End-to-End Neural Speaker Diarization with Permutation-Free Objectives

Yusuke Fujita[1,2] , *Naoyuki Kanda*[1], *Shota Horiguchi*[1], *Kenji Nagamatsu*[1], *Shinji Watanabe*[2]

[1] Hitachi, Ltd. Research & Development Group, Japan
[2] Center for Language and Speech Processing, Johns Hopkins University, USA
{yusuke.fujita.su, naoyuki.kanda.kn, shota.horiguchi.wk, kenji.nagamatsu.dm}@hitachi.com,
shinjiw@ieee.org

Abstract

In this paper, we propose a novel end-to-end neural-network-based speaker diarization method. Unlike most existing methods, our proposed method does not have separate modules for extraction and clustering of speaker representations. Instead, our model has a single neural network that directly outputs speaker diarization results. To realize such a model, we formulate the speaker diarization problem as a multi-label classification problem, and introduces a permutation-free objective function to directly minimize diarization errors without being suffered from the speaker-label permutation problem. Besides its end-to-end simplicity, the proposed method also benefits from being able to explicitly handle overlapping speech during training and inference. Because of the benefit, our model can be easily trained/adapted with real-recorded multi-speaker conversations just by feeding the corresponding multi-speaker segment labels. We evaluated the proposed method on simulated speech mixtures. The proposed method achieved diarization error rate of 12.28%, while a conventional clustering-based system produced diarization error rate of 28.77%. Furthermore, the domain adaptation with real-recorded speech provided 25.6% relative improvement on the CALLHOME dataset. Our source code is available online at https://github.com/hitachi-speech/EEND.

Index Terms: end-to-end speaker diarization, permutation-free scheme, overlapping speech, neural network

1. Introduction

Speaker diarization is the process of partitioning speech segments according to the speaker identity. It is an important process for a wide variety of applications such as information retrieval from broadcast news, meetings, and telephone conversations [1, 2]. It also helps automatic speech recognition performance in multi-speaker conversation scenarios in meetings (ICSI [3, 4], AMI [5, 6]) and home environments (CHiME-5 [6–10]).

Typical speaker diarization systems are based on extraction and clustering of speaker representations [11–18]. The system first extracts speaker representations such as i-vectors [12, 13, 17, 19], d-vectors [18, 20], or x-vectors [16, 21]. Then, the speaker representations of short segments are partitioned into speaker clusters. Various clustering algorithms have been adopted, such as Gaussian mixture models [11, 12], agglomerative hierarchical clustering [11, 13, 16, 17], mean shift [14], k-means [15, 18], Links [18, 22], and spectral clustering [18]. These clustering-based diarization methods have shown to be effective in various datasets (see the DIHARD challenge 2018 activities, e.g., [23–25]).

However, there are two problems in the clustering-based method. Firstly, the clustering-based method implicitly assumes one speaker per segment, so it is difficult to deal with speaker-overlapping speech. Secondly, it cannot be optimized to minimize diarization errors directly because the clustering is performed in an unsupervised manner.

To deal with speaker-overlapping speech, a neural network based source separation model was recently proposed [26]. The model separates one speaker's time-frequency mask in one iteration, and separates another speaker's mask in another iteration. Utilizing the source separation technique, speaker diarization is realized even in overlapping speech. However, their source separation training objective does not necessarily minimize diarization errors. Aiming at the speaker diarization problem, it will be better to use a diarization error-oriented objective function. Moreover, there is another drawback in their method that real multi-speaker recordings cannot be used for training, because their model requires clean, non-overlapping reference speech for training.

For the optimization based on diarization errors, a fully supervised diarization method has been proposed [27]. This method formulates the speaker diarization problem based on a factored probabilistic model, which consists of modules for speaker change, speaker assignment and feature generation. However, in their method, the speaker-change model assumes one speaker for each segment, which hinders the application of the method for speaker-overlapping speech.

In this paper, we propose a novel end-to-end neural network-based speaker diarization model (EEND). In contrast to previous techniques, the EEND can both deal with overlapping speech as well as be trained directly to minimize diarization errors. Given an audio recording with utterances by multiple speakers, our recurrent neural network estimates joint speech activities of all speakers frame-by-frame. This model is categorized as multi-label classification, similar to the well-known method in sound event detection (SED) [28]. Unlike SED, the output speaker labels are ambiguous in the training stage, i.e. not corresponding to any fixed class, which is known in the source separation research field as a permutation problem. To solve the problem, we introduce a permutation-free scheme [29, 30] into the training objective function. The model is trained in an end-to-end fashion using the objective function that provides minimal diarization errors.

The EEND has various advantages over the conventional methods. Firstly, the EEND can explicitly handle overlapping speech by simply feeding overlapping speech as input during training and inference. Secondly, the EEND does not require separate modules for speech activity detection, speaker identification, source separation, or clustering. The proposed model integrates their functionality into a single neural network. Thirdly, unlike the source separation model, the EEND does not require

The first author performed the work while at Center for Language and Speech Processing, Johns Hopkins University as a Visiting Scholar.

clean, non-overlapping speech for training the model with synthetic conversational mixtures. This enables the use of domain adaptation using real overlapping speech conversations.

2. Proposed Method

2.1. Neural probabilistic model of speaker diarization

Speaker diarization is the process of partitioning speech segments according to the speaker identity. In other words, speaker diarization determines "who spoke when." We formulate the speaker diarization task as a multi-label classification problem. It can be formulated as follows:

Given an observation sequence $X = (\mathbf{x}_t \in \mathbb{R}^F \mid t = 1, \cdots, T)$ from an audio signal, estimate the speaker label sequence $Y = (\mathbf{y}_t \mid t = 1, \cdots, T)$. Here, $\mathbf{x}_t$ is a F-dimensional observation feature vector at time index t. Speaker label $\mathbf{y}_t = [y_{t,c} \in \{0, 1\} \mid c = 1, \cdots, C]$ denotes a joint activity for multiple (C) speakers at time index t. For example, $y_{t,c} = 1$ and $y_{t,c'} = 1$ represent an overlap situation of both speakers c and c' being present at time index t. Thus, determining Y is a sufficient condition to determine the speaker diarization information.

The most probable speaker label sequence $\hat{Y}$ is estimated among all possible speaker label sequences $\mathcal{Y}$, as follows:

$$\hat{Y} = \underset{Y \in \mathcal{Y}}{\arg\max}\, P(Y|X). \tag{1}$$

$P(Y|X)$ can be factorized using conditional independence assumption as follows:

$$P(Y|X) = \prod_t P(\mathbf{y}_t | \mathbf{y}_1, \cdots \mathbf{y}_{t-1}, X), \tag{2}$$

$$\approx \prod_t P(\mathbf{y}_t|X) \approx \prod_t \prod_c P(y_{t,c}|X). \tag{3}$$

Here, we assume the frame-wise posterior is conditioned on all inputs, and each speaker is present independently.

The frame-wise posterior $P(y_{t,c}|X)$ is modeled with bidirectional long short-term memory (BLSTM), as follows:

$$\mathbf{h}_t^{(1)} = \mathrm{BLSTM}_t(\mathbf{x}_1, \cdots, \mathbf{x}_T) \in \mathbb{R}^{2H}, \tag{4}$$

$$\mathbf{h}_t^{(p)} = \mathrm{BLSTM}_t(\mathbf{h}_1^{(p-1)}, \cdots, \mathbf{h}_T^{(p-1)})\ (2 \leq p \leq P), \tag{5}$$

$$\mathbf{z}_t = \sigma(\mathrm{Linear}(\mathbf{h}_t^{(P)})) \in (0, 1)^C, \tag{6}$$

where $\mathrm{BLSTM}_t(\cdot)$ is a BLSTM layer which accepts an input sequence and outputs $2H$-dimensional hidden activations $\mathbf{h}_t^{(p)}$ at time index t.[1] We use P-layer stacked BLSTMs. The frame-wise posteriors $\mathbf{z}_t$ is calculated from $\mathbf{h}_t^{(P)}$ using a fully-connected layer $\mathrm{Linear} : \mathbb{R}^{2H} \rightarrow \mathbb{R}^C$ and the element-wise sigmoid function $\sigma(\cdot)$.

The difficulty on training of the model described above is that the model have to deal with the speaker permutations: changing an order of speakers within a correct label sequence is also regarded as correct. An example of the permutations in a two-speaker case is shown in Fig. 1. In this paper, we call this the label ambiguity. This label ambiguity obstructs the training of the neural network when we just use a standard binary cross entropy loss function.

To cope with the label ambiguity problem, we introduce two permutation-free loss functions as shown in Fig. 1. The first loss function is the permutation-invariant training (PIT) loss function, which is used for considering all the permutations of ground-truth speaker labels. The second loss function is the Deep Clustering (DPCL) loss function, which is used for encouraging hidden activations of the network to be speaker-discriminative representations. Note that the use of multi-label classification model is similar to the well-known SED method [28]. However, in contrast to the SED, the multi-label classification model used for speaker diarization suffers from the label ambiguity problem. Our contribution is introducing two permutation-free loss functions to cope with the label ambiguity problem.

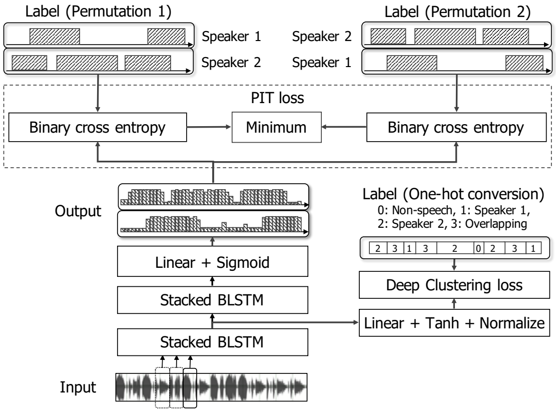

Figure 1: *Two-speaker end-to-end neural speaker diarization (EEND) model trained with the PIT loss and the DPCL loss*

2.2. Permutation-invariant training loss

The neural network is trained to minimize the error between the output $\mathbf{z}_t$ predicted in Eq. 6 and the ground-truth speaker label $\mathbf{l}_t$. Considering that the speaker label has ambiguity of their permutations, we introduce the permutation-free scheme [29, 30]. More specifically, we utilize the utterance-level permutation-invariant training (PIT) criterion [31] in the proposed method. We apply the PIT criterion on time sequence of speaker labels instead of time-frequency mask used in [31]. The PIT loss function is written as follows:

$$J^{\mathrm{PIT}} = \frac{1}{TC} \min_{\phi \in \mathrm{perm}(C)} \sum_t \mathrm{BCE}(\mathbf{l}_t^{\phi}, \mathbf{z}_t), \tag{7}$$

where $\mathrm{perm}(C)$ is a set of all the possible permutation of $(1, \ldots, C)$, and $\mathbf{l}_t^{\phi}$ is the ϕ-th permutation of the ground-truth speaker label, $\mathrm{BCE}(\cdot, \cdot)$ is the binary cross entropy function between the label and the output.

2.3. Deep Clustering loss

Assuming that the neural network extracts speaker representation in lower layers and then performs segmentation using higher layers, the middle layer activations can be regarded as the speaker representation. Therefore, we introduce a speaker representation learning criterion on the middle layer activations.

Here, the q-th layer activations $\mathbf{h}_t^{(q)}$ obtained from Eq. 5 are transformed into normalized D-dimensional embedding $\mathbf{v}_t$ as follows:

$$\mathbf{v}_t = \mathrm{Normalize}(\mathrm{Tanh}(\mathrm{Linear}(\mathbf{h}_t^{(q)}))) \in \mathbb{R}^D, \tag{8}$$

[1] It is a concatenated vector of H-dimensional forward and backward LSTM outputs.

where $\mathrm{Tanh}(\cdot)$ is the element-wise hyperbolic tangent function and $\mathrm{Normalize}(\cdot)$ is the L2 normalization function. We apply the Deep Clustering (DPCL) loss function [29] so that the embeddings are partitioned into speaker-dependent clusters as well as overlapping and non-speech clusters. For example in a two-speaker case, we generate four clusters (Non-speech, Speaker 1, Speaker 2, and Overlapping) as shown in Fig. 1.

DPCL loss function [29] is used as follows:

$$J^{\mathrm{DC}} = \|VV^{\top} - L'L'^{\top}\|_F^2, \tag{9}$$

where $V = [\mathbf{v}_1, \cdots, \mathbf{v}_T]^{\top}$, and $L' \in \mathbb{R}^{T \times 2^C}$ is a matrix for each row represents one-hot vector converted from $\mathbf{l}_t$ where those elements are in the power set of speakers. $\| \cdot \|_F$ is the Frobenius norm. The loss function encourages the two embeddings at different time indices to be close together if they are in the same cluster and far away if they are in different clusters.

Then we use multi-objective training introducing a mixing parameter α:

$$J^{\mathrm{MULTI}} = (1-\alpha)J^{\mathrm{PIT}} + \alpha J^{\mathrm{DC}}. \tag{10}$$

Thus, we derive end-to-end neural speaker diarization with the above permutation-free objective.

3. Experiments

3.1. Data

This paper mainly conducted our experiments with simulated speech mixtures to verify the effectiveness of the proposed method for controlled overlap situations. Each mixture is simulated by Algorithm 1. Unlike the existing mixture simulation for source separation studies [29], we consider a diarization-style mixture simulation: each speech mixture should have dozens of utterances per speaker with reasonable silence intervals between utterances. The silence intervals are controlled by the average interval β. Larger β values generate speech with less overlap. We show performance for differing overlap ratio controlled by β in the result section Sec.3.5.

The set of utterances used for the simulation is comprised of Switchboard-2 (Phase I, II, III), Switchboard Cellular (Part 1, Part2), and NIST Speaker Recognition Evaluation datasets (2004, 2005, 2006, 2008). All recordings are telephone speech sampled at 8 kHz. Total number of speakers in these corpora is 6,381. We split them into 5,743 speakers for the training set and 638 speakers for the test set. Since there is no time annotations in these corpora, we extract utterances using speech activity detection (SAD) based on time-delay neural networks and statistics pooling[2]. This data preparation and SAD is performed using Kaldi speech recognition toolkit [32].

The set of background noises is from MUSAN corpus [33]. We used 37 recordings which are annotated as "background" noises. The set of room impulse responses (RIRs) is the Simulated Room Impulse Response Database used in [34]. The total number of RIRs is 10,000. The SNR values are sampled from 10, 15, and 20 dBs. We generated two-speaker mixtures for each speaker have 20-40 utterances ($N_{\text{spk}} = 2, N_{\text{umin}} = 20, N_{\text{umax}} = 40$). We used differing number of mixtures for the training set, and 500 mixtures for the test set.

3.2. Experimental setup

We extracted 23-dimensional log-Mel-filterbank features with 25 ms frame length and 10 ms frame shift. Each features are concatenated with those from the previous 7 frames and subsequent 7 frames. To deal with a long audio sequence in our BLSTM, we subsampled the concatenated features by a factor of 10.

Algorithm 1: Mixture simulation.

```
Input: $\mathcal{S}, \mathcal{N}, \mathcal{I}, \mathcal{R}$    // Set of speakers, noises, RIRs and SNRs
       $\mathcal{U} = \{U_s\}_{s \in \mathcal{S}}$    // Set of utterance lists
       $N_{\text{spk}}$    // #speakers per mixture
       $N_{\text{umax}}, N_{\text{umin}}$    // Max. and min. #utterances per speaker
       $\beta$    // average interval
Output: $\mathbf{y}$    // mixture
Sample a set of $N_{\text{spk}}$ speakers $\mathcal{S}'$ from $\mathcal{S}$
$\mathcal{X} \leftarrow \emptyset$    // Set of $N_{\text{spk}}$ speakers' signals
forall $s \in \mathcal{S}'$ do
    $\mathbf{x}_s \leftarrow \emptyset$    // Concatenated signal
    Sample $\mathbf{i}$ from $\mathcal{I}$    // RIR
    Sample $N_u$ from $\{N_{\text{umin}}, \ldots, N_{\text{umax}}\}$
    for $u = 1$ to $N_u$ do
        Sample $d \sim \frac{1}{\beta} \exp\left(-\frac{d}{\beta}\right)$    // Interval
        $\mathbf{x}_s \leftarrow \mathbf{x}_s \oplus \mathbf{0}^{(d)} \oplus U_s[u] * \mathbf{i}$
    $\mathcal{X}$.add $(\mathbf{x}_s)$
$L_{\max} = \max_{\mathbf{x} \in \mathcal{X}} |\mathbf{x}|$
$\mathbf{y} \leftarrow \sum_{\mathbf{x} \in \mathcal{X}} \left(\mathbf{x} \oplus \mathbf{0}^{(L_{\max} - |\mathbf{x}|)}\right)$
Sample $\mathbf{n}$ from $\mathcal{N}$    // Background noise
Sample $r$ from $\mathcal{R}$    // SNR
Determine a mixing scale $p$ from $r, \mathbf{y}$, and $\mathbf{n}$
$\mathbf{n}' \leftarrow$ repeat $\mathbf{n}$ until reach the length of $\mathbf{y}$
$\mathbf{y} \leftarrow \mathbf{y} + p \cdot \mathbf{n}'$
```

For our neural network, we used 5-layer BLSTM with 256 hidden units in each layer. For the DPCL loss, we used the second layer of BLSTM outputs to form 256-dimensional embedding. We used the Adam [35] optimizer with initial learning rate of 10^{-3}. The batch size was 10. The number of training epoch was 20. Our implementation was based on Chainer [36].

Because the output of the neural network is a probability of speech activity for each speaker, a threshold is required to obtain the decision of speech activity for each frame. We set the threshold to 0.5 for the evaluation on simulated speech mixtures. Furthermore, we apply 11-frame median filtering to prevent the production of unreasonably short segments.

3.3. Performance metric

We evaluated the proposed method with diarization error rate (DER) [37]. In many prior studies, DER had not included miss or false alarm errors due to using oracle speech/non-speech labels. Overlapping speech segments had also been excluded from the evaluation. For our DER computation we evaluated all errors, including both non-speech and overlapping speech segments, because the proposed method includes both speech activity detection and overlapping speech detection functionality. As is typical, we use a collar tolerance of 250 ms around both the start and end of each segment.

3.4. Baseline system

We compared the proposed method with the two conventional clustering-based systems [23]. The i-vector system and the x-vector system were created using the Kaldi CALLHOME di-

[2] The SAD model: `http://kaldi-asr.org/models/m4`

Table 1: *Effect of loss functions evaluated on simulated speech generated with $\beta = 2$. The models are trained using 10,000 mixtures genarated with $\beta = 2$.*

PIT loss	DPCL loss	DER (%)
-	-	41.74
✓	-	25.14
✓	✓	23.79

Table 2: *Effect of the number of training mixtures evaluated on simulated speech generated with $\beta = 2$. The models are trained with $\beta = 2$.*

Number of training mixtures	DER(%)
10,000	23.79
20,000	14.66
100,000	12.28

arization recipe[3]. To evaluate non-speech segments, we used speech segments extracted by SAD as described in Sec. 3.1.

3.5. Results

We evaluated the effect of the proposed loss functions. Without PIT loss, we used binary cross entropy loss with the fixed permutation[4]. With PIT and DPCL losses, we set the mixing parameter $\alpha = 0.5$. The results are shown in Table 1. It is observed that PIT loss is essential for training of our neural network. It also demonstrates that DPCL loss helps improve performance.

The comparison with different numbers of training mixtures is shown in Table 2. It is observed that increasing the number of training samples improves the performance. Because our proposed method can be trained with any speech mixture with corresponding time annotations, it is possible to utilize large scale speech corpora for improving robustness of the system.

We compared the proposed method with the baseline systems using the simulated speech mixtures. The results are shown in Table 3. It is observed that miss rate is dominant in the DER of the baseline systems, due to the lack of capability for overlapping speech. In contrast, the proposed method achieved significantly low miss rate. The results indicate that the proposed method successfully detects overlapping segments as well as single-speaker segments and silence segments. Regarding the confusion error rate, the proposed method is better than the i-vector system, while it is worse than the x-vector system. For reducing the confusion errors, it is possible to use data augmentation for learning noise and speaker variations, which is utilized in the x-vector system.

To investigate the robustness to variable conditions, we evaluated different overlap ratio controlled by the average interval β. Larger β values generate less overlapping speech. The results with different overlapping ratios are shown in Table 4. The baseline systems show better performance on less overlapping speech as expected. However, the proposed method unexpectedly showed degraded performance on less overlapping speech. The result suggests that the network had overfit to the specific overlap ratio: 27.3%. Investigation with various overlap ratio settings of training data is among our future work.

Table 3: *Detailed DERs (%) evaluated on simulated speech generated with the $\beta = 2$. MI, FA and CF denote miss, false alarm and confusion error rates, respectively. The proposed model is trained using 100,000 mixtures generated with $\beta = 2$.*

Method	DER	MI	FA	CF
i-vector	33.74	25.82	**1.05**	6.88
x-vector	28.77	25.82	**1.05**	**1.90**
EEND (proposed)	**12.28**	**4.47**	5.20	2.61

Table 4: *DERs (%) on different overlapping conditions. For the evaluation on simulated mixtures, the proposed model is trained using 100,000 mixtures generated with $\beta = 2$. For the evaluation on the CALLHOME dataset, the proposed model is trained with 26,712 telephone recordings. The DER without the domain adaptation is shown in the parenthesis.*

Evaluation set	Simulated mixtures			CALLHOME
β	2	3	5	-
overlap ratio (%)	27.3	19.1	11.1	11.8
i-vector	33.74	30.43	25.96	12.10
x-vector	28.77	24.46	19.78	**11.53**
EEND	**12.28**	**14.36**	**19.69**	23.07 (31.01)

In addition, we evaluated the proposed method on real telephone conversations using the CALLHOME dataset. We split two-speaker recordings from the CALLHOME dataset into two subsets: an adaptation set of 155 recordings and a test set of 148 recordings. Our neural network was trained with a set of 26,172 two-speaker recordings from telephone speech recordings as described in Sec.3.1. The overlap ratio of the training data was 5.8%. Then, it was retrained with the adaptation set. For this retraining, we used the Adam optimizer with initial learning rate of 10^{-6} and ran 5 epochs. For the postprocessing, we adjusted the threshold to 0.6 so that the DER of the adaptation set has the minimum value. Table 4 shows the DERs evaluated on the CALLHOME test set. Unfortunately, the proposed method produces worse DER than the baseline systems. This is likely because our training set has very different overlap ratio (5.8%) from the CALLHOME test set (11.8%). To reduce this condition mismatch, we tried domain adaptation. The result showed a significant DER reduction. The relative improvement introduced by domain adaptation was 25.6%. Although the DER of the proposed method was still behind those of the baseline systems, we expect it will be much improved by developing better simulation techniques of training data or just by feeding more real data, as was suggested by the result with simulated data. We will address these directions in our future work.

4. Conclusion

We proposed an end-to-end neural speaker diarization method that is directly optimized with a diarization-error-oriented objective. The experimental results show that the proposed method outperforms conventional clustering-based methods evaluated on simulated speech mixtures. Furthermore, domain adaptation with real speech data achieved a significant DER reduction on the CALLHOME dataset.

5. Acknowledgements

We would like to thank Matthew Maciejewski and Xuankai Chang for their comments that greatly improved the manuscript.

[3] `https://github.com/kaldi-asr/kaldi/tree/master/egs/callhome_diarization`

[4] We sorted the speaker names in a lexical order to obtain the fixed permutation.

6. References

[1] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Trans. on ASLP*, vol. 14, no. 5, pp. 1557–1565, 2006.

[2] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Trans. on ASLP*, vol. 20, no. 2, pp. 356–370, 2012.

[3] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The ICSI meeting corpus," in *Proc. ICASSP*, vol. I, 2003, pp. 364–367.

[4] Özgür Çetin and E. Shriberg, "Overlap in meetings: ASR effects and analysis by dialog factors, speakers, and collection site," in *Proc. MLMI*, 2006.

[5] S. Renals, T. Hain, and H. Bourlard, "Interpretation of multiparty meetings the AMI and Amida projects," in *2008 Hands-Free Speech Communication and Microphone Arrays*, 2008, pp. 115–118.

[6] N. Kanda, Y. Fujita, S. Horiguchi, R. Ikeshita, K. Nagamatsu, and S. Watanabe, "Acoustic modeling for distant multi-talker speech recognition with single- and multi-channel branches," in *Proc. ICASSP*, 2019.

[7] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. Interspeech*, 2018, pp. 1561–1565.

[8] J. Du, T. Gao, L. Sun, F. Ma, Y. Fang, D.-Y. Liu, Q. Zhang, X. Zhang, H.-K. Wang, J. Pan, J.-Q. Gao, C.-H. Lee, and J.-D. Chen, "The USTC-iFlytek Systems for CHiME-5 Challenge," in *Proc. CHiME-5*, 2018.

[9] C. Boeddeker, J. Heitkaemper, J. Schmalenstroeer, L. Drude, J. Heymann, and R. Haeb-Umbach, "Front-End Processing for the CHiME-5 Dinner Party Scenario," in *Proc. CHiME-5*, 2018.

[10] N. Kanda *et al.*, "Hitachi/JHU CHiME-5 system: Advances in speech recognition for everyday home environments using multiple microphone arrays," in *Proc. CHiME-5*, 2018.

[11] S. Meignier, "LIUM_SPKDIARIZATION: An open source toolkit for diarization," in *CMU SPUD Workshop*, 2010.

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

[13] G. Sell and D. Garcia-Romero, "Speaker diarization with PLDA i-vector scoring and unsupervised calibration," in *Proc. SLT*, 2014, pp. 413–417.

[14] M. Senoussaoui, P. Kenny, T. Stafylakis, and P. Dumouchel, "A study of the cosine distance-based mean shift for telephone speech diarization," *IEEE/ACM Trans. on ASLP*, vol. 22, no. 1, pp. 217–227, 2014.

[15] D. Dimitriadis and P. Fousek, "Developing on-line speaker diarization system," in *Proc. Interspeech*, 2017, pp. 2739–2743.

[16] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *Proc. ICASSP*, 2017, pp. 4930–4934.

[17] M. Maciejewski, D. Snyder, V. Manohar, N. Dehak, and S. Khudanpur, "Characterizing performance of speaker diarization systems on far-field speech using standard methods," in *Proc. ICASSP*, 2018, pp. 5244–5248.

[18] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with LSTM," in *Proc. ICASSP*, 2018, pp. 5239–5243.

[19] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. on ASLP*, vol. 19, no. 4, pp. 788–798, 2011.

[20] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *Proc. ICASSP*, 2018, pp. 4879–4883.

[21] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. ICASSP*, 2018, pp. 5329–5333.

[22] P. A. Mansfield, Q. Wang, C. Downey, L. Wan, and I. L. Moreno, "Links: A high-dimensional online clustering method," *arXiv preprint arXiv:1801.10123*, 2018.

[23] 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, pp. 2808–2812.

[24] M. Diez, F. Landini, L. Burget, J. Rohdin, A. Silnova, K. Zmolíková, O. Novotný, K. Veselý, O. Glembek, O. Plchot, L. Mosner, and P. Matejka, "BUT system for DIHARD speech diarization challenge 2018," in *Proc. Interspeech*, 2018, pp. 2798–2802.

[25] L. Sun, J. Du, C. Jiang, X. Zhang, S. He, B. Yin, and C.-H. Lee, "Speaker diarization with enhancing speech for the first DIHARD challenge," in *Proc. Interspeech 2018*, 2018, pp. 2793–2797.

[26] 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*, 2019.

[27] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully Supervised Speaker Diarization," in *Proc. ICASSP*, 2019.

[28] S. Adavanne and T. Virtanen, "A report on sound event detection with different binaural features," DCASE2017 Challenge, Tech. Rep., 2017.

[29] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. ICASSP*, 2016, pp. 31–35.

[30] D. Yu, M. Kolbæk, Z. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. ICASSP*, 2017, pp. 241–245.

[31] M. Kolbæk, D. Yu, Z. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Trans. on ASLP*, vol. 25, no. 10, pp. 1901–1913, 2017.

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

[33] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprints arXiv:1510.08484*, 2015.

[34] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. ICASSP*, 2017, pp. 5220–5224.

[35] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. ICLR*, 2015.

[36] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proc. Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

[37] "The 2009 (RT-09) rich transcription meeting recognition evaluation plan," http://www.itl.nist.gov/iad/mig/tests/rt/2009/docs/rt09-meeting-eval-plan-v2.pdf, 2009.