IALP 2019, Shanghai, Nov 15-17, 2019

Development of a Filipino Speaker Diarization in Meeting Room Conversations

[1]Angelica H. De La Cruz, [2]Rodolfo C. Raga Jr.
{ahdelacruz@national-u.edu.ph, rodolfo.raga@jru.edu}
[1]National University, Philippines
[2]Jose Rizal University, Philippines

***Abstract*—Speaker diarization pertains to the process of determining speaker identity at a given time in an audio stream. It was first used for speech recognition and over time became useful in other applications such as video captioning and speech transcription. Recently, deep learning techniques have been applied to speaker diarization with considerable success, however, deep learning are conventionally data intensive and collecting large training samples can be difficult and expensive to collect especially for resource scarce languages. This study focused on investigating a speaker diarization approach for meeting room conversations in the Filipino language. To compensate for lack of resources, a one shot learning strategy was explored using Siamese neural network. Among the experiments conducted, the lowest diarization error rate yielded to 46%. There are, however, more parameters that can be tuned to improve the diarization results. To the best of our knowledge, no work in speaker diarization dedicated for Filipino language has yet been done.**

Keywords—speaker diarization, segmentation, clustering, MFCC, one shot, siamese network, meetings

I. Introduction

Communication is an important part of the daily lives of people. It is a way to impart knowledge and information to other people. For instance, a teacher is sharing his knowledge and presenting facts and ideas to his students through spoken or written discussions. Communication is also a way for people to express thoughts and feelings and at the same time, understand others. With the advent of technology, people have better means of communication and easier access to knowledge and information. For this reason, a huge amount of data are being generated everyday from people communicating around the world, and so, organizing these data and extracting information from this vast amount of data has always been a challenge.

In organizations such as universities and industry companies, communication is an essential part to their growth and development. Discussions for making decisions are always noted or recorded for future reference. However, writing notes during a meeting sometimes lead to missed or incomplete information being recorded due to the fast-paced flow of discussion, which is why audio recording of the meetings are done to avoid these situations. The audio recordings are then transcribed but this process is very time-consuming. Also, people that will use these for future reference will only be interested to know the important details. With the advancement of technology, people are now relying to automatic means of extracting important information.

Speaker diarization is a process of determining "who spoke when?" in an audio stream by partitioning the audio, that contains an unknown number of speakers and unknown amount of speech data, into homogeneous segments according to the speaker identity [1,2,3]. It was initially proposed as a topic related to automatic speech recognition which serves as an upstream processing step, but over the years, it become very useful for many tasks such as information retrieval, providing a more structured notes for medical records, automatic notes generation for meetings, court houses and parliaments, call center data analysis, and automatic transcription of broadcast news in television or radio [2,3].

The first Machine Learning based works of speaker diarization began around 2006 but significant improvements started only around 2012 and at the time, it was considered an extremely difficult task. Most methods used back then were GMMs or HMMs based. The first freely-distributed algorithm for speaker diarization with reasonable accuracy is LIUM[1]. The algorithm in the core of LIUM uses GMM with i-vectors [4]. A number of researchers have explored using neural network embedding for diarization tasks which largely focused on feedforward deep neural networks (DNNs) [6]. Currently, IBM Watson [2] offers a service for speaker diarization as part of the speech-to-text module in different languages. However, the Filipino language is not included in the languages catered by IBM Watson. Also, to the best of the researcher's knowledge, local researchers both in the academe and the industry have not yet developed and released a Filipino speaker diarization system. A key factor why local researchers were not able to fully explore this field is because of lack of data. But today, there has been a trend on using just a few samples in order to train a deep neural network known as one-shot learning, few-shot learning and n-shot learning that are mostly used in computer vision. While learning models notoriously require huge amounts of training data, a recent study [7] was able to show that it is possible to learn much information about a category from just one, or a handful, of examples.

Building on this trend, this study focused on developing a speaker diarization system for meeting room conversations in Filipino language through one-shot learning [9] implemented using a distinct form of neural network known as siamese neural networks [8]. This study will serve as a baseline in developing speaker diarization systems in Filipino and will also contribute to the exploration of using few samples as input in deep learning in the local setting. This paper is organized as follows: Section II discusses related works; methodology in Section III; results and discussion in Section IV; conclusion in Section V.

II. Related Works

A. Speaker Diarization

Speaker diarization is normally a process of speech segmentation followed by clustering. Speech segmentation is the task of detecting voice change point or speaker turns for

[1] http://www-lium.univ-lemans.fr/diarization

[2] https://www.ibm.com/watson

978-1-7281-5014-7/19/$31.00 ©2019 IEEE

partitioning an audio sample into regions each of which corresponds to only one speaker [10] while speaker clustering is a process that groups the speech segments together based on the similarity and association of the speech features in each of the segment.

1) Voice Activity Detection (VAD)

One of the most important preprocessing steps in speaker diarization is Voice Activity Detection (VAD), also known as speech activity detection. It is the process of detecting speech in an audio recording. The output of other subtasks in speaker diarization highly depends on the performance of the VAD [10]. VAD classifies audio into silence, speech or non-speech segments. The general approach for VAD is using Maximum Likelihood classification with Gaussian Mixture Models (GMMs), however, this approach requires prior knowledge in order to train the model in detecting speech and non-speech. The simplest systems only include two classes which are speech and non-speech, others also include silence and noise and more complex systems add more classes such as background noise, room noise, music, cross-talk, speech noise, and speech music. One of the approaches used for VAD is the model-based approach where labelled training data is required in order to train the VAD model. Existing studies for VAD include [11,12, 13,14,15,16].

2) Speaker Segmentation

Speaker segmentation is the task of detecting voice change point or speaker turns for partitioning an audio sample into regions each of which corresponds to only one speaker [17]. Speech features used in this task depends on the domain of analysis of the speech signal. In the frequency domain, features such as Distance between Power Spectral Density, Mel Cepstral Coefficients, Fundamental frequency, pitch, and formants can be used. Methods for speaker segmentation are divided into two, first is the blind segmentation where no prior knowledge is required and the other one is the model-based segmentation. Blind segmentation is a method that is speaker, text and language independent. However, the performance of the segmentation is low. In order to improve the segmentation process, model-based segmentation was introduced. Model-based segmentations make use of language models, voice databases and speech corpus. The output of this task is a sequence of segments where the boundaries indicate speaker turns or voice change. Existing works include [18, 19].

3) Speaker Clustering

After determining segments where speaker change point occurs, these segments are clustered or grouped together based on their similarity and association of the speech features in each of the segment. This process is called speaker clustering. The most commonly used algorithm for clustering speech segments is hierarchical clustering algorithm. Both top-down and bottom-up approaches were used for general speaker diarization systems, some have also explored using the hybrid approach [12]. The common features used are spectral energy and MFCC. It finds the closest pair of clusters using BIC distance measure and merges these clusters then iterate the process until such time it reaches the stopping criterion. The stopping criterion is determined based on the local BIC, global BIC or a certain threshold. Agglomerative clustering is sometimes done separately for each gender and bandwidth condition.

B. One Shot Learning

Over the years, machine learning and deep learning algorithms were able to successfully achieve state-of-the-art performance in various applications given large training samples [8]. However, in most cases, it is difficult and expensive to collect large datasets.

Deep learning is considered to be somehow similar to a human brain, but humans do not need thousands of samples in order to recognize an object. For instance, if humans are to recognize a face of a person, the human will only need to see that person's face for a few times, or even only once, and the human brain will already be able to recognize the person's face. Unlike in deep learning, in order for the machine to recognize a person's face, it would need hundreds or thousands of examples. This is the idea behind one shot learning, for a machine to be able to learn an object category with just one or few samples.

One shot learning is an object categorization problem introduced in 2006 [7] that is mostly found in computer vision. Since learning models requires a huge amount of training samples, [7] showed that it is possible to learn such models with only a few samples by using Bayesian Program Learning (BPL). However, BPL has its flaws such as its learning cannot be transferred unlike deep learning. Months after, Google Deepmind [9] demonstrated the ability of a memory-augmented neural network to rapidly assimilate new data, and leverage this data to make accurate predictions after only a few samples. This shows that deep neural networks can be used in one shot learning.

C. Siamese Neural Network

The term siamese means twins. Siamese networks are neural networks containing two or more identical sub network components [20]. This concept is first introduced by Bromley and LeCun in 1994 to solve signature verification as an image matching problem [20]. A siamese network is an artificial neural network that use the same weights while working in tandem on two different input vectors to compute comparable output vectors. The neural network is not learning to classify but is actually learning a similarity function between the inputs.

1) One Shot Learning with Siamese Neural Networks

A study [8] explored using siamese neural networks for one shot learning in image classification which employ a unique structure to naturally rank similarity between inputs. To implement one shot learning with siamese neural network, the first thing one should do is to create a base network. For most cases in the literature, convolutional neural network is the deep learning network used as a base network in the siamese neural network. The base network for a siamese network do not require a very large amount of data for training since it will not be used for learning how to classify inputs, it will only be used for determining how similar the two inputs are.

a) Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning algorithm which takes an image as an input and then outputs a class as the label of the image. The first layer of CNN is the convolutional layer where high level features of the input are being extracted followed by the pooling layer that is used to reduce the spatial size of the convolved feature and capture the dominant features. A fully connected layer is then added to the convolutional layer and pooling layer to learn non-linear combinations of the high-level features then classifies using the softmax classification technique.

In one shot learning, only a few samples are used as input to train the neural network. As the CNN will be used as the base network, CNN is trained using only few samples. The computed weights and bias value from the CNN architecture were saved and loaded to the siamese network.

The saved weights of the CNN were transferred to the siamese network that has the same parameters and weights. Assuming that the neural network model is trained properly, two pairs of input are used, the network then computes for the contrastive loss function then compare the two identical sub networks based on the loss value.

III. METHODOLOGY

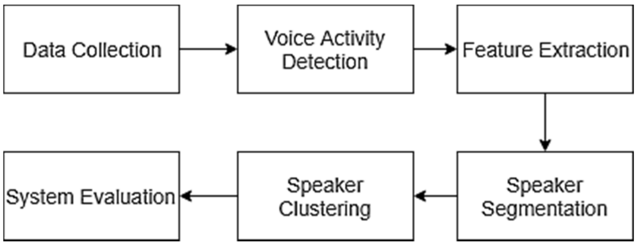

Fig. 1 Overview of the Methodology

A. Data Collection

Meeting room recordings were collected using a conference system equipment. A console (Kevler LM-500), a chairman unit microphone (Kevler LM-501) and 3 delegate unit microphones (Kevler LM-502) to tune and record the meeting. The duration of the meeting was 14 minutes and 27 seconds. There are 4 speakers in the meeting, consists of 2 male and 2 female individuals. Since the goal of the study is to explore the use of few samples for training, only 7% (52 seconds) of the data were used for training the base network and the entire recording was used for testing the model performance.

B. Voice Activity Detection

The speech data was cleaned using Audacity to remove the background noise such as the sound of the air-conditioning unit in the venue. After cleaning, the speech data underwent voice activity detection. In implementing VAD, energy features were extracted. The threshold used in detecting speech activity was set to the minimum energy seen in the speech signal which is 12.5% of the actual maximum energy of the signal. The sample window size is set to 250 ms and a sample overlap of 100 ms. Timestamps of detected speech were stored in a JSON file.

After extracting timestamps in the audio signal that was identified as voice, the original recording was cut into segments based on the timestamps. These audio segments were concatenated again to generate a single audio file without the non-voice segments.

The training data were selected from the segments classified as audio files having voice. These segments were manually selected to have comparable lengths of audio for each speaker and also to make sure that the segments used for training is good. The segment is considered good if it has no overlapping speech and misclassified silence considered as voice. Training data were also manually labelled.

C. Feature extraction

The speech feature used in this study is Mel Frequency Cepstral Coefficient (MFCC) extracted using the *librosa* library in python. 13 coefficients were extracted in a 25 ms frame length with an overlap of 10 ms. 12-20 cepstral coefficients is typically optimal for speech analysis, depending on the sampling rate, which is why 13 number of coefficients were used and also because the first 13 coefficients stores the most important information of the speech. It is also an efficient number of coefficient because it has a smaller dimension that makes the computation faster.

D. Speaker Segmentation

In order to detect speaker change turns in the segments, one-shot learning is implemented. A base network is trained using a few samples (52 seconds or 7% of the entire audio) of data to develop the model (as shown in Figure 2). The weights of the developed model from the base network are loaded to the siamese network. The input for the siamese network are the extracted MFCC from two windows, one for each node. The siamese network determines the similarity between the two segments based on a certain threshold. Similar segments are treated as a speech from the same speaker and are not tagged as a speaker change point, while the dissimilar segments were treated as a speech from different speakers and were tagged as a speaker change point. The audio were cut to speech segments based on the speaker change point.

There are two experimental set-ups implemented for speaker segmentation and clustering. Each experimental set-up corresponds to the one shot learning model generated using CNN with different parameters as its base network.

The generated weights from the CNN model were saved as JSON file. Afterwards, it is loaded to the siamese neural network in order to create two identical sub networks. The similarity function calculates the similarity of the pair of inputs which passed through the siamese network. It will predict whether the inputs are the same or different. The similarity function is determined by the difference of the values of the computed loss for each sub network. In a siamese network, a distance-based loss function is used called contrastive loss function. It is calculated on pairs that tries to ensure that semantically similar examples are embedded close together. The experimental set-ups are shown in Table 1.

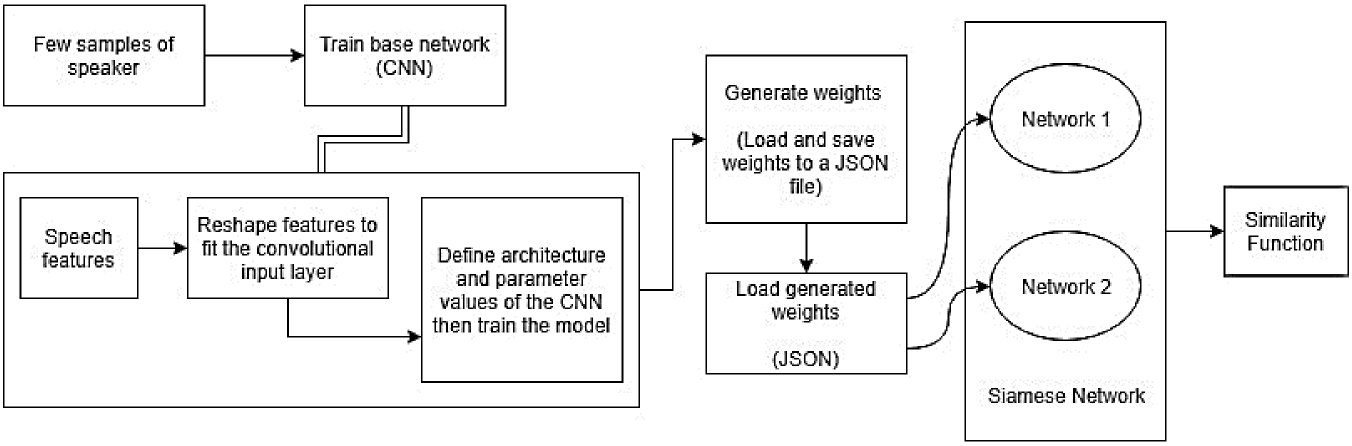
Few samples of speaker
Train base network (CNN)
Speech features
Reshape features to fit the convolutional input layer
Define architecture and parameter values of the CNN then train the model
Generate weights (Load and save weights to a JSON file)
Load generated weights (JSON)
Network 1
Network 2
Siamese Network
Similarity Function

Fig. 2 Creating a base network and loading weights to the Siamese network

Table 1. Experimental set-ups for training the base network

Experiment	Base Network Architecture	Parameters
1	Convolutional layer	Dense: 32 Activation: ReLU Kernel: 1
	Pooling layer	Pool size: 1 Dropout: 0.2
	Flattened	
	3 Hidden layers	Dense: [1]128, [2] 64, [3] 32 Activation: ReLU Dropout: 0.2
	Output layer	Dense: 4 Activation: Softmax
	Model accuracy	**94.72%**
2	Convolutional layer	Dense: 32 Activation: ReLU Kernel: 1
	Pooling layer	Pool size: 1 Dropout: 0.4
	Flattened	
	3 Hidden layers	Dense: [1]128, [2] 64, [3] 32 Activation: ReLU Dropout: 0.2
	Output layer	Dense: 4 Activation: Softmax
	Model accuracy	**84.24%**

The concatenated speech segments resulted after conducting VAD were cut into segments with a length of 25 ms. MFCC were extracted for each set of two segments and were inputted in the siamese network. The similarity score produced by the network is used by the system to determine if the previous speech segment (input 1) is spoken by the same speaker of the current speech segment (input 2). The threshold used for the similarity is 0.113, if the similarity score is less than the threshold, it is considered a speaker change, otherwise, it is considered the same speaker. The threshold is acquired by conducting a separate series of experiment, where it was found that setting a slightly higher threshold resulted to more inaccurate prediction of speaker change. This process is illustrated in Figure 3.

The contrastive loss function used in getting the similarity score is categorical cross-entropy given by the equation:

$$-(ylog(p) + (1 - y)\log(1 - p)) \qquad (1)$$

where y is the binary indicator of the correct classification for an observation and p is the probability observation o is of the correct class.

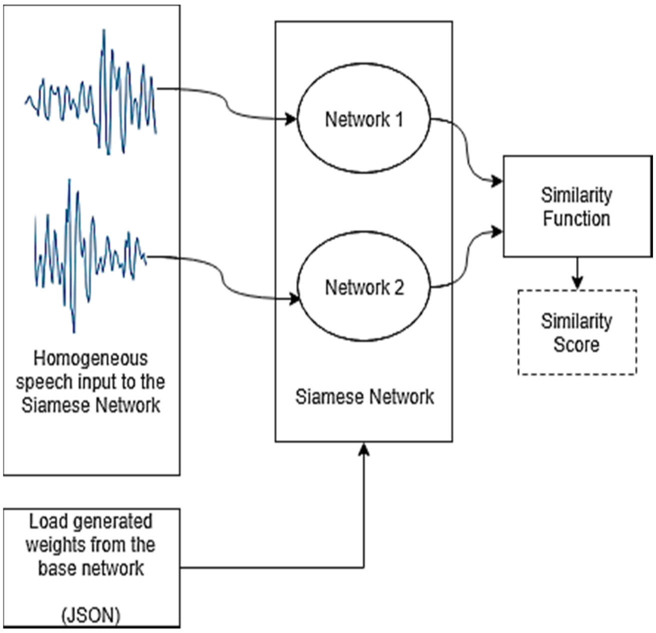
Homogeneous speech input to the Siamese Network
Network 1
Network 2
Siamese Network
Similarity Function
Similarity Score
Load generated weights from the base network (JSON)

Fig. 3 Identifying similar segments

E. Speaker Clustering

Speech segments are clustered using agglomerative hierarchical clustering algorithm. In agglomerative hierarchical clustering algorithm, each vector is assigned initially as a different cluster. The Euclidean distance between all pairs of vectors are computed and stored in a distance matrix. The two closest clusters are successively merged until only one remains, obtaining the whole clustering dendrogram as output [22]. Each identified clusters of speech segments are labelled as one speaker.

F. Evaluation

In order to evaluate the speaker diarization system, Diarization Error Rate will be used. It is the main metric that is used for speaker diarization experiments as described and used by NIST in the Rich Transcription Evaluations. It is measured as the fraction of time that is not attributed correctly to a speaker or to non-speech.

$$DER = \frac{\sum_{S=1}^{S} dur(s)\cdot(\mathrm{MAX}\left(N_{ref}(s),N_{hyp}(S)\right)-N_{correct}(S)}{\sum_{S=1}^{S} dur(s)\cdot N_{ref}} \quad (2)$$

Equation 1 is used to compute for the diarization error rate where S is the total number of speaker segments where both reference and hypothesis files contain the same speaker/s pair/s. It is obtained by collapsing together the hypothesis and reference speaker turns. The terms $N_{ref}(s)$ and $N_{hyp}(s)$ indicate the number of speaker speaking in segment s, and $N_{correct}(s)$ indicates the number of speakers that speak in segment and have been correctly matched between reference and hypothesis. Segments labelled as non-speech are considered to contain 0 speakers. When all speakers/non-speech in a segment are correctly matched the error for that segment is 0.

IV. Results and Discussion

The results of speaker diarization is manually evaluated by the researchers. The result of each experiment set-up is shown in Table 2.

Table 2. DER of each experiment

Experiment	DER	Average DER of various systems [23]
1	60.12 %	42.14 %
2	46.46 %	

In the first experiment set-up, the DER is 60.12%. The resulting segments in the experiment were 68 segments in total. For the second experiment set-up, the DER is 46.46%. The resulting segments were 154 segments.

The best performing set-up of the current diarization system is comparable to various diarization systems developed for a diarization challenge in John Hopkins University for Track 2 (systems that automatically estimates speech segments through VAD) [23]. The performance of the existing diarization systems range from 37.19% to 55.93% which used different features such as 24-MFCC, i-vectors, x-vectors and combined features (fusion) with and without Variational Bayes (VB). Approaches used were BLSTM-DNN for speaker segmentation and Agglomerative Hierarchical Clustering for speaker clustering.

During the validation phase, it can be observed that the system was more able to capture and segment correctly the utterances from female speakers than utterances made by the male speakers. It can also be observed that overlap speeches were not considered by the system.

Another observation was the false detection in speaker change during the abrupt change in the amplitude in the signal. When there is a sudden spike and sudden noise, the system treats it as a speaker change.

The performance of the base network used may have also affected the diarization performance of the system. Lastly, implosive words were being segmented or was treated as a speaker change point. Utterances with low energy or low in volume were also treated as silence, which made a negative effect in the system.

V. Conclusion and Future work

In this study, a Filipino speaker diarization is developed for meeting room conversations using one shot learning implemented with siamese neural network. The speech feature used for segmentation and clustering was MFCC.

Experiments showed that it one shot learning can be an approach to develop a Filipino speaker diarization system, however, it still needs a lot of improvement.

For future work, it is recommended to work with other speech features such as pitch and LPCC.

References

[1] Zhang, A., Wang, Q., Zhu, Z., Paisley, J., & Wang, C. (2018). Fully supervised speaker diarization. arXiv preprint arXiv:1810.04719.

[2] Anguera, X., Bozonnet, S., Evans, N., Fredouille, C., Friedland, G., & Vinyals, O. (2012). Speaker diarization: A review of recent research. IEEE Transactions on Audio, Speech, and Language Processing, 20(2), 356-370.

[3] Yoav, R. (2019, February). Speaker Diarization with Kaldi. Retrieved from https://towardsdatascience.com/speaker-diarization-with-kaldi-e30301b05cc8 by April 8, 2019. (footnote)

[4] NIST. (2009). “The NIST Rich Transcription 2009 (RT’09) evaluation”. http://www.itl.nist.gov/iad/mig/tests/rt/2009/docs/rt09-meeting-evalplan-v2.pdf (footnote)

[5] Basu, J., Khan, S., Roy, R., Pal, M., Basu, T., Bepari, M. S., & Basu, T. K. (2016, October). An overview of speaker diarization: Approaches, resources and challenges. In 2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA) (pp. 166-171). IEEE.

[6] Wang, Q., Downey, C., Wan, L., Mansfield, P. A., & Moreno, I. L. (2018, April). Speaker diarization with lstm. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5239-5243). IEEE.

[7] Fei-Fei, L., Fergus, R., & Perona, P. (2006). One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence, 28(4), 594-611.

[8] Koch, G., Zemel, R., & Salakhutdinov, R. (2015, July). Siamese neural networks for one-shot image recognition. In ICML deep learning workshop (Vol. 2).

[9] Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., & Lillicrap, T. (2016). One-shot learning with memory-augmented neural networks. arXiv preprint arXiv:1605.06065.

[10] Moattar, M. H., & Homayounpour, M. M. (2012). A review on speaker diarization systems and approaches. Speech Communication, 54(10), 1065-1103.

[11] Wooters, C., Fung, J., Peskin, B., & Anguera, X. (2004, November). Towards robust speaker segmentation: The ICSI-SRI fall 2004 diarization system. In RT-04F Workshop (Vol. 23, p. 23).

[12] Nguyen, P., Rigazio, L., Moh, Y., & Junqua, J. C. (2002, May). Rich transcription 2002 site report. Panasonic speech technology laboratory (PSTL). In Proc. Rich Transcription Workshop (RT-02).

[13] Sun, H., Ma, B., Khine, S. Z. K., & Li, H. (2010, March). Speaker diarization system for RT07 and RT09 meeting room audio. In 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 4982-4985). IEEE.

[14] Zhu, X., Barras, C., Meignier, S., & Gauvain, J. L. (2005). Combining speaker identification and BIC for speaker diarization. In Ninth European Conference on Speech Communication and Technology.

[15] Kristjansson, T., Deligne, S., & Olsen, P. (2005). Voicing features for robust speech detection. In Ninth European Conference on Speech Communication and Technology.

[16] Anguera, X., Wooters, C., Peskin, B., & Aguiló, M. (2005, July). Robust speaker segmentation for meetings: The ICSI-SRI spring 2005 diarization system. In International Workshop on Machine Learning for Multimodal Interaction (pp. 402-414). Springer, Berlin, Heidelberg.

[17] Meignier, S., Bonastre, J. F., & Igounet, S. (2001). E-HMM approach for learning and adapting sound models for speaker indexing. In 2001: A Speaker Odyssey-The Speaker Recognition Workshop.

[18] Hain, T., Johnson, S. E., Tuerk, A., Woodland, P. C., & Young, S. J. (1998, February). Segment generation and clustering in the HTK broadcast news transcription system. In Proc. 1998 DARPA Broadcast News Transcription and Understanding Workshop (pp. 133-137).

[19] Gauvain, J. L., Lamel, L. F., & Adda, G. (1998). Partitioning and transcription of broadcast news data. In Fifth International Conference on Spoken Language Processing.

[20] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1994). Signature verification using a" siamese" time delay neural network. In Advances in neural information processing systems (pp. 737-744).

[21] Rao, S. J., Wang, Y., & Cottrell, G. W. (2016). A Deep Siamese Neural Network Learns the Human-Perceived Similarity Structure of Facial Expressions Without Explicit Categories. In CogSci.

[22] Khoury, E., El Shafey, L., Ferras, M., & Marcel, S. (2014, June). Hierarchical speaker clustering methods for the nist i-vector challenge. In Odyssey: The Speaker and Language Recognition Workshop (pp. 254-259).

[23] Sell, G., Snyder, D., McCree, A., Garcia-Romero, D., Villalba, J., Maciejewski, M., ... & Khudanpur, S. (2018). Diarization is Hard: Some Experiences and Lessons Learned for the JHU Team in the Inaugural DIHARD Challenge. In Interspeech (pp. 2808-2812).