The general and prototypical speaker diarization architecture and its components that summarized by these works are illustrated in Figure 1 – 3.

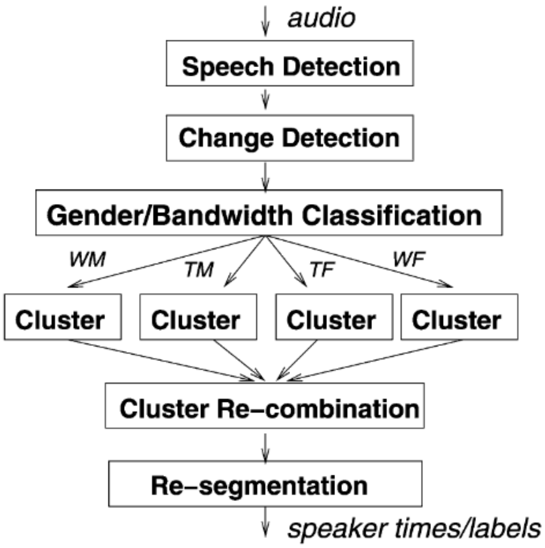


Figure 1 General speaker diarization architecture; Picture extracted from [3]

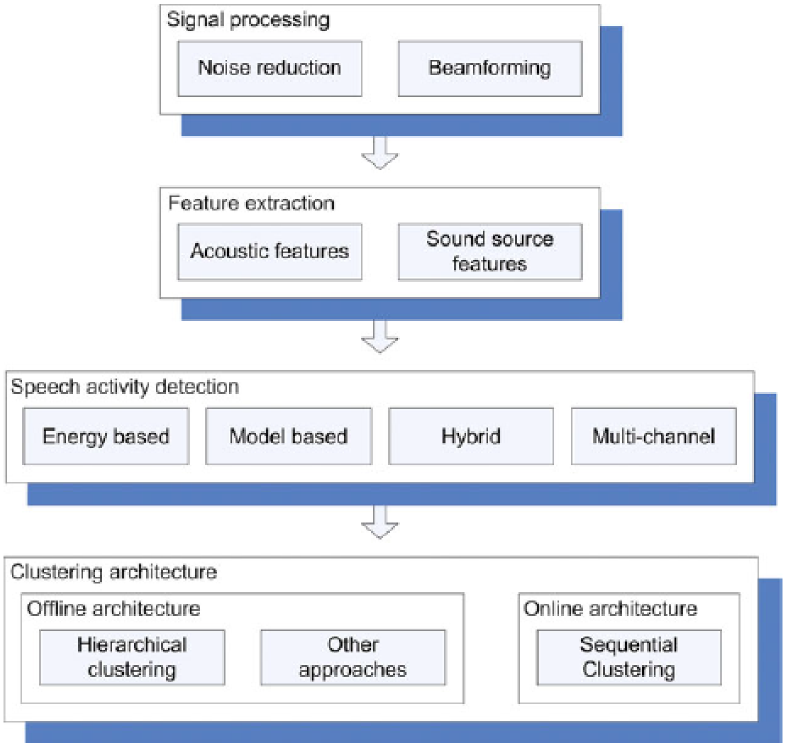


Figure 2 General speaker diarization architecture; Picture extracted from [15]

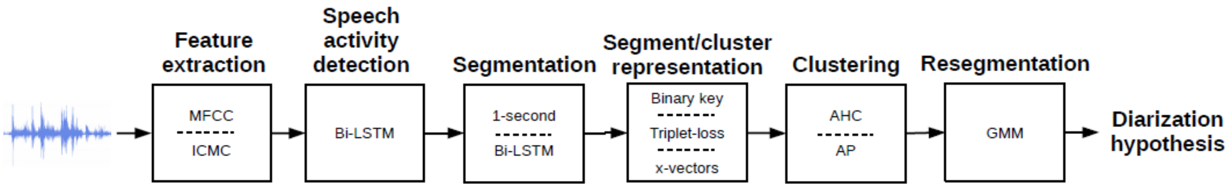


Figure 3 General speaker diarization architecture; Picture extracted from [16]

The common components of speaker diarization system in the literature consists of feature extraction, speech activity detection (SAD) or voice activity detection (VAD), segmentation or speaker change detection and clustering. Other components that may be incorporated in the system design includes domain-specific signal preprocessing such as noise reduction using Wiener filtering [17] and multi-channel acoustic beamforming firstly proposed in [18], gender/bandwidth classification [3]., speaker embedding extraction or segment/cluster representation and post-processing such as resegmentation and SAD post-filtering [19].

The function of each component that exists in previous work and our system will be further explained in the following.

2.2. VAD and SAD

Voice activity detection (VAD) or speech activity detection (SAD) is commonly used as the preprocessing step in various speech processing tasks including speech recognition, speech verification and therefore is included in the majority of the speaker diarization system. It is used to identify the region of the audio data that is being voiced or containing speech, from the unvoiced or non-speech regions that contain silence or background noise.

Energy-based voice detection and model-based voice detection are two main approaches of SAD. The energy-based voice detection removes the silence part based on the energy level and has the advantage of simplicity and speed. However, this approach fails to distinguish the load noise from the speech, and therefore is ineffective in many application domains of speaker diarization [20, 21]. To avoid the limitation of energy-based detection, model-based detection that is developed on the different acoustic phenomena, are more frequently used in speaker diarization system [15]. The dominant energy-based approach is the use of GMM. The simple SAD approaches divide the input signal into speech or non-speech regions using the GMM to model two different classes while the more complex approaches will use GMMs to model more voice classes including gender (male/female), bandwidth conditions, music and noise. One example is the SAD process to use five models for different type of non-speech voice and three models for speech [22].

As explained in Section 1.2, the attempt to improve current SAD/VAD accuracy is out of the scope of this dissertation. Therefore, we will directly implement the suitable approach in the system. Model-based approach will be used in the system design given the limitation of energy-based approach. The speed and simplicity of implementation is the top factors to consider to choose suitable algorithm.

2.3. Acoustic feature extraction

The raw audio data is usually converted into a sequence of acoustic feature vectors that contains speaker specific information before the segmentation and clustering. This feature extraction step try to acquire the acoustic features that contain formant information, model the mode of excitation and the shape of the vocal tract when people producing speech [15].

The cepstral-domain features that are generated from short-time segment (typically 10-30ms) of the speech is the most popular techniques in the community of speaker diarization. These features are capable of capturing the energy fluctuation in different frequency bands and resonance properties of vocal tract [15]. Common features include Mel Frequency Cepstral Coefficients (MFCC) [23], Perceptual Linear Predictive Coefficients (PLPC) [24], Linear Prediction Cepstral Coefficients (LPCC) [25] and constant Q transform Mel-frequency Cepstral Coefficients (ICMC) [26] . These features are different in frequency analysis and frequency smoothing techniques and the comparison and evaluation of these acoustic features in speaker identification task is presented in [27].

2.4. Segmentation

Speaker segmentation aims at splitting the original audio stream into segments containing one active speaking speaker and many other literatures adopt speaker change detection (SCD) techniques to fine the speaker change points in the audio [2, 3] . One popular segmentation algorithms is the use of Bayesian information criterion (BIC) firstly introduced in [28] and firstly used in speaker segmentation in [29]. Many state-of-art systems incorporate BIC as a segmentation metric in the following [18, 30]. As BIC approach is computationally intensive, several works (e.g. [31]) propose modification or other technologies used with BIC to speed up the process. Some common alternative segmentation approaches include Generalized Likelihood Ratio (GLR) [32] and Kullback–Leibler (KL) divergence [33]. Some recent papers propose advanced machine learning technology including deep neural network (DNN) to find speaker change points [34].

2.5. Speaker modelling

The speaker modelling is the heart of the many state-of-art speaker diarization system, and the similar step is called embedding extraction in some other research. This step aims to find speaker-discriminative identifiers that can be used to distinguish unique speaker from each other. Some models or embeddings are first found to be effective in speaker recognition or identification tasks, and then are adopted in speaker diarization tasks. Famous speaker models used in state-of-art system include Gaussian Mixture Model (GMM) and Hidden Markov Model (HMM) and famous speaker embeddings include speaker factors [35], i-vectors [8] and d-vectors [36, 37].

2.6. Clustering

Clustering is another important component in speaker diarization focusing on agglomeration of segments from segmentation step into groups that from the same speaker. Clustering approaches can be divided into two categories: offline clustering and online clustering, which determine whether the system can be performed in offline or online manner.

One of the most common clustering approach in the literature of speaker diarization is the bottom-up approach of agglomerative hierarchical clustering (AHC) [2]. This approach starts at certain number of clusters and successively merge the clusters and reduce the number clusters by one at each iteration until only one cluster is left. Assume the initial number of cluster is K, then the iterative process generate a set of clustering solutions $\mathrm{C} = (C_1, C_2, \dots, C_K)$ with decreasing number of clusters, where solution C_1 has K clusters and C_K has one cluster. Then some clustering selection technique is used to select the best clustering solution from C.

Another popular approaches proposed recently is unsupervised i-vector clustering. [38] propose a system that uses i-vectors and probabilistic linear discriminant analysis (PLDA) which has preferable performance for multi-language telephone conversation data.

3. Design and construction of software system

In this section, the discussion of the design and construction of the software system will be divided into three subsections: the design of the speaker diarization system, the adaptions of the real-time system and the design of the visualization panel. The algorithm that is applied and the reason to choose a certain algorithm will be presented.

3.1. Design of the speaker diarization system

3.1.1. Overview

The design of the proposed speaker diarization system will use the diarization system firstly proposed in [39] and later upgraded in [40] as the pipeline. The system overview is illustrate in Figure 4.

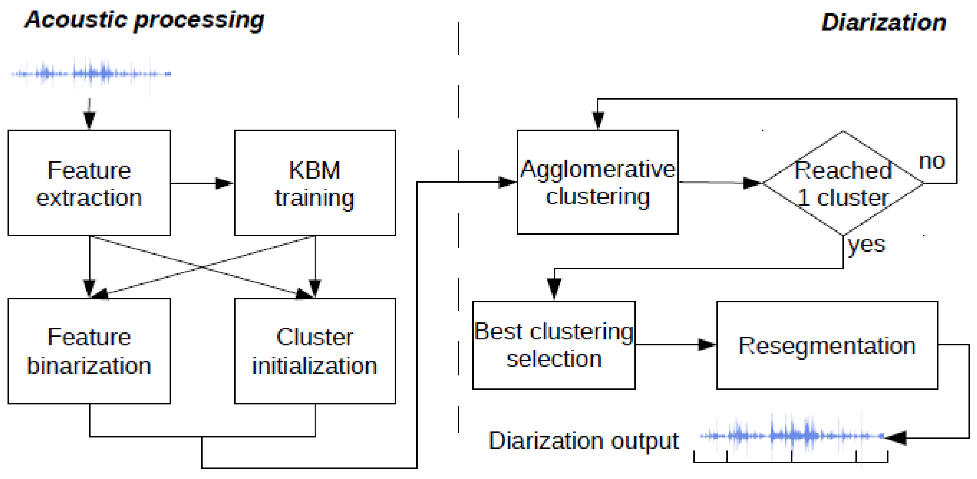


Figure 4 Diarization pipeline, picture extracted from [39]

For the proposed speaker diarization system, suitable modification on the pipeline system are made. The system design presented in this section is only for the proposed system that work in offline manners. For real-time version of this system, necessary adaption will be discussed in the section 3.2.

The workflow of the proposed speaker diarization system is presented in Figure 5 and it consists of below main blocks:

1) Voice activity detection block to detect and remove the regions of silence from the input audio
2) Acoustic feature extraction block to compute the Mel Frequency Cepstral Coefficients (MFCCs) from non-silent region of the input data to form to a features vector
3) Segmentation block to divide the non-silent audio regions into segments that contains one active speaker
4) Binary Key Background Model (KBM) Training block to train the MFCCs from step 2) to obtain the KBM that model the acoustic space of the audio
5) Feature Binarization block to transform the vector of features of segments or clusters into Cumulative Vector (CV) and Binary Key (BK), which is used as the representation of segments or clusters
6) Clustering block to merge the segments into clusters of homogenous speakers based on the CVs or BKs

7) Resegmentation block to refine the clustering result

Step 4) and 5) constitutes the Binary Key (BK) speaker modelling, which is the heart of this system.

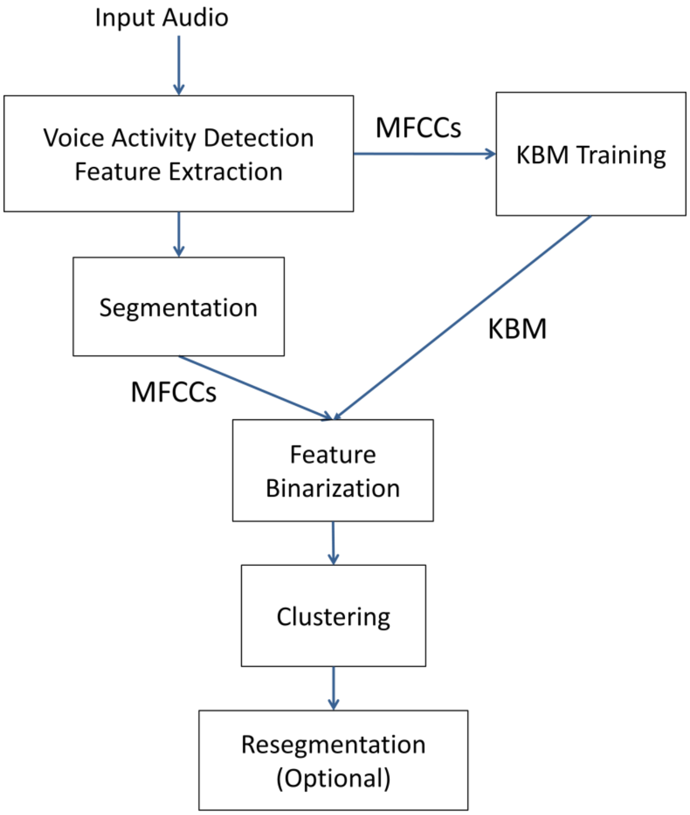


Figure 5 System workflow

The role of each step and the classic design in existing system will be presented in the following. The algorithm we have adopted in our system and the corresponding design idea will also be further explained respectively.

3.1.2. Voice Activity Detection

The model-based VAD algorithm that Google developed for the WebRTC project [41] will be applied in the design of our system, the reason includes:

1. The WebRTC project targets on real-time capabilities and its VAD algorithm has been widely used for different delay-sensitive scenarios [42], which is suitable for our offline and real-time applications.
2. The free and open-source implementation of the algorithm is available and there is Python interface (Py-webrtcvad [43]) that is well-suited for our development.

The VAD process in the system can be performed on the all audio data for the audio recording or the new input data for real-time audio stream. The result of the VAD is a mask vector that used to exclude the data segments or the acoustic features that contains no speech from the original data.

3.1.3. Feature extraction

In this dissertation project, the focus will be put on the MFCCs which are frequently used in the community of speaker diarization and more specifically, the online speaker diarization system [10, 8]. Librosa, the python library for music and audio analysis [44, 45], will be used to extract the MFCCs. The python code is simple and straightforward as below:

```
Features = librosa.feature.mfcc(y = input_data, sr = sampling_rate,
dct_type=2,n_mfcc = ncoeff, n_mels = nfilters, n_fft = length_of_fft_window,
hop_length = hop).T
```

The parameters used for MFCCs extraction will be discussed in section 4.3.

3.1.4. Segmentation

Speaker segmentation or speaker change detection (SCD) in many literatures, aims at finding the speaker change points in the audio so that splitting the original audio stream into segments that contains only one speaker [2, 3]. Although many system will used various distance metric to decide whether the speech of the adjacent windows are from the same or different speakers, a simple method that use fixed-sized windows to splits the audio into small equal-sized segments, will be used in our system.

3.1.5. BK Speaker Modelling

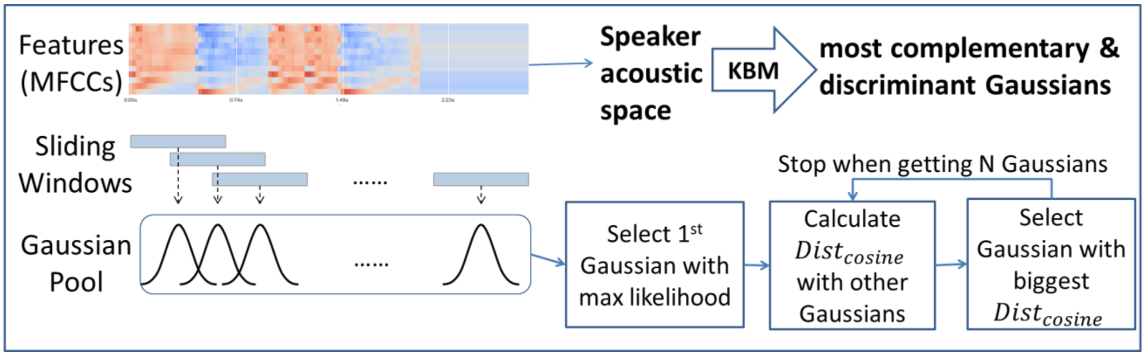
In this project, we will adopt the Binary Key (BK) speaker modelling techniques that is firstly proposed in [46] and later used in speaker diarization context in [39]. This technique has advantages of being domain-robust, requiring no external training data and running faster than real-time, so therefore can be utilized for our offline and real-time system.

The process of BK speaker modelling can be divided into two steps: KBM training and feature binarization, which are further explained as below.

KBM Training

The first step of the binary key speaker modelling is to train a GMM-like model called KBM from the acoustic features (MFCCs for our system) extracted from the input data. While the majority of the speaker modelling or speaker embedding extraction techniques requires large amount of external training data that is from the same domain of the tested data, the BK techniques train the KBM directly from the input tested data. Therefore, BK modelling is domain-robust and can avoid the negative impacts from the mismatch of the acoustic conditions between training and tested datasets.

Figure 6 shows the process of KBM training. To obtain the KBM, multiple Gaussians are trained on the data that is separated by a 2-second sliding window on the input features data. The mean and the standard deviation of the 2-second data will be calculated to get the Gaussian, which will be realized by multivariate_normal function from Scipy library [47] in Python. The shifting rate of the sliding window is determined automatically to have enough number of Gaussians. All resulted Gaussians form a Gaussian pool that covers all acoustic space in the input audio. Then *N* Gaussians are selected from the pool using single-linkage clustering strategy with cosine similarity as similarity measures. The first Gaussian is selected with maximum likelihood and the Gaussians with highest dissimilarity with previously selected Gaussians is selected subsequent until getting *N* Gaussians. Then the KBM is formed by these *N* Gaussians, which are considered the most complementary and discriminat Gaussians that can be used to represent the speaker acoustic space. The number *N* can be a fixed number or be determined relatively by a percentage of the total number of Gaussians in the pool.



Features (MFCCs)
Speaker acoustic space
KBM
most complementary & discriminant Gaussians
Sliding Windows
......
Stop when getting N Gaussians
Gaussian Pool
......
Select 1st Gaussian with max likelihood
Calculate $Dist_{cosine}$ with other Gaussians
Select Gaussian with biggest $Dist_{cosine}$

Figure 6 KBM Training

Although KBM is a GMM-liked model and GMM trained on the test data can also be used to produce speaker discriminative representation, [39] and [46] demonstrates that the KBM outperform classic GMM in distinguishing the speakers. Moreover KBM has the advantage of lower computational cost and consequently shorter processing time, comparing the expectation-maximization algorithm in GMM training.

Feature binarization

The second step of the binary key speaker modelling is the feature binarization that transforms feature vectors of an utterance to into a binary key, and this process is shown in Figure 7.

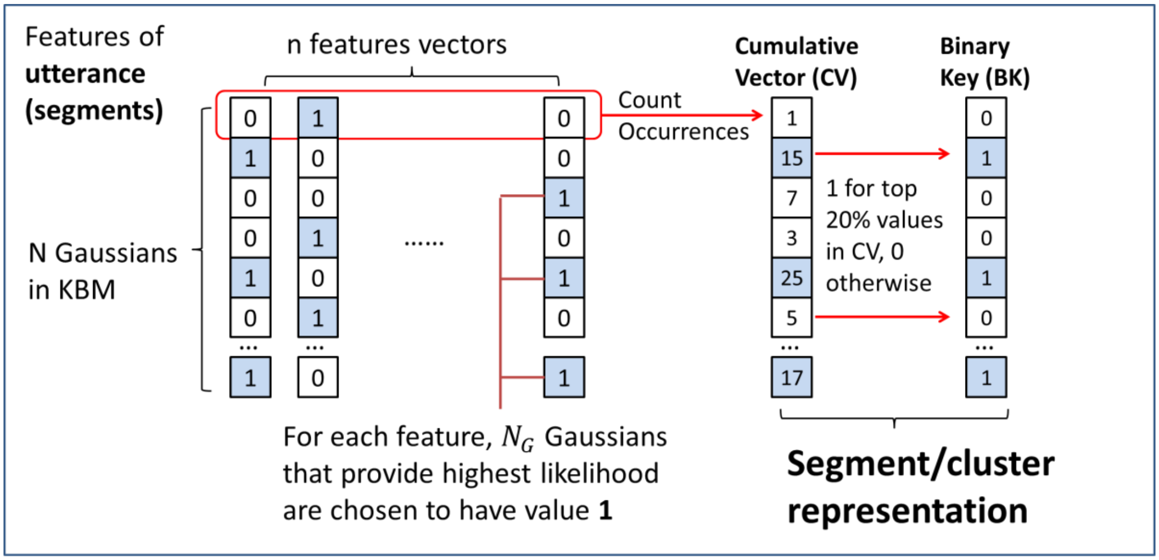


Figure 7 Feature Binarization

Firstly, a matrix with number of rows equal to N (size of KBM) and number of columns equal to number of feature vectors are created. For each column of each feature vector, N_G rows representing the Gaussians that provide highest likelihood for the given features are chosen to have value 1. All other rows in this column are set to have value 0. This process recorded the best N_G Gaussians from the KBM that represent each features. Secondly, we sum the values of each row, to form a vector called Cumulative Vector (CV) of size N. This CV contains the count of the occurrences of each Gaussians has been selected as the top-likelihood Gaussian for all feature vectors, so intuitively the larger number in the CV indicates higher impact of the corresponding Gaussian components. Thirdly, the binary key (BK) is obtained by setting the value as 1 for the positions that have top M% values in CV and the value as 0 for the other positions. For example if M equals to 20, then the positions whose values in CV are over 80th percentile will be set to have value 1 in BK.

The resulted CV or BK can be used to represent the input utterance and theoretically, the utterances from same speaker will have similar Gaussians in the KBM that have highest impact in modelling the speech, and therefore generate similar CVs or BKs. The input series of features in this step can come from the utterance of a short segment or a speaker cluster. Segment assignments and clustering can be performed by comparing the CVs or BKs from the segment or clusters by some similarity measures.

While the CV store the relative importance of the Gaussian components in KBM, BK only store the components that have greatest impact to fit the input features. Therefore, there is information that is missing in the process of transforming the CVs into BKs. In this project, CV will be considered over BK, as [48] found that CV is more speaker discriminative as the segment / cluster representation comparing with BK.

Because the value in the CV is the number of occurrences of the corresponding Gaussian has been selected for the input feature vectors, the longer feature vectors generally results in CV of larger magnitude. Consequently, to compare two CVs from feature vectors of different sizes, the angle instead of the magnitude of the vectors should be considered. The cosine similarity is therefore proposed as the similarity measures to compare CVs. The computation of cosine similarity between two vectors of positive integer values is simple and fast and the formula is:

$$S_{cos}(x, y) = \frac{x \cdot y}{\|x\|\|y\|}$$

$$, where\ S_{cos}(x, y) \in (0, 1)$$

The value of cosine similarity is between 0 and 1, where values close to 1 indicate high similarity between two CVs, while values close to 0 indicate high dissimilarity.

In programming levels, cosine similarity can be easily computed by cdist function from Scipy library [47] and the pseudo code is:

```
from scipy.spatial.distance import cdist
S = 1 - cdist(CV1, CV2, metrice = 'cosine')
# CV1, CV2 is ndarray
```

[40] has demonstrated the system using cosine similarity have more preferable performance comparing system using KL2 divergences between Gaussians. Moreover, the cosine similarity for the CVs has the advantage of simplicity and speed.

3.1.6. Clustering

The offline clustering adopted in this system is the AHC mentioned in section 2.6. The workflow of the AHC in my system is illustrated in Figure 8.

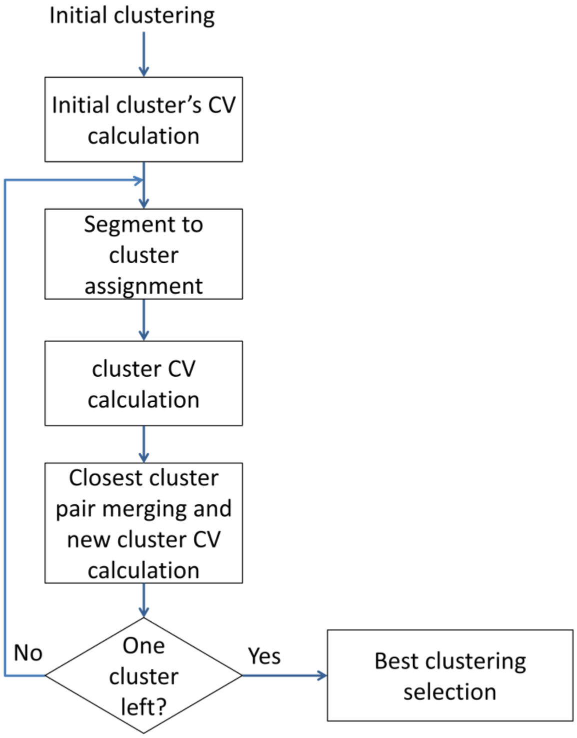

Initial clustering
Initial cluster's CV calculation
Segment to cluster assignment
cluster CV calculation
Closest cluster pair merging and new cluster CV calculation
No
One cluster left?
Yes
Best clustering selection

Figure 8 Agglomerative hierarchical clustering

There are many cluster initialization techniques have been studied in previous studies and one of the most common and simple approach is to divide the input audio data into a number of equal-sized chunks. This uniform initialization method generally results in equivalent performance found in [49] and has advantages of simplicity and speed. The initial number of clusters K should be larger than K_{opt} , the optimum number of clusters for the audio.

After the cluster initialization, the CVs for the initial clusters are calculated as the cluster embedding using the techniques in Feature Binarizaion. Then the segments of the input data are reassigned to the current clusters, by comparing the CVs of the segments to the CVs of the current cluster by cosine similarity. The segment is assigned to the cluster if highest cosine similarity between their CVs is achieved. After the reassignment, CVs are calculated for the new clusters. Then the cosine similarities will be calculated between the CVs of the new clusters and the cluster pair with highest similarity is merged. The total number of clusters is consequently reduced by one. The CV of the new clusters after merging will be calculated. If there is more than one cluster is left, the iterative process will continue from segment to cluster assignment. For each iteration, the clustering results will be stored to form a set of clustering solutions with size K.

Because the segment size is fixed and equal, the calculation of CVs for each segment only needs to be performed once. CV of segments can be used to obtain the CV of cluster. Therefore the calculation of CV in this clustering step requires low computational cost. The main computational cost of this step is put on the calculation of cosine similarity between segment and cluster, or between cluster and cluster. The overall processing time required for clustering is short and it is demonstrated in section 4.4.

The best clustering block in Figure 8 is responsible for selecting the best clustering solutions from all clustering solutions generated from previous iteration. Classic criteria to select best clustering includes Kullback-Leibler (KL)-based metrics [50], Generalized Likelihood Ratio (GLR) metrics [32] , T-test T_s metric [51]. The best clustering techniques that we adopted in this system is elbow criterion applied over the curve of within-class sum-of-squares (WCCS) per the number of clusters.

3.2. The adaptions for the real-time system

The proposed system in the section 3.1 is an offline system and several modification and adaptions are required to make the system to process the audio data in the real-time context.

For the VAD and feature extraction, the process can be performed in the online manner (i.e. the strict left-to-right manner). Both VAD and MFCC extraction can be performed on the whole recorded audio or on the newly input data. Therefore, no special modification needs to be made for the real-time system.

For the KBM training, the process are usually performed on the all data of the recorded audio to model the whole acoustic space of the input audio. In the real-time situation, KBM can also be performed on the newly input data combined with data stored before. However, two challenges exist. First, the minimum size of the initial Gaussian pool requires at least 10-12 seconds audio data for the KBM training. This means we need to have some data in the beginning of the real-time audio stream as the buffer to train the KBM and the diarization results are not available in the real-time for the these data. This is acceptable as the time buffer required is short. In the application that a short buffers are needed, the minimum size of the Gaussian pools can be lowered. The second challenge of KBM training in real-time context is that the although the KBM training is fast, this process will still lead to delay of the results when data stream is long due to the fact that training is processed on all data that is available. Since the KBM training is not necessary for every second of data in the real-time data stream, a possible solution to this challenge is to use a separate thread to train the KBM constantly so that consequent feature binarization and clustering can use the up-to-date KBM without being stuck by the KBM training.

For the features binarization, the calculation of the CVs can be performed in an online manner such that the CV can directly extracted from the features of newly input data. Therefore, no special modification needs to be made for the real-time system.

For the clustering, the AHC clustering method of the proposed system is an offline clustering method so modification is necessary. The proposed solution is to use a fast and naive online clustering method based on threshold first for the new data, then the offline clustering that run on another thread will be used to update the previous result when it complete. This method allows the results of the online diarization have the quality of offline clustering without causing too much delay.

In summary, the real-time version of the system combines the components that operates in online manner including VAD, feature extraction, feature binarization and online clustering, with components that operates in offline manner including KBM training, AHC clustering and resegmentation. Consequently, both the speed requirement for real-time processing and the quality of the offline processing can be achieved.

3.3. Design of the Visualization Panel

A visualization panel for the speaker diarization / recognizer system can help the users to understand and examine the diarization results intuitively. The target of the design of the visualization panel includes:

1. To show the number of the speakers
2. To distinguish different speakers and their speech on the timeline
3. To allow the users to playback the audio to compare with the diarization results
4. To allow the users to choose the start point at the timeline to play the audio and pause it in anytime
5. To use the same programming language as the speaker diarization system

To complete the above targets of the visualization panel, two python modules viewer.py and player.py are designed respectively.

The player.py is the module to open, play and pause the input audio files. Python packages Pyaudio is incorporated in the script to realize the necessary functions of the audio player.

The viewer.py will be used to show diarization results of the input audio or audio stream. The x-axis is the timeline of the audio while the y-axis shows the number of the speakers. Rectangle of different colors will be used to display different speakers and their speech in the timeline. The position and the length of the rectangles will be determined by the position and the length of the speech in the timeline respectively. The Matplotlib library, which is the most popular library in Python for 2D plotting will be used in this part.

The demonstration of the visualization panel is illustrated in Figure 9.

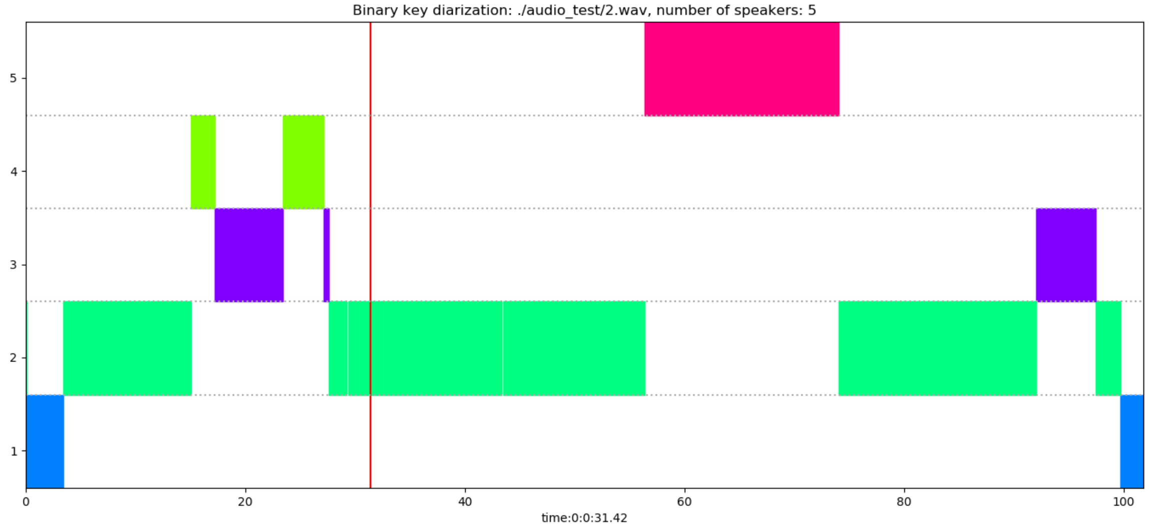


Figure 9 Visualization panel

3.4. Programming language and third party package

All code used to develop above diarization system is written in Python 3.7. The third party python package that used in the system includes:

- Numpy
- Scipy
- Librosa
- Py-webrtvad
- Pyaudio (only for audio playbacks)

All these packages are well tested in multiple operating systems including Windows and Linux. Although the overall diarization system proposed is tested only on Windows, it is operating-system free that can be easily implemented on the other platform.

4. Experimental results and discussion

4.1. Testing environment

The experiments of the proposed speaker diarization system are conducted on a personal laptop with Intel® Core™ i5-4200U CPU @ 1.60GHz 2.30GHz and Windows 8.1 operating system. This laptop is purchased in 2013 and the system configuration is somewhat outdated. It can be argued that the faster running time can be achieved by the computer with higher processing and computational power.

4.2. Data

The data used for testing purpose includes the broadcast news audios from SAIVT-BNews database [52] and meeting audios from ICSI Meetings Recorder corpus [53]. All data are free for research purpose, and used for testing or training in other speaker diarization research.

4.3. Parameters setting

The parameters used in the proposed systems and their values are summarized in table 1. All values are set based on the empirical studies.

Parameter	Value	Explanation or remark
Window length for feature extraction	25 ms	MFCC is the short-term spectral features that extracted from the short window (typically 20-30ms) of audio
Frame shift for feature extraction	10 ms	The length of the resulted MFCCs length = audio duration / 10ms the hop length (number of samples between successive frames when extracting MFCCs) = sampling rate * 0.01
Number of mel filters used	20	19-dimensional MFCCs using a 20-channel Mel-filterbank
Number of MFCCs employed	19	
Sliding Window length for computing Gaussians	0.02s	The shifting rate of the sliding windows on the MFCCs is determined based on the window length and audio duration to ensure there will be 1024 Gaussians trained
Minimum number of Gaussians in the initial	1024	

Gaussian pool		
KBM size	0.3 * Gaussian pool size	The resulted KBM size would be around 300 - 350, which is empirically the optimal KBM size that achieve lowest DER
Top Gaussians per features	5	Number of Gaussian component selected for top-likelihood foe each features in feature binarization step
Segment size; Segment increment after and before; Segment shifting	1s	When commutating the CVs for the input futures vectors, segments of 1s augmented 1s after and before (totaling 3s) will be used.
Number of initial clusters	16	The number of initial clusters should be larger than the maximum number of speakers in the tested data
Number of GMM components	6	These are parameters for resegmentation. Larger EM iterations and larger number of GMM components require more time to process
Number of expectation-maximization (EM) iterations	10	

Table 1. Summary of the parameters and their values

4.4. Runtime analysis

The proposed speaker diarization system are used to processed multiple recorded audio files with duration ranging from 20s to 5mins of every 20s increment to see the runtime of each component and the total system. The corresponding real-time factor (xRT, defined by runtime/audio duration) is calculated. For each file, the system is run for ten times to get the average processing time to avoid the outliers. The result is summarized in table 2.

Components	xRT, defined by runtime/audio duration)
VAD	0.0016 (around 0.16s for audio of 100s)
Feature extraction	0.0021 (around 0.2s for audio of 100s)
KBM training	0.017 – 0.025 (xRT is lower when duration is longer)
Feature binarization (Calculation of CV for all features data)	0.00024 (around 0.024s for audio of 100s)
Clustering (AHC)	0.0015 – 0.002 (xRT is lower when duration is longer)
Resegmentation	0.0058
Total system	0.027 – 0.04 (xRT is lower as the duration

	is longer)

Table 2. Real-time factor of different components.

The real-time factor of the total system on audio files is from 0.027 to 0.04. The average running time for the whole system is 3.4s for 100s audio is 3.4s (0.034xRT) and 8.47s for 300s audio.(0.028xRT). The overall trend of the real-time factor is decreasing as the duration of the tested data is increasing so therefore this system is preferable to process audio of long duration.

The components of VAD, features extraction and features binarization is running really fast and achieve low real-time factors. This means in real-time application, when these components performed in online manner, the output can be generate process the newly input data with short delay.

The most time-consuming part is the KBM training, but it still achieve real-time factor from 0.017 to 0.025. The average running time for the KBM training is 3.4s for 100s audio is 2.06s (0.0206xRT) and 5.08s for 300s audio (0.017xRT). That means in real-time version of the system, the KBM can be updated every 2s for audio around 100s and be updated every 5s for audio around 300s. It can be argued that this update frequency is good enough to have the KBM that is updated to model the acoustic space of the existing audio.

The AHC achieve the real-time factor from 0.0015 – 0.002. The average running time for the KBM training is 3.4s for 100s audio is 0.18s (0.0018xRT) and 0.45s for 300s audio (0.0015xRT). This speed of the AHC method demonstrates that the offline clustering result can be used to constantly update the online clustering results in the version of real-time analysis

4.5. Observation

There are several observation made during the testing of the proposed diarization system.

Firstly, the proposed system fails to distinguish between music and speech and usually classify the music in the broadcast news as one speaker. For example, the speaker 1 (marked in blue color) in the diarization of the example illustrated in Figure 10, are from the music in the start and end of the news.

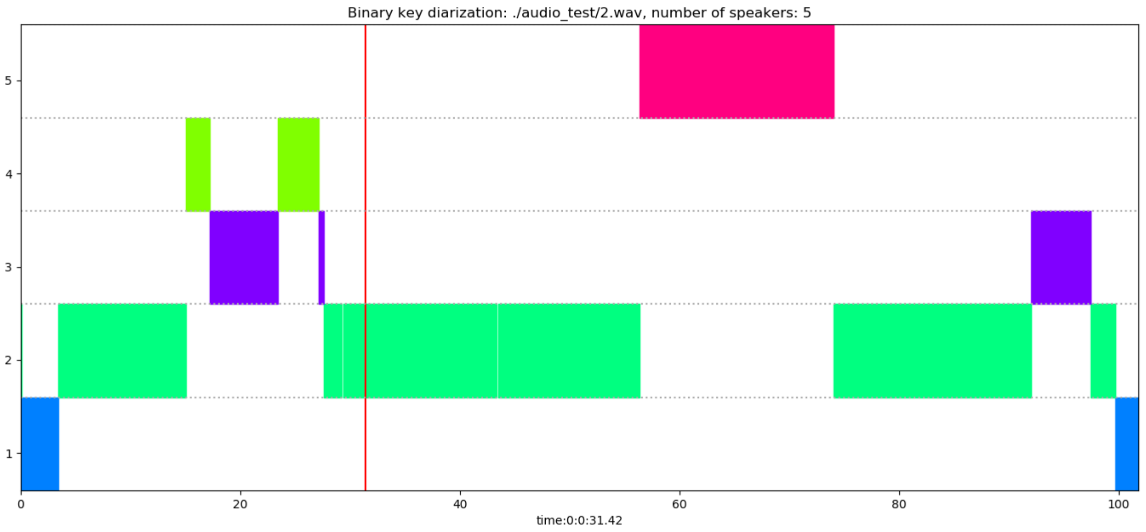


Figure 10 Diarization result of the broadcast news with ID 3054300 in AIVT-BNews database

This problem of failure to remove music is due to the fact that we only use a VAD that separates voice and silence, so music will also be included in the consequent steps of feature extraction, feature binarization and clustering. However this problem won't cause large confusion and have other negative effect on the usability of the proposed system, as the music can still be distinguished from the speech from speakers.

Secondly, the system has general better performance for fewer speakers. One example that the diarization on a two-person speaking news audio successfully divide all speaker change points and cluster correctly (except for the music part) is illustrated in Figure 11.

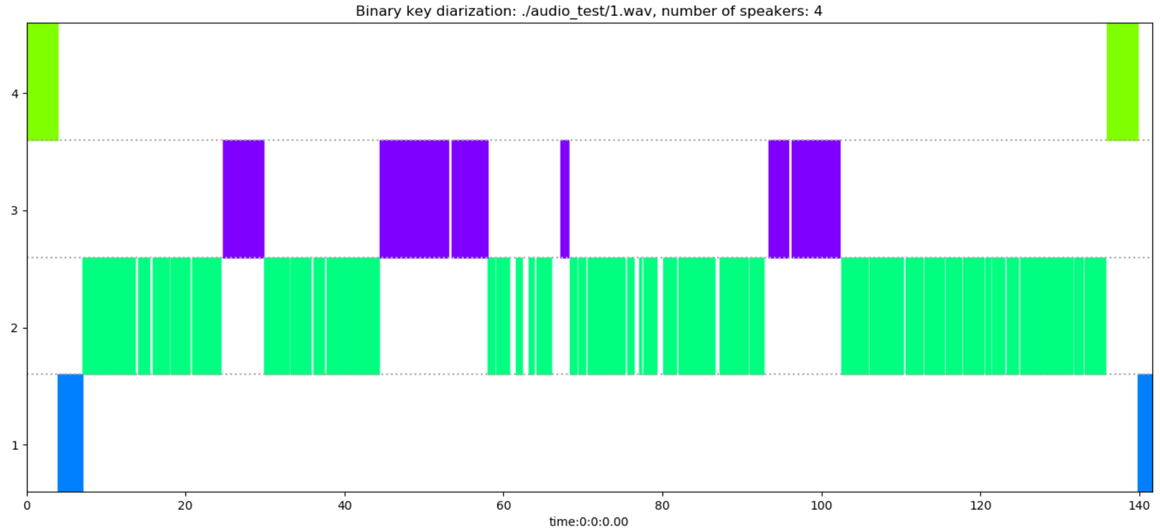


Figure 11 Diarization result of the broadcast news with ID 3063115 in AIVT-BNews database

However for broadcast news with more than 4 speakers, the diarization often fails to find correct number of speakers and often misses some speaker turns of the short interruption from other speakers.

Thirdly, the diarization has also been tested on the audio where the languages is Chinese and the no evidence is shown that the system have worse accuracy for Chinese language audio. Theoretically, the system is language-independent because the acoustic features (MFCCs) that we used depends no language-specific information.

5. Conclusions

This dissertation proposes a speaker diarization system that incorporates the binary key modelling approaches for offline and real-time speaker diarization tasks. The proposed system is domain-robust, operating-system free, language-independent and requires no external training data. A visualization panel is also designed for the users of the system to examine the diarization results.

The real-time version of the system is proposed on the basis of the offline system, which is the main contribution of this dissertation to fill the gap of lack of research on the real-time diarization system. Although it is stated in some literatures that only online diarization techniques can be used for real-time application, this dissertation proposes to combine components that operates in online manner including VAD, feature extraction, feature binarization and online clustering, with components that operates in offline manner including KBM training, AHC clustering and resegmentation. Consequently, the results generated by the real-time system can be corrected constantly by the results of offline system, thanks to high processing speed of the system. Runtime analysis of each component in the system is also made to demonstrate the speed of the overall system, and the low time delay for the real-time application.

There are several questions that remain answers and therefore the directions of the further investigations are suggested as follows:

- The frequency to run KBM training and offline clustering in real-time system need to be further determined

- The relationship between the frequency to run KBM training in real-time application and system accuracy need to be further analyzed

- The system need to be further improved in terms of speed and audio processing for real-time streaming data

Bibliography

[1] D. A. Reynolds and P. Torres-Carrasquillo, "Approaches and Applications of Audio Diarization," *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05)*, vol. V, pp. 953-956, 2005.

[2] X. A. Miro, 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, 2012.

[3] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Transactions on audio, speech, and language processing*, vol. 14, 2006.

[4] X. Anguera Miró, "Robust speaker diarization for meetings," *Universitat Politècnica de Catalunya.*, 2006.

[5] X. Anguera, C. Wooters and J. Hernando, "Acoustic Beamforming for Speaker Diarization of meeting," *IEEE Transactions on Audio, Speech, and Language*

Processing, pp. 2011-2022, 2007.

[6] M. Kunešová, Z. Zajíc and V. Radová, "Experiments with Segmentation in an Online Speaker Diarization System," *International Conference on Text, Speech, and Dialogue*, pp. 429-437, 2017.

[7] A. Zhang, Q. Wang, Z. Zhu, J. Paisley and C. Wang, "Fully supervised speaker diarization," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[8] W. Zhu and J. Pelecanos, "Online speaker diarization using adapted i-vector transforms," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5045-5049, 2016.

[9] S. Araki, M. Fujimoto, K. Ishizuka, H. Sawada and S. Makino, "A DOA based speaker diarization system for real meetings," *Hands-Free Speech Communication and Microphone Arrays*, pp. 29-32, 2008.

[10] J. Geiger, F. Wallhoff and G. Rigoll, "GMM-UBM based open-set online speaker diarization," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[11] K. Otsuka, S. Araki, K. Ishizuka, M. Fujimoto, M. Heinrich and J. Yamato, "A realtime multimodal system for analyzing group meetings by combining face pose tracking and speaker diarization," in *Proceedings of the 10th international conference on Multimodal interfaces*, 2008.

[12] G. Garau and H. Bourlard, "Using audio and visual cues for speaker diarisation initialisation," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010.

[13] E. Gonina, G. Friedland, H. Cook and K. Keutzer, "Fast Speaker Diarization Using a High-LevelScripting Language," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, 2011.

[14] D. Dimitriadis and P. Fousek, "Developing On-Line Speaker Diarization System," in *INTERSPEECH*, Stockholm, Sweden, 2017.

[15] T. H. Nguyen, E. S. Chng and H. Li, "Speaker Diarization: An Emerging Research," in *Speech and Audio Processing for Coding, Enhancement and Recognition*, New York, Springer, 2015, pp. 229-277.

[16] J. Patino, R. Y. Héctor Delgado, H. Bredin, C. Barras and N. Evan, "ODESSA at Albayzin Speaker Diarization Challenge 2018," in *IberSPEECH*, Barcelona, Spain,

2018.

[17] G. Friedland, A. Janin, D. Imseng, X. A. Miro, L. Gottlieb, M. Huijbregts, M. Knox and O. Vinyals, "The ICSI RT-09 speaker diarization system," in *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.

[18] X. Anguera, C. Wooters, B. Peskin and M. Aguiló, "Robust Speaker Segmentation for Meetings: The ICSI-SRI Spring 2005 Diarization System," *International Workshop on Machine Learning for Multimodal Interaction,* pp. 402-414, 2005.

[19] Barras, Claude, X. Zhu, S. Meignier and J.-L. Gauvain, "Multistage speaker diarization of broadcast news," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 14, no. 5, pp. 1502-1512, 2006.

[20] D. Istrate, C. Fredouille, S. Meignier, L. Besacier and J. Bonastre, "NIST RT'05S evaluation: pre-processing techniques and speaker diarization on multiple microphone meetings," in *International Workshop on Machine Learning for Multimodal Interaction*, Springer, Berlin, 2005, pp. 428-439.

[21] D. A. v. Leeuwen, "The TNO speaker diarization system for NIST RT05s meeting data," in *International Workshop on Machine Learning for Multimodal Interaction*, Springer, Berlin, 2005, pp. 440-449.

[22] D. Liu and F. Kubala, "Fast speaker change detection for broadcast news transcription and indexing," *Sixth European Conference on Speech Communication and Technology,* 1999.

[23] Davis, Steven and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE transactions on acoustics, speech, and signal processing,* vol. 28, no. 4, pp. 357-366, 1980.

[24] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *the Journal of the Acoustical Society of America,* vol. 87, no. 4, pp. 1738-1752, 1990.

[25] P. S. Jadhav, "Classification of Musical Instruments sounds by Using MFCC and Timbral Audio Descriptors," *International Journal on Recent and Innovation Trends in Computing and Communication,* vol. 3, no. 7, pp. 52-56, 2017.

[26] H. Delgado, M. Todisco, M. Sahidullah, A. K. Sarkar, N. Evans, T. Kinnunen and Z.-H. Tan, "Further Optimisations of Constant Q Cepstral Processing for Integrated Utterance and Text-dependent Speaker Verification," in *2016 IEEE Spoken Language Technology Workshop (SLT),* 2016.

[27] D. A. Reynolds, "Experimental evaluation of features for robust speaker

identification," *IEEE Transactions on Speech and Audio Processing,* pp. 639-643, 1994.

[28] G. Schwarz, "Estimating the dimension of a model," *The annals of statistics,* vol. 6(2), pp. 461-464, 1978.

[29] S. S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion," *Proc. DARPA broadcast news transcription and understanding workshop,* vol. 8, pp. 127-132, 1998.

[30] M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin and S. Meignier, "An Open-source State-of-the-art Toolbox for Broadcast News Diarization," *Interspeech,* 2013.

[31] Y. Huang, O. Vinyals, G. Friedland, C. Müller, N. Mirghafori and C. Wooters, "A fast-match approach for robust, faster than real-time speaker diarization," *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU),* pp. 693-698, 2008.

[32] H. Gish, M. H. Siu and R. Rohlicek, "Segregation of speakers for speech recognition and speaker identification," *1991 International Conference on Acoustics, Speech, and Signal Processing,* pp. 873-876, 1991.

[33] M. A. Siegler, U. Jain, B. Raj and R. M. Stern, "Automatic segmentation, classification and clustering of broadcast news audio," *Proc. DARPA speech recognition workshop*, 1997.

[34] R. Wang, M. Gu, L. Li, M. Xu and T. F. Zheng, "Speaker segmentation using deep speaker vectors for fast speaker change scenarios," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 5420-5424, 2017.

[35] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface and C. Vair, "Stream-based speaker segmentation using speaker factors and eigenvoices," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008.

[36] L. Wan, Q. Wang, A. Papir and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[37] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey and A. McCree, "Speaker diarization using deep neural network embeddings," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 4930-4934, 2017.

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

[39] X. Anguera and J.-F. Bonastre, "Fast speaker diarization based on binary keys," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[40] J. Patino, H. Delgado, N. Evans and X. Anguera, "EURECOM submission to the Albayzin 2016 speaker diarization evaluation," *Proc. IberSPEECH*, 2016.

[41] "WebRTC," [Online]. Available: https://webrtc.org/. [Accessed 2019].

[42] J. H. Ko, J. Fromm, M. Philipose, I. Tashev and S. Zarar, "Limiting Numerical Precision of Neural Networks to Achieve Real-Time Voice Activity Detection," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, 2018.

[43] "https://github.com/wiseman/py-webrtcvad," 2019. [Online]. Available: https://github.com/wiseman/py-webrtcvad. [Accessed June 2019].

[44] B. McFee, C. Raffe, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg and O. Nieto,

"librosa: Audio and Music Signal Analysis in Python," in *Proceedings of the 14th python in science conference*, 2015.

[45] "Librosa," 2018. [Online]. Available: https://librosa.github.io/librosa/. [Accessed Jun 2019].

[46] X. Anguera and J.-F. Bonastre, "A Novel Speaker Binary Key Derived from Anchor Models," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[47] "SciPy," 2019. [Online]. Available: https://www.scipy.org/. [Accessed Jun 2019].

[48] G. Hernandez-Sierra, J. R.Calvo, J.-F. Bonastre and P.-M. Bousquet, "Session compensation using binary speech representation for speaker recognition," *Pattern Recognition Letters*, pp. 17-23, 2014.

[49] X. Anguera, C. Wooters and J. M. Pardo, "Robust Speaker Diarization for Meetings: ICSI RT06s evaluation system," in *Ninth International Conference on Spoken Language Processing*, 2006.

[50] J. Rougui, M. Rziza, D. Aboutajdine, M. Gelgon and J. Martinez, "Fast incremental clustering of gaussian mixture speaker models for scaling up retrieval in on-line broadcast," in *2006 IEEE International Conference on Acoustics Speech and Signal*

Processing Proceedings, 2006.

[51] T. H. Nguyen, E. S. Chng and H. Li, "T-test distance and clustering criterion for speaker diarization," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[52] H. Ghaemmaghami, D. Dean and S. Sridharan, "Speaker attribution of Australian broadcast news data," in *Proceedings of the First Workshop on Speech, Language and Audio in Multimedia (SLAM): CEUR Workshop Proceedings*, 2013.

[53] ICSI Meetings Recorder corpus, 2006. [Online]. Available: http://www1.icsi.berkeley.edu/Speech/mr/.