Multi-target Filter and Detector for Unknown-number Speaker Diarization

Chin-Yi Cheng, Hung-Shin Lee, Yu Tsao, *Senior Member, IEEE*, and Hsin-Min Wang, *Senior Member, IEEE*

***Abstract*—A strong representation of a target speaker can aid in extracting important information regarding the speaker and detecting the corresponding temporal regions in a multi-speaker conversation. In this study, we propose a neural architecture that simultaneously extracts speaker representations that are consistent with the speaker diarization objective and detects the presence of each speaker frame by frame, regardless of the number of speakers in the conversation. A speaker representation (known as a z-vector) extractor and frame-speaker contextualizer, which is realized by a residual network and processing data in both the temporal and speaker dimensions, are integrated into a unified framework. Testing on the CALLHOME corpus reveals that our model outperforms most methods presented to date. An evaluation in a more challenging case of concurrent speakers ranging from two to seven demonstrates that our model also achieves relative diarization error rate reductions of 26.35% and 6.4% over two typical baselines, namely the traditional x-vector clustering model and attention-based model, respectively.**

***Index Terms*—speaker diarization, speaker representations**

I. Introduction

SPEAKER diarization is the process of determining when individual speakers are active in a recording. The aim is to generate a diary of the presence of each speaker at each point in time. This technique has been extensively used for speech processing in various scenarios, such as conference conversations, broadcast news, debates, and cocktail parties [1]. However, the resilience of speaker diarization remains weak owing to the challenges that are posed by variations in the recording channels, environment, reverberation, ambient noise, and the number of speakers [2].

Over the past decade, researchers have tackled diarization problems using probabilistic models [3] or neural networks [4]. Several methods involve two steps, segmentation and clustering. In the segmentation step, a 1.5-second sliding window (with a 50% overlap) is typically used to divide a session into a sequence of short segments. Subsequently, a speaker model is used to extract the speaker representation (e.g., the x-vector [5], [6], [7], [8], i-vector [9], [10], or d-vector [11], [12]) of each segment. Segments with highly homogeneous characteristics form a group during the clustering process. Different clustering techniques have been applied according to various similarity measures, such as probabilistic linear discriminant analysis (PLDA) and cosine similarity [13], [14], [15], [10]. For example, agglomerative hierarchical clustering (AHC) and spectral clustering (SC) were used in [6], [10] and [16], [17], respectively. The unbounded interleaved-state recurrent neural network (UIS-RNN), which originated from both the Gaussian mixture model (GMM) [18], [19] and hidden Markov model (HMM) [7], was used in [12]. Moreover, several post-processing methods, such as Variational Bayes (VB) [20] and the LSTM-based method [21], have been applied to refine the initial diarization results.

Several recent studies [22], [23], [24], [25] have focused on end-to-end (E2E) speaker diarization. Fujita *et al.* [22] reformulated the diarization task as a multi-label classification problem and used the permutation-invariant training (PIT) [26] technique. Moreover, in [27], reliable speaker representations were derived using a selector to assist a voice activity detector in diarizing a session. Self-attention [28], [29] and frame selection [30] have also been used in E2E speaker diarization.

The traditional segmentation-clustering method cannot handle overlapping speech in a session effectively. Target speaker voice activity detection (TS-VAD) [31] has achieved good performance in overlapping speech processing. It relies on an x-vector/SC procedure [5], [8] to provide first-stage timestamps of the speech of each "target" (active) speaker, which are used to extract the first-stage i-vector for each target speaker from frames in which the speaker is active. Finally, it uses the i-vectors of all speakers and MFCCs to generate the diarization results. Unfortunately, it can only be applied to sessions with a fixed number of speakers, because its neural structure contains a tensor concatenation of speaker representations. Inspired by the dual-path recurrent neural network (DPRNN) [32], [33], we propose a unified structure known as Multi-target Filter and Detector (MTFAD) that can handle conversations with various numbers of speakers using a single model. Furthermore, as the quality of the speaker representations has an impact on the diarization performance, we extend TS-VAD by using a neural filter that can directly extract speaker representations that are suitable for the diarization task.

The main contributions of this study are threefold: First, we significantly extend the practical scope of TS-VAD while inheriting its excellent performance in the speaker diarization task. In this sense, MTFAD offers an advantage over TS-VAD because it does not set a limit on the number of speakers in a session. In addition to expanding the practical use, because the training data of different speaker numbers can be used together to train a single model, the model is more powerful than multiple separate models that are each trained with the data of a specific number of speakers. Second, we design a filter that can be jointly trained with the diarization model to extract speaker representations. Using this filter, we achieve better diarization performance and avoid pretraining of the i-vector extractor. Third, unlike in certain previous studies [6], [12], which dealt with only non-overlapping speech, our model also performs well on data containing overlapping speech regions.

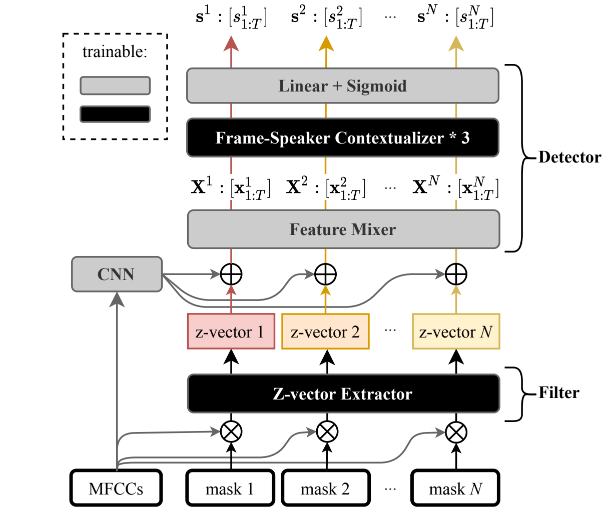
$\mathbf{s}^1 : [s^1_{1:T}]$
$\mathbf{s}^2 : [s^2_{1:T}]$
···
$\mathbf{s}^N : [s^N_{1:T}]$
trainable:
Linear + Sigmoid
Frame-Speaker Contextualizer * 3
Detector
$\mathbf{X}^1 : [\mathbf{x}^1_{1:T}]$
$\mathbf{X}^2 : [\mathbf{x}^2_{1:T}]$
···
$\mathbf{X}^N : [\mathbf{x}^N_{1:T}]$
Feature Mixer
CNN
z-vector 1
z-vector 2
···
z-vector N
Z-vector Extractor
Filter
MFCCs
mask 1
mask 2
···
mask N

Fig. 1. Structure of MTFAD, where $\oplus$, N, and T denote the concatenation operator, number of speakers, and number of frames, respectively. $\mathbf{X}^i$ and $\mathbf{s}^i$ are the Speaker-aware frames and diarization scoring vector for speaker i, respectively. Both the black and gray blocks are trainable. $\otimes$ is an element-wise product operator of MFCCs, and the binary masks (labels of speaker occurrences) are obtained in the first step (x-vector/AHC diarization). Each speaker representation (e.g., z-vector) goes through the Feature Mixer with the frame-level MFCCs separately to generate the Speaker-aware frames.

II. Multi-target Filter and Detector

A. Framework

Inspired by TS-VAD [31], we propose a two-step diarization method known as MTFAD that can adapt to different numbers of speakers. MTFAD not only addresses the main weakness of TS-VAD, which can handle only a fixed number of speakers (e.g., four in [31]) in a session, but also uses an improved speaker representation. As illustrated in Fig. 1, the first step of MTFAD (the Filter) relies on the traditional x-vector/AHC diarization method to generate the initial timestamps for each speaker (i.e., the labels of speaker occurrences). With the timestamps, the frames corresponding to each speaker are used to extract the speaker representation. In this case, the representation may be the traditional i-vector and x-vector, or our specially designed z-vector for diarization (see Section II-C). The second step of MTFAD (the Detector) requires two inputs: the frame-level MFCCs and the representation of each speaker. The frame-level MFCCs first go through a four-layer convolutional neural network (CNN). The convolutionized MFCCs and each speaker representation are concatenated as the input of the Detector. The output of the Detector is the final diarization result for each speaker.

In TS-VAD, the second step is implemented using BiLSTM. First, the first two layers of the BiLSTM take the frame-level MFCCs and four i-vectors as the input, and output four speaker detection (SD) vector sequences. Subsequently, the four SD vector sequences are concatenated along the feature dimension and passed through the third layer of the BiLSTM to generate the final diarization result for each speaker. It is this concatenation that causes the TS-VAD to only handle four-speaker recordings. Furthermore, in the filter part, TS-VAD uses the i-vector as the speaker representation, which is defeated by the x- and z-vectors in our experiments. The detailed MTFAD structure is described in the following subsections.

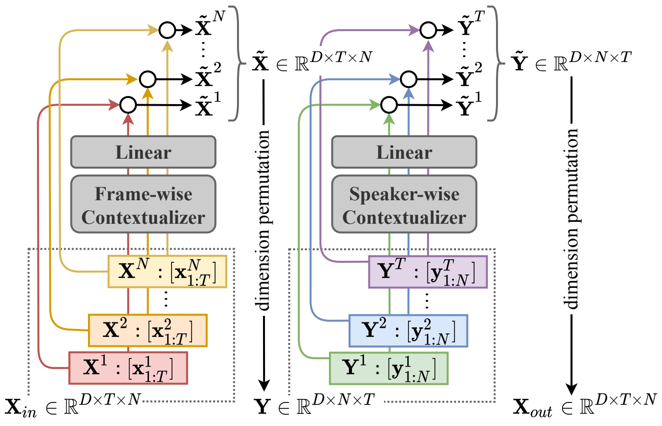
$\tilde{\mathbf{X}}^N$
$\tilde{\mathbf{X}}^2$
$\tilde{\mathbf{X}}^1$
$\tilde{\mathbf{X}} \in \mathbb{R}^{D\times T\times N}$
Linear
Frame-wise Contextualizer
$\mathbf{X}^N : [\mathbf{x}^N_{1:T}]$
$\mathbf{X}^2 : [\mathbf{x}^2_{1:T}]$
$\mathbf{X}^1 : [\mathbf{x}^1_{1:T}]$
$\mathbf{X}_{in} \in \mathbb{R}^{D\times T\times N}$
dimension permutation
$\mathbf{Y} \in \mathbb{R}^{D\times N\times T}$
$\tilde{\mathbf{Y}}^T$
$\tilde{\mathbf{Y}}^2$
$\tilde{\mathbf{Y}}^1$
$\tilde{\mathbf{Y}} \in \mathbb{R}^{D\times N\times T}$
Linear
Speaker-wise Contextualizer
$\mathbf{Y}^T : [\mathbf{y}^T_{1:N}]$
$\mathbf{Y}^2 : [\mathbf{y}^2_{1:N}]$
$\mathbf{Y}^1 : [\mathbf{y}^1_{1:N}]$
dimension permutation
$\mathbf{X}_{out} \in \mathbb{R}^{D\times T\times N}$

Fig. 2. The Frame-Speaker Contextualizer. The operator ○ denotes the residual addition of two tensors. Each $\mathbf{X}^i$ ($\mathbf{Y}^j$) passes through the Frame-wise (Speaker-wise) Contextualizer separately. $\mathbf{X}_{out}$ is used as $\mathbf{X}_{in}$ for the subsequent Frame-Speaker Contextualizer.

B. Detector: Feature Mixer / Frame-Speaker Contextualizer

The Detector consists of a Feature Mixer and three consecutive Frame-Speaker Contextualizers. Each speaker representation is concatenated with the convolutionized MFCCs, and subsequently processed by the Feature Mixer to generate the corresponding Speaker-aware frames (SAFs), $\mathbf{X}^i$, where i is the speaker index, as indicated in Fig. 1. As illustrated in Fig. 2, the stack of the Speaker-aware frames of all speakers, $\mathbf{X}_{in} \in \mathbb{R}^{D\times T\times N}$, is the input of the Frame-Speaker Contextualizer, where D, T, and N denote the dimension of the SAF features, number of frames, and number of speakers, respectively. Inspired by DPRNN, the Frame-Speaker Contextualizer is designed to handle different numbers of speakers in one session. It contains two stages. In the first stage, the SAF of each speaker goes through the Frame-wise Contextualizer separately to generate the temporal contextual information of each speaker using

$$\tilde{\mathbf{X}}^i = Linear(Contextualizer_F(\mathbf{X}^i)) + \mathbf{X}^i. \tag{1}$$

The output of the first stage, $\tilde{\mathbf{X}}$, is the stack of $\tilde{\mathbf{X}}^i$, $i = 1, ..., N$. The input of the second stage, $\mathbf{Y}$, is generated by applying a dimension permutation to $\tilde{\mathbf{X}}$. Slicing the input by time frames yields $\mathbf{Y}^j \in \mathbb{R}^{D\times N}$, $j = 1, ..., T$, which is treated as the activity of the individual speakers in a single frame j. Similar to the frame-wise processing of the SAF, $\mathbf{Y}^j$ goes through the Speaker-wise Contextualizer to generate the speaker contextual information for each frame using

$$\tilde{\mathbf{Y}}^j = Linear(Contextualizer_S(\mathbf{Y}^j)) + \mathbf{Y}^j. \tag{2}$$

$\tilde{\mathbf{Y}}^j$, $j = 1, ..., T$, is stacked and permuted to $\mathbf{X}_{out} \in \mathbb{R}^{D\times T\times N}$. $\mathbf{X}_{out}$ is used as the input, $\mathbf{X}_{in}$, of the subsequent Frame-Speaker Contextualizer. In Eq. (2), the number of speakers N is the length of the input sequence; therefore, it is variable. Finally, for each speaker, the corresponding SAF from the output of the previous Frame-Speaker Contextualizer is passed through a linear-sigmoid layer to generate the final diarization result. MTFAD enables information sharing among all speakers and frames with the Frame- and Speaker-wise contextualizers, which not only retains the advantages of TS-

TABLE I

DER (%) of MTFAD with x-vector on the SWB+SRE dataset. During testing, the x-vectors were derived from the speaker timestamps labeled by the first-stage diarization of x-vector/AHC. **Threshold** denotes the case that number of clusters in AHC is determined automatically, and **Oracle** denotes the case that number of clusters in AHC is based on true number of speakers. **Ideal** denotes the case that the target speakers in the test session are all known, and their x-vectors are derived from corresponding raw utterances.

Ratio ($\|T\|/(\|T\|+\|I\|)$)	**Threshold**	**Oracle**	**Ideal**
0%	29.92	26.37	**16.65**
25%	29.82	25.91	**12.03**
50%	30.05	26.26	**11.14**
75%	29.80	26.22	**10.36**
100%	30.43	27.74	**10.22**

VAD, but also does not exhibit the limitation of handling only conversations of a fixed number of speakers.

C. Filter: z-vector (diari“z”ation vector)

We argue that the quality of speaker representations is critical for the diarization performance. Therefore, we design a filter specifically for extracting speaker representations that are suitable for diarization. Using the speaker timestamps provided by the x-vector/AHC diarization step and the MFCCs of the session, the filter generates the corresponding speaker representations (i.e., z-vectors), as indicated in Fig. 1. The filter comprises ResNet and Attentive Statistic Pooling [34]. These z-vectors can be used as inputs to the Detector instead of the x-vectors or i-vectors. In MTFAD, the speaker representation extraction and detection are combined into an end-to-end process by integrating the Filter with the Detector. Furthermore, because the Filter and Detector are trained jointly, the z-vector is expected to be more suitable than the x- and i-vectors for solving the diarization problem.

III. Experiments and Results

Two corpora were used in our experiments: one was simulated from the Switchboard and NIST SRE datasets, and the other was CALLHOME. All overlapping regions were counted during the performance evaluation. The results were evaluated by diarization error rate (DER) and Jaccard error rate (JER), with the standard 250 ms collar. The JER is based on the Jaccard index [35]. In all experiments, we calculated the loss between the projected result and the answer using cross-entropy in the training phase. For the model settings, both the Feature Mixer and Frame-wise Contextualizer were implemented using BiLSTM. The Speaker-wise Contextualizer can be implemented using Transformer [36] or BiLSTM. As the number of speakers is limited in both datasets (i.e. less than 10), a lightweight BiLSTM is sufficient to gather all information regarding the speakers. Therefore, in this study, the Speaker-wise Contextualizer was implemented using BiLSTM.

A. SWB+SRE Simulated Corpus

We utilized the additional Switchboard corpus and the NIST-SRE dataset. The total number of speakers in the 683 hours of data from SRE and Switchboard was 6,392. We simulated the training data and evaluation data by Algorithm 1 in [28]. The speakers differed in the two sets. To achieve an overlap ratio of 20%, with two, three, and four participants, we selected the parameters β for 3, 6, and 9 seconds in the algorithm, resulting in 137, 226, and 320 hours of data, respectively.

TABLE II

Results (%) on the SWB+SRE dataset. All MTFAD models are based on the first-stage diarization of x-vector/AHC.

Method	**Threshold**		**Oracle**	
	DER	JER	DER	JER
x-vector/AHC	38.49	53.38	40.38	52.58
MTFAD (i-vector)	23.72	36.21	19.50	29.24
MTFAD (x-vector)	25.16	38.91	18.61	28.88
MTFAD (z-vector)	**23.06**	**32.67**	**13.46**	**18.95**

In the training phase, the ground-truth, Rich Transcription Time Marked (RTTM) format, was used as the first-stage diarization. In the inference phase, we used the x-vector/AHC to produce the first-stage RTTM files for the evaluation data. The *Threshold* parameters in the x-vector/AHC were set based on the performance of the training set. The generation of speaker representations followed the approach described in Section II-A. In both phases, the i-vector and x-vector speaker representations were obtained using the pretrained extractors of Kaldi [37]. For the z-vector, first-stage RTTM was used as the input to the MTFAD model.

Results and discussion. First, we investigated the impact of the quality of the speaker representations on the diarization performance. The Filter in MTFAD was removed and the z-vectors were replaced with the x-vectors (cf. Fig. 1). During the MTFAD training, the x-vectors were extracted from the utterances of the target speakers in the unmixed Switchboard and NIST-SRE datasets (denoted as T) or derived from the speaker timestamps labeled by the first-stage diarization of x-vector/AHC (denoted as I). The ratio of $|T|/(|T|+|I|)$ represents the extent to which true speaker representations are used for training. As indicated in Table I, this ratio hardly affected the performance. However, regardless of the ratio, all *Ideal* test conditions outperformed their *Threshold* and *Oracle* counterparts. Although *Ideal* is a cheating condition, these results demonstrate the importance of accurate speaker representation. Therefore, we conclude that extracting better speaker representations is key to producing superior results.

Thereafter, we compared the effects of different speaker representation models, including the z-vector, i-vector, and x-vector models. As indicated in Table II, whereas the MTFAD model with the i-vector or x-vector exhibited improved performance over the baseline x-vector/AHC method, the MTFAD model with the z-vector achieved the best performance. The results confirm that the z-vector jointly trained with the MTFAD model is more effective than the x-vector and i-vector that are obtained from pretrained models, and better speaker representation yields superior diarization performance.

B. CALLHOME (LDC2001S97)

CALLHOME is a telephone dataset containing conversations in multiple languages. The dataset includes a total of 500 conversations are recorded at a sampling rate of 8 kHz. The number of speakers in each conversation varies from two to

TABLE III

Results (%) on CALLHOME (assuming the number of speakers is known). Both TS-VAD and MTFAD were based on the first-stage diarization of the x-vector/AHC and i-vector. For each specific number of speakers, MTFAD* and TS-VAD were trained on the same training data. MTFAD was trained with all training data containing different numbers of speakers.

Method	**Oracle #2**		**Oracle #3**		**Oracle #4**	
	DER	JER	DER	JER	DER	JER
SA-EEND -EDA [29]	8.35	N/A	13.20	N/A	21.71	N/A
x-vector/AHC	9.17	24.94	15.24	37.04	20.28	45.35
TS-VAD	9.51	20.60	14.71	33.62	20.18	44.77
MTFAD*	8.72	17.90	14.50	33.60	18.15	43.24
MTFAD	**7.82**	**17.87**	**13.10**	**32.43**	**18.12**	**39.02**

TABLE IV

Results (%) on CALLHOME. All MTFAD models were based on the first-stage diarization of the x-vector/AHC.

Method	**Threshold (Estimated)**		**Oracle**	
	DER	JER	DER	JER
x-vector/AHC [29]	19.43	N/A	18.98	N/A
SA-EEND-EDA [29]	15.29	N/A	15.43	N/A
MTFAD (i-vector)	14.52	30.09	14.10	27.92
MTFAD (x-vector)	14.55	30.01	13.15	26.80
MTFAD (z-vector)	**14.31**	**29.21**	**12.66**	**24.56**

seven. As the CALLHOME dataset was too small to train our model, we used the SWB+SRE dataset for pretraining.

In the training phase, we pretrained the MTFAD models on the SWB+SRE dataset. We followed the instructions of Kaldi to divide the set equally into two parts. CALLHOME-1 was used to fine-tune the pretrained models, whereas CALLHOME-2 was used for evaluation. We determined the Threshold parameters in x-vector/AHC based on the performance of CALLHOME-1. In the inference phase, we used x-vector/AHC to produce the first-stage RTTM files on CALLHOME-2. We used the same approach to produce three types of speaker representations as the experiments on the SWB+SRE simulated corpus.

As the speaker numbers in a session in CALLHOME varies from two to seven, it was necessary to pretrain and fine-tune the TS-VAD models separately for each number of speakers. For this purpose, the SWB+SRE and CALLHOME-1 datasets were split into 2-, 3-, and 4-speaker subsets for training the corresponding TS-VAD models. For each TS-VAD model, we also trained the corresponding MTFAD* model using the same training data and procedure for comparison.

Results and discussion. First, we compared MTFAD with TS-VAD. The results of the evaluation using CALLHOME-2 are presented in Table III. The experiments were conducted under the 2-, 3-, and 4-speaker conditions, and the number of speakers was assumed to be known. All MTFAD and TS-VAD systems were based on the first-stage diarization of the x-vector/AHC. It can be observed From the table that MTFAD* always outperformed the corresponding TS-VAD and x-vector/AHC baselines under the same conditions. Moreover, MTFAD achieved better results than MTFAD* because it was trained with all training data containing different numbers of speakers, whereas each MTFAD* model was trained with only a subset of training data containing a specific number of speakers. The results demonstrate the advantage of the MTFAD Detector: MTFAD significantly performed TS-VAD because its detector could use all data during training. After overcoming the weakness of TS-VAD, MTFAD with the i-vector has already outperformed SA-EEND-EDA [29].

TABLE V

DER (%) on the 2-speaker CALLHOME task. **rel. %** represents the relative reduction over the baseline x-vector/AHC.

Method	**DER**	**rel. %**
x-vector/AHC [29]	8.93	-
BLSTM-EEND [24]	23.07	-158.3
SA-EEND [28]	10.99	-23.1
SA-EEND-EDA [29]	8.35	6.5
SA-EEND-EDA + Frame Selection [30]	7.84	12.2
MTFAD	**7.82**	**12.4**

Subsequently, we compared the effects of different speaker representation models, including the z-vector, i-vector, and x-vector models. It can be observed from Table IV that all the three MTFAD models outperformed not only the x-vector/AHC baseline, but also the strong SA-EEND-EDA model [29]. Furthermore, the results confirm that the z-vector-based MTFAD was superior to the i-vector-based and x-vector-based MTFAD under both the *Threshold* and *Oracle* conditions. Furthermore, greater improvements were observed under the *Oracle* condition. This is because the model could estimate a more accurate z-vector for each speaker when the number of speakers was correct in the first-stage diarization. In contrast, under the *Threshold* condition, the incorrect number of speakers being predicted in the first-stage x-vector/AHC could cause certain z-vectors to not match the actual speakers, thereby leading to fewer reductions in the DER and JER. The results in Tables III and IV reveal that, with its well-designed Filter and Detector, MTFAD is a flexible and effective diarization model that can extract more accurate speaker vectors to handle conversations with different numbers of speakers.

Finally, we compared MTFAD with other models. As most end-to-end models have only been evaluated in 2-speaker experiments, we compared different models in the 2-speaker CALLHOME task. It is clear from Table V that MTFAD outperformed all models, with a 12.2% relative reduction in the DER over the x-vector/AHC [29]. According to Tables IV and V, MTFAD outperformed all of the models compared in this study for both 2-speaker and multi-speaker tasks.

IV. Conclusion

We have proposed the MTFAD model, which is composed of a Frame-Speaker Contextualizer based detector and z-vector filter, for speaker diarization. The structure of its detector allows MTFAD to handle conversations with varying numbers of speakers and to use data with any number of speakers during training. The detector addresses the weaknesses of TS-VAD while preserving its strengths. The z-vector filter that is dedicated to diarization also improves the performance compared to the traditional i-vector and x-vector approaches.

References

[1] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, and G. Friedland, "Speaker diarization: A review of recent research," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 2, pp. 356–370, 2012.

[2] N. Ryant, P. Singh, V. Krishnamohan, R. Varma, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, "The third DIHARD diarization challenge," 2020. [Online]. Available: http://arxiv.org/abs/2012.01477

[3] M. Diez, L. Burget, and P. Matejka, "Speaker diarization based on bayesian HMM with eigenvoice priors," in *Proc. Odyssey*, 2018.

[4] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," in *Proc. Interspeech*, 2019.

[5] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. ICASSP*, 2018.

[6] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *Proc. ICASSP*, 2017.

[7] M. Diez, L. Burget, S. Wang, J. Rohdin, and H. Černocký, "Bayesian HMM based x-vector clustering for speaker diarization," in *Proc. Interspeech*, 2019.

[8] G. Sell, D. Snyder, A. Mccree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, and S. Khudanpur, "Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge," in *Proc. Interspeech*, 2018.

[9] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, pp. 788–798, 2011.

[10] G. Sell and D. Garcia-Romero, "Speaker diarization with PLDA i-vector scoring and unsupervised calibration," in *Proc. IEEE SLT*, 2014.

[11] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *Proc. ICASSP*, 2018.

[12] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully supervised speaker diarization," in *Proc. ICASSP*, 2018.

[13] S. Ioffe, "Probabilistic linear discriminant analysis," in *Proc. ECCV*, 2006.

[14] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. ICCV*, 2007.

[15] P. Kenny, T. Stafylakis, P. Ouellet, M. Jahangir Alam, and P. Dumouchel, "PLDA for speaker verification with utterances of arbitary duration," in *Proc. ICASSP*, 2013.

[16] H. Ning, M. Liu, H. Tang, and T. Huang, "A spectral clustering approach to speaker diarization," in *Proc. Interspeech*, 2006.

[17] T. J. Park, K. J. Han, J. Huang, X. He, B. Zhou, P. Georgiou, and S. Narayanan, "Speaker diarization with lexical information," in *Proc. Interspeech*, 2019.

[18] C. Fredouille, S. Bozonnet, and N. Evans, "The LIA-EURECOM RT'09 speaker diarization system," in *Proc. RT*, 2009.

[19] S. H. Shum, N. Dehak, R. Dehak, and J. R. Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, no. 10, pp. 2015–2028, 2013.

[20] G. Sell and D. Garcia-Romero, "Diarization resegmentation in the factor analysis subspace," in *Proc. ICASSP*, 2015.

[21] M. Sahidullah, J. Patino, S. Cornell, R. Yin, S. Sivasankaran, H. Bredin, P. Korshunov, A. Brutti, R. Serizel, E. Vincent, N. Evans, S. Marcel, S. Squartini, and C. Barras, "The speed submission to DIHARD II: Contributions & lessons learned," 2019. [Online]. Available: http://arxiv.org/abs/1911.02388

[22] Y. Fujita, S. Watanabe, S. Horiguchi, Y. Xue, and K. Nagamatsu, "End-to-End neural diarization: Reformulating speaker diarization as simple multi-label classification," 2020. [Online]. Available: http://arxiv.org/abs/2003.02966

[23] T. von Neumann, K. Kinoshita, M. Delcroix, S. Araki, T. Nakatani, and R. Haeb-Umbach, "All-neural online source separation, counting, and diarization for meeting analysis," in *Proc. ICASSP*, 2018.

[24] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," in *Proc. Interspeech*, 2019.

[25] L. Bullock, H. Bredin, and L. P. Garcia-Perera, "Overlap-aware diarization: Resegmentation using neural end-to-end overlapped speech detection," in *Proc. ICASSP*, 2020.

[26] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. ICASSP*, 2017.

[27] N. Zeghidour, O. Teboul, and D. Grangier, "DIVE: End-to-end speech diarization via iterative speaker embedding," 2021. [Online]. Available: http://arxiv.org/abs/2105.13802

[28] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," in *Proc. IEEE ASRU*, 2019.

[29] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, "End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors," in *Proc. Interspeech*, 2020.

[30] S. Horiguchi, P. Garcia, Y. Fujita, S. Watanabe, and K. Nagamatsu, "End-to-end speaker diarization as post-processing," in *Proc. ICASSP*, 2021.

[31] I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya, I. Sorokin, T. Timofeeva, A. Mitrofanov, A. Andrusenko, I. Podluzhny, A. Laptev, and A. Romanenko, "Target-speaker voice activity detection: A novel approach for multi-speaker diarization in a dinner party scenario," in *Proc. Interspeech*, 2020.

[32] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation," in *Proc. ICASSP*, 2020.

[33] C. Li, Y. Luo, C. Han, J. Li, T. Yoshioka, T. Zhou, M. Delcroix, K. Kinoshita, C. Boeddeker, Y. Qian, S. Watanabe, and Z. Chen, "Dual-path RNN for long recording speech separation," in *Proc. IEEE SLT*, 2021.

[34] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," *Proc. Interspeech*, 2018.

[35] "Jaccard index." [Online]. Available: https://en.wikipedia.org/wiki/Jaccard_index

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, 2017.

[37] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. S. Silovsky, G. Stemmer, and K. V. Vesely, "The Kaldi speech recognition toolkit," in *Proc. IEEE ASRU*, 2011.