# A MULTITASK LEARNING FRAMEWORK FOR SPEAKER CHANGE DETECTION WITH CONTENT INFORMATION FROM UNSUPERVISED SPEECH DECOMPOSITION

[1]*Hang Su,* [1]*Danyang Zhao,* [1]*Long Dang,* [2]*Minglei Li,* [1]*Xixin Wu,* [1]*Xunying Liu and* [1]*Helen Meng*

[1] The Chinese University of Hong Kong, Hong Kong SAR, China
[2] Huawei Cloud, Shenzhen, China

## ABSTRACT

Speaker Change Detection (SCD) is a task of determining the time boundaries between speech segments of different speakers. SCD system can be applied to many tasks, such as speaker diarization, speaker tracking, and transcribing audio with multiple speakers. Recent advancements in deep learning lead to approaches that can directly detect the speaker change points from audio data at the frame-level based on neural network models. These approaches may be further improved by utilizing speaker information in the training data, and utilizing content information extracted in an unsupervised manner. This work proposes a novel framework for the SCD task, which utilizes a multitask learning architecture to leverage speaker information during the training stage, and adds the content information extracted from an unsupervised speech decomposition model to help detect the speaker change points. Experiment results show that the architecture of multitask learning with speaker information can improve the performance of SCD, and adding content information extracted from unsupervised speech decomposition model can further improve the performance. To the best of our knowledge, this work outperforms the state-of-the-art SCD results [1] on the AMI dataset.

***Index Terms***— Multitask Learning, Speaker Change Detection, Unsupervised Speech Decomposition

## 1. INTRODUCTION

Speaker Change Detection (SCD) is a task of determining the time boundaries between speech segments of different speakers. Applying proper SCD techniques can benefit not only the speaker diarization systems [2], which aims at detecting "who spoke when", but also the tasks of transcribing audio with multiple speakers [3, 4] and speaker tracking in multimedia data processing [5].

Some approaches to detecting the speaker change points in an audio stream are based on the distance between two adjacent sliding windows. More specifically, two adjacent windows are shifted along the audio, and the distance between them is computed based on a pre-defined distance metric. The change point is determined if the distance between two adjacent windows is larger than a fine-tuned threshold. In order to apply this approach, proper features that represent the windowed signal, as well as proper distance metrics are important. Commonly used features include Mel-frequency cepstral coefficients (MFCC) [6, 7], I-vectors [8, 9] and features extracted from neural networks [10, 11, 12]. As regards distance metrics, common ones are the Bayesian information criterion [13, 14], the generalized likelihood ratio [15] and the Kullback-Leibler divergence [16].

With the recent advancements of deep learning, many new approaches utilize trained neural network models from labeled audio data to directly predict the speaker change points, which usually yield state-of-the-art results. Some approaches build on a pre-trained Automatic Speech Recognition (ASR) system, which first split the speech into word-level segments and then train a neural network model to detect whether there's a speaker change between these segments [3, 17]. However, these approaches rely on a good ASR system to narrow down the search space of speaker change point, which is also their limitation, because the performance of ASR system will affect the word-level segmentation. Some methods that do not require a pre-trained ASR system are also proposed. These methods directly detect the speaker change point from audio data in frame-level based on neural network models, such as Deep Neural Networks (DNN) [18], Convolutional Neural Network (CNN) [2, 19, 20] and Long Short-Term Memory (LSTM) [21, 1, 22]. We believe that there is space for further improvement of these methods. On the one hand, speaker information in the training data are rarely used in these methods but can be leveraged. On the other hand, content information of speech should be useful for SCD task, but are also rarely used in these methods. In particular, to obtain spoken content information, we should investigate the use of unsupervised methods because manual transcription is costly.

This work builds upon a state-of-the-art system that utilizes a neural network to predict the speaker change points from audio data at the frame-level [1]. A novel framework is proposed to leverage speaker information during training by using a multitask learning architecture. Meanwhile, spoken content information of speech is extracted based on an unsupervised speech decomposition method [23]. The extracted content information is then fed into the multitask learning framework to help detect the speaker change points.

The paper is organized as follows: Section 2 introduces how the baseline system work for the SCD task. Section 3 illustrates our proposed framework for improving the baseline system. Section 4 shows the details of experiments. Section 5 presents the conclusions of this work.

## 2. BASELINE SYSTEM

### 2.1. SCD as a sequence labeling problem

The state-of-the-art system of utilizing neural network to predict the speaker change points from audio data at the frame-level treats the SCD task as a sequence labeling task [1]. The input of this sequence labeling task is the sequence of feature vectors $X = \{x_1, x_2, ..., x_T\}$ extracted from the audio recording ($T$ is the sequence length), and the expected output is the corresponding sequence of labels $y = \{y_1, y_2, ..., y_T\}$ with $y_t \in \{0, 1\}$. If there is no speaker change at time step $t$, then $y_t = 0$, otherwise $y_t = 1$. The objective is then to find a function $f : X \longrightarrow Y$ that matches a feature sequence to a label sequence.

### 2.2. Model architecture

The neural network is applied to estimate the function $f : X \longrightarrow Y$. As shown in Fig. 1, two kinds of inputs are independently used to perform the SCD task. The first is a 57-dimensional MFCC extracted every 10ms on a 20ms window. The second is the waveform, which is fed to SincNet convolutional layers [24] to form an end-to-end system. MFCC or the output of SincNet layers are then passed into two Bi-LSTM layers and three fully connected dense layers whose weights are shared across the sequence. The activation function of the first two dense layers is the tanh activation function, and the last dense layer is a linear layer followed by the softmax function to be the output layer.

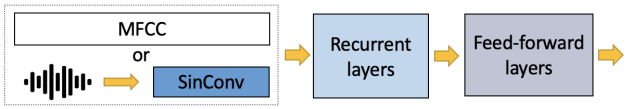


**Fig. 1**. Model architecture of the baseline system

### 2.3. Training

The long audio sequences are split into fixed-length (2s) overlapping sequences as shown in Fig. 2. Random noise which includes technical noises (Dual Tone Multiple Frequencies (DTMF) tones, fax machine noises, etc) and ambient sounds (car idling, footsteps, animal noises, etc) is then added to each split sequence of audio for data augmentation. Then, the MFCC or waveform of the split sequences are randomly put into the model mentioned in section 2.2. The label of output is a sequence of ones (with speaker change) and zeros (not speaker change). Note in Fig. 2 that 200ms on both sides of the exact change point are all labeled as ones. Finally, cross entropy loss is used to train the model.

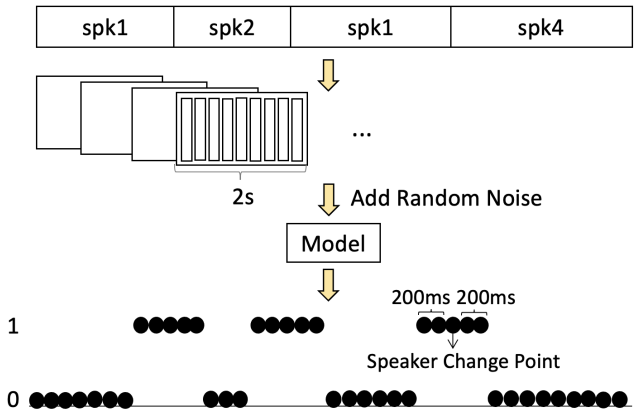


**Fig. 2**. Training process for speaker change detection

### 2.4. Prediction

As shown in Fig. 3, the long audio sequences are also split into fixed-length (2s) overlapping sequences. Then, the MFCC or waveform of split sequences are input to the trained model to obtain a sequence of scores between 0 and 1. Since the input sequences are overlapped, each time step (0.1s) has multiple candidate scores. These candidate scores are averaged to obtain the final prediction score of a time step. Final prediction scores of all time steps of a long audio sequence form a curve, where the local maxima (peak) greater than a tunable threshold $\theta$ are marked as speaker change points.

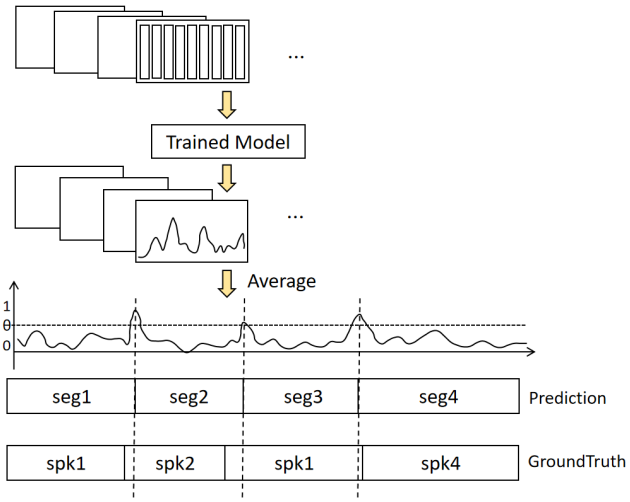


**Fig. 3**. Prediction process for speaker change detection

## 3. APPROACH

Fig. 4 illustrates our proposed approach, which mainly consists of a multitask learning step, and an unsupervised speech decomposition step. We will elaborate on these two parts below.

### 3.1. Multitask learning for utilizing speaker information

Intuitively, speaker information would benefit speaker change detection. In order to fully utilize the speaker information that exist in the training data of the SCD task, a multitask learning architecture is proposed. As shown in Fig. 4 (a), different from the baseline system, a new speaker branch is added to learn the speaker information after the recurrent layers, instead of only using one SCD branch to predict the speaker change point.

Two methods are proposed to learn the speaker information through the speaker branch. The first method is utilizing the speaker branch to predict speaker ID information. The output of the feed-forward layers of the speaker branch is a sequence of predicted speaker ID vectors in frame-level, and the output of the feed-forward layers of the SCD branch is the same as the output of the baseline system mentioned in section 2.3. Cross entropy loss of the two branches are independently computed, and then added together as the total loss. This method forces the recurrent layers to learn a better hidden representation to identify speakers, which should benefit the performance in SCD.

The second method use triplet loss [25] to be the loss of speaker branch. The triplet loss is shown as equation (1)

$$\mathcal{L} = \max(d(a,p) - d(a,n) + margin, 0) \tag{1}$$

where $a$ is an anchor input, $p$ is a positive input of the same speaker as $a$, $n$ is a negative input of a different speaker from $a$, $d(x,y)$ is the distance between $x$ and $y$, and $margin$ is a positive constant which can be set manually. Minimizing the triplet loss is equivalent to making $d(a,p) - d(a,n) + margin$ close to 0. Thus, the triplet loss would make the embedding of the same speaker more similar and make the embedding of different speakers more different. The cross entropy loss of the SCD branch and the triplet loss of the speaker branch are added together as the total loss. This method forces the recurrent layers to learn a better hidden representation to distinguish between speakers, which will also benefit the task of SCD.

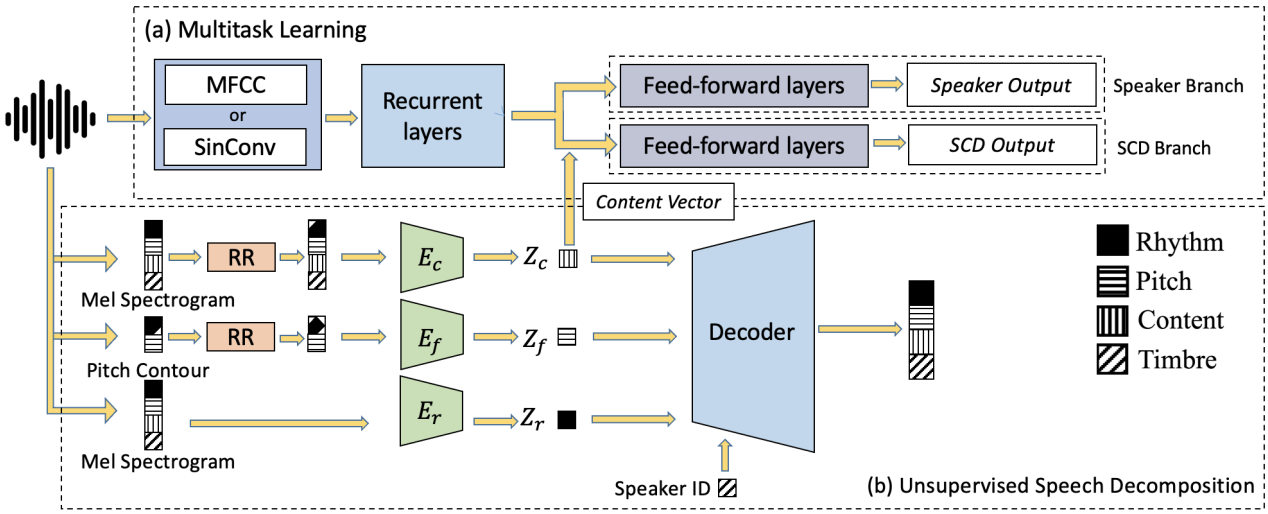


**Fig. 4**. Framework of proposed method

### 3.2. Unsupervised speech decomposition for extracting content information

Spoken content information is also important for the SCD task, because the human is able to do the SCD task only based on dialog text data without audio. In this work, we use an unsupervised speech decomposition method proposed by Qian et al. [23] to extract content information of speech – this method decomposes spoken information into rhythm, pitch, timbre, and content in an unsupervised manner. As shown in Fig. 4 (b), the model mainly consists of a content encoder ($E_c$), a pitch encoder ($E_f$), a rhythm encoder ($E_r$), and a decoder whose input is the speaker ID (timbre) information and three encoder layers' outputs. The mel spectrogram is fed into the Random Resampling (RR) module before it is fed into the content encoder ($E_c$). The RR module first randomly splits the input signal into segments, then randomly stretches or squeezes each segment before concatenating them together. The purpose of setting this module is to disrupt the time information, which makes the output signal of the RR module contains partially and randomly contaminated rhythm information, while keeping other information intact. The pitch contour, which is pitch frequency against time, is fed into the Random Resampling (RR) module before being fed into the pitch encoder ($E_f$). The pitch contour contains pitch information and parts of the rhythm information (the reason why the rhythm block of pitch contour in Fig. 4 (b) misses a corner). After the pitch contour passes through the RR module, the rhythm information is disrupted and pitch information keeps intact. Furthermore, the mel spectrogram which contains all speech information is fed into the rhythm encoder ($E_r$) directly. Then, the output of three encoders and speaker ID information are input to a decoder module. Finally, the mean square error loss is computed between the output of the decoder module and the original mel spectrogram.

The unsupervised speech decomposition method essentially performs several key functions – first, timbre information is directly fed to the decoder, so timbre can be ignored by all the encoders. Second, only the rhythm encoder has access to the full rhythm information, while other types of information can be obtained elsewhere. Hence the rhythm encoder will prioritize passing the rhythm information, and remove other information given bottleneck constraints. Hence, the content encoder becomes the only module that encodes the spoken content information. Finally, with only the rhythm encoder encoding rhythm information and only the content encoder encoding spoken content information, the pitch encoder must encode the pitch information. Therefore, after the training of unsupervised speech decomposition model, the content encoder, rhythm encoder, and pitch encoder are able to extract the content, rhythm, and pitch information of speech respectively.

### 3.3. Training and prediction

First, the audio clips that only containing one speaker in the training data of the SCD task are used to train the unsupervised speech decomposition model. Then, same as the baseline system, the long audio sequences are split into fixed-length (2s) overlapping sequences with 0.4 second shift. The 2-second audio clips are randomly fed into the multitask learning framework to train the SCD model. Meanwhile, these audio clips are also fed into the pre-trained unsupervised speech decomposition model to extract the content information of the corresponding clips, which are the output vectors of the content encoder. Since the content information of speech is useful for detecting the speaker change point, the content vectors are then concatenated together with the input of feed-forward layers of the SCD branch. The loss from the two branches are added to form the total loss. Parameters of the pre-trained unsupervised speech decomposition model are not updated during the training of the SCD model.

The prediction processes are nearly the same as the baseline system as shown in Fig. 3. The only difference is that the "trained model" in Fig. 3 becomes the proposed model shown in Fig. 4. The audio clips are fed into both the trained SCD model with multitask learning architecture and the trained unsupervised decomposition model. The extracted content vectors are concatenated together with the input of the feed-forward layers in the SCD branch. The output of SCD branch will be further processed to obtain the segmentation boundaries as mentioned in section 2.4.

## 4. EXPERIMENTS

### 4.1. Dataset

Experiments are conducted on the AMI corpus [26], which is an open-source dataset with 100 hours of conversational recordings. Each conversation is conducted in English with 4 to 5 speakers in the meeting domain. The information of "who spoke when" is provided in the annotation. The training, validation, and test sets are exactly the same as the baseline system. The training set includes

118 conversations (about 70 hours), the validation set includes 26 conversations (about 15 hours) and the test set includes 24 conversations (about 15 hours).

### 4.2. Evaluation metric

Coverage, purity, and F1 are used as the evaluation metrics for this task. Given R the set of reference speech turns, and H the set of predicted segments, the coverage is calculated as equation (2)

$$coverage(R, H) = \frac{\sum_{r \in R} max_{h \in H} |r \cap h|}{\sum_{r \in R} |r|} \quad (2)$$

where $|r|$ is the duration of segment $r$ and $r \cap h$ is the intersection of segments r and h. Purity is the dual metric where the role of R and H are interchanged. Detecting too many speaker changes would result in high purity but low coverage, while missing many speaker change points would result in high coverage but low purity. F1 is the harmonic average of coverage and purity.

### 4.3. Experimental setup

The reproduced baseline system and the proposed methods are compared in our experiments. All systems for the SCD task use the Adam [27] as optimizer with learning rate of 0.0005. The batch size of training is 512 when taking MFCC as the input, and 128 when taking waveform as the input. The threshold $\theta$ (mentioned in section 2.4) and model using for test are selected on the validation set. The best model is selected in terms of the maximal F1 score. Given the best model, the $\theta$ is selected in terms of the maximal coverage score whose corresponding purity score is marginally larger or equal to 85%. The settings of SincNet convolutional layers, LSTM layers, dense layers and the methods for adding random noise follow exactly the same as the work of Bredin et al. [1] (our baseline). The triplet loss in the proposed methods is calculated in each batch, which is based on the output of the last feed-forward layer in the speaker branch. Every frame in a batch is treated as the anchor once. The positive input is selected as a random frame with the same speaker as the corresponding anchor. The negative input is selected as a random frame with a different speaker from the corresponding anchor. The margin in triplet loss is set to 1. The Euclidean distance is used to calculate the distance between frames. The implementation of pre-training the unsupervised speech decomposition model follows the work of Qian et al. [23], except that the hop length for the extraction of mel spectogram and pitch contour is set at 10ms, which is consistent with the extracted MFCC.

### 4.4. Results and discussions

Table 1 shows the experiment results with MFCC as the input, and Table 2 shows the results of using waveform as the input. The F1 results from Bredin et al. [1] for the validation set and test set are $81.94\%$ and $83.71\%$ when using the MFCC as the input, and $86.63\%$ and $87.19\%$ when using waveform as the input. The baseline result presented in this work is reproduced based on the publication. As shown in Tables 1 and 2, the F1 of the reproduced baseline is comparable or even higher than those originally reported. Both methods for utilizing speaker information with multitask learning architecture bring improvement over the baseline for both input forms. This demonstrates that utilizing speaker information with the proposed multitask learning architecture can improve the performance of SCD task. We also tried to use the speaker triplet loss to pretrain a neural network model with 2 LSTM layers and 2 dense layers sequentially, and then adding another 2 dense layers after the original 2 dense layers to fine-tune the model on the SCD task. The result (please see the second row in both tables) is better than the baseline, but worse than the proposed multitask learning architecture. Finally, as shown in Tables 1 and 2, adding spoken content information into the multitask learning framework further improves the performance of the SCD task for both input forms. To the best of our knowledge, the proposed method has attained a new state-of-the-art result on the AMI dataset for the SCD task in terms of the F1 score of purity and coverage.

**Table 1**. Results of using MFCC as the input on both validation and test set in terms of purity (%), coverage (%), and F1 (%).

| | Validation | | | Test | | |
|---|---|---|---|---|---|---|
| | **Purity** | **Coverage** | **F1** | **Purity** | **Coverage** | **F1** |
| Baseline | 85.01 | 79.90 | 82.27 | 86.54 | 80.72 | 83.53 |
| Pretrain+finetune | 85.08 | 80.78 | 82.87 | 87.04 | 81.18 | 84.01 |
| Multitask (spk id) | 85.07 | 79.98 | 82.44 | 86.84 | 82.97 | 84.34 |
| Multitask (triplet) | 85.02 | 81.14 | 83.03 | 86.04 | 83.31 | 84.65 |
| Triplet + content | 85.04 | 81.68 | **83.33** | 86.16 | 84.56 | **85.35** |

**Table 2**. Results of using waveform as the input on both validation and test set in terms of purity (%), coverage (%), and F1 (%).

| | Validation | | | Test | | |
|---|---|---|---|---|---|---|
| | **Purity** | **Coverage** | **F1** | **Purity** | **Coverage** | **F1** |
| Baseline | 85.38 | 89.49 | 87.39 | 85.62 | 89.71 | 87.62 |
| Pretrain+finetune | 85.00 | 90.51 | 87.67 | 85.16 | 90.92 | 87.95 |
| Multitask (spk id) | 85.00 | 91.74 | 88.24 | 85.61 | 91.04 | 88.24 |
| Multitask (triplet) | 85.26 | 91.49 | 88.27 | 85.66 | 91.02 | 88.26 |
| Triplet + content | 85.00 | 91.92 | **88.32** | 85.68 | 91.75 | **88.61** |

## 5. CONCLUSIONS

This work proposes a novel neural network framework for detecting speaker change points from audio data at the frame level. First, a multitask learning architecture is designed to utilize the speaker information in training data. A speaker branch is added based on the baseline system to help the model learn to identify speakers, or learn to distinguish different speakers using triplet loss. Experiments show that both methods for utilizing speaker information with multitask learning architecture outperform the baseline system, which demonstrates that utilizing speaker information with the proposed multitask learning architecture can improve the performance of the SCD task. Then, a spoken content vector extracted from a pre-trained unsupervised speech decomposition model is added to the multitask learning architecture to help predict the speaker change points. Experimental results show that this can further improve the performance of the SCD task. Overall, to the best of our knowledge, the proposed approach has achieved a new state-of-the-art result on the AMI dataset for the SCD task.

## 6. ACKNOWLEDGMENT

This work is partially supported by the CUHK TDLEG Grant (2016-2019) and a grant from the CUHK Stanley Ho Big Data Decision Analytics Research Centre.

## 7. REFERENCES

[1] Hervé Bredin, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill, "Pyannote. audio: neural building blocks for speaker diarization," in *ICASSP 2020-2020 IEEE International Conference*

*on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7124–7128.

[2] Marek Hrúz and Zbyněk Zajíc, "Convolutional neural network for speaker change detection in telephone speaker diarization system," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4945–4949.

[3] Hagai Aronowitz and Weizhong Zhu, "Context and uncertainty modeling for online speaker change detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8379–8383.

[4] Daben Liu and Francis Kubala, "Fast speaker change detection for broadcast news transcription and indexing," in *Sixth European Conference on Speech Communication and Technology*, 1999.

[5] Soonil Kwon and Shrikanth S Narayanan, "Speaker change detection using a new weighted distance measure," in *Seventh international conference on spoken language processing*, 2002.

[6] Sylvain Meignier, Daniel Moraru, Corinne Fredouille, Jean-François Bonastre, and Laurent Besacier, "Step-by-step and integrated approaches in broadcast news speaker diarization," *Computer Speech & Language*, vol. 20, no. 2-3, pp. 303–330, 2006.

[7] C. Barras, Xuan 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. 1505–1512, 2006.

[8] Brecht Desplanques, Kris Demuynck, and Jean-Pierre Martens, "Factor analysis for speaker segmentation and improved speaker diarization," in *16th annual conference of the international speech communication association (INTERSPEECH 2015)*. International Speech Communication Association (ISCA), 2015, pp. 3081–3085.

[9] Leonardo V Neri, Hector NB Pinheiro, Ren Tsang, George D da C Cavalcanti, and André G Adami, "Speaker segmentation using i-vector in meetings domain," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5455–5459.

[10] Renyu Wang, Mingliang Gu, Lantian Li, Mingxing Xu, and Thoms Fang Zheng, "Speaker segmentation using deep speaker vectors for fast speaker change scenarios," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5420–5424.

[11] Hervé Bredin, "TristouNet: triplet loss for speaker turn embedding," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 5430–5434.

[12] Arindam Jati and Panayiotis G Georgiou, "Speaker2Vec: Unsupervised learning and adaptation of a speaker manifold using deep neural networks with an evaluation on speaker segmentation.," in *INTERSPEECH*, 2017, pp. 3567–3571.

[13] Scott Chen, Ponani Gopalakrishnan, et al., "Speaker, environment and channel change detection and clustering via the bayesian information criterion," in *Proc. DARPA broadcast news transcription and understanding workshop*. Virginia, USA, 1998, vol. 8, pp. 127–132.

[14] Mauro Cettolo, Michele Vescovi, and Romeo Rizzi, "Evaluation of BIC-based algorithms for audio segmentation," *Computer Speech & Language*, vol. 19, no. 2, pp. 147–170, 2005.

[15] Herbert Gish, Man-Hung Siu, and Jan Robin Rohlicek, "Segregation of speakers for speech recognition and speaker identification.," in *ICASSP*, 1991, vol. 91, pp. 873–876.

[16] Matthew A Siegler, Uday Jain, Bhiksha Raj, and Richard M Stern, "Automatic segmentation, classification and clustering of broadcast news audio," in *Proc. DARPA speech recognition workshop*, 1997, vol. 1997.

[17] Leda Sarı, Samuel Thomas, Mark Hasegawa-Johnson, and Michael Picheny, "Pre-training of speaker embeddings for low-latency speaker change detection in broadcast news," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6286–6290.

[18] Vishwa Gupta, "Speaker change point detection using deep neural nets," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4420–4424.

[19] Marek Hrúz and Marie Kunešová, "Convolutional neural network in the task of speaker change detection," in *International Conference on Speech and Computer*. Springer, 2016, pp. 191–198.

[20] Lukas Mateju, Petr Cerva, and Jindrich Zdánskỳ, "An approach to online speaker change point detection using DNNs and WFSTs," in *Interspeech*, 2019, pp. 649–653.

[21] Ruiqing Yin, Hervé Bredin, and Claude Barras, "Speaker change detection in broadcast TV using bidirectional long short-term memory networks," in *Interspeech 2017*. ISCA, 2017.

[22] Aidan OT Hogg, Christine Evers, Alastair H Moore, and Patrick A Naylor, "Overlapping speaker segmentation using multiple hypothesis tracking of fundamental frequency," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1479–1490, 2021.

[23] Kaizhi Qian, Yang Zhang, Shiyu Chang, Mark Hasegawa-Johnson, and David Cox, "Unsupervised speech decomposition via triple information bottleneck," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7836–7846.

[24] Mirco Ravanelli and Yoshua Bengio, "Speaker recognition from raw waveform with SincNet," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 1021–1028.

[25] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[26] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al., "The AMI meeting corpus: A pre-announcement," in *International workshop on machine learning for multimodal interaction*. Springer, 2005, pp. 28–39.

[27] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.