INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

End-to-End Speaker Diarization for an Unknown Number of Speakers with Encoder-Decoder Based Attractors

Shota Horiguchi[1], *Yusuke Fujita*[1], *Shinji Watanabe*[2], *Yawen Xue*[1], *Kenji Nagamatsu*[1]

[1]Hitachi, Ltd., Japan
[2]Johns Hopkins University, USA

{shota.horiguchi.wk, yusuke.fujita.su, yawen.xue.wn}@hitachi.com, shinjiw@ieee.com

Abstract

End-to-end speaker diarization for an unknown number of speakers is addressed in this paper. Recently proposed end-to-end speaker diarization outperformed conventional clustering-based speaker diarization, but it has one drawback: it is less flexible in terms of the number of speakers. This paper proposes a method for encoder-decoder based attractor calculation (EDA), which first generates a flexible number of attractors from a speech embedding sequence. Then, the generated multiple attractors are multiplied by the speech embedding sequence to produce the same number of speaker activities. The speech embedding sequence is extracted using the conventional self-attentive end-to-end neural speaker diarization (SA-EEND) network. In a two-speaker condition, our method achieved a 2.69 % diarization error rate (DER) on simulated mixtures and a 8.07 % DER on the two-speaker subset of CALLHOME, while vanilla SA-EEND attained 4.56 % and 9.54 %, respectively. In unknown numbers of speakers conditions, our method attained a 15.29 % DER on CALLHOME, while the x-vector-based clustering method achieved a 19.43 % DER.

Index Terms: speaker diarization, encoder-decoder, attractor calculation

1. Introduction

Speaker diarization is the task to estimate "who spoke when" from an audio recording. It is a key technology for various applications using automatic speech recognition (ASR) in multi-talker scenarios such as telephone conversations [1], meetings [2], conferences and lectures [3], TV shows [4], and movies [5]. Accurate diarization has been proven to improve ASR performance by constraining a speech mask when constructing a beamformer for speech separation [6, 7].

One major approach for speaker diarization is the clustering-based method [8, 9], which applies the following processes to an input audio one by one: speech activity detection, speech segmentation, feature extraction, and clustering. Progress on better speaker embeddings, such as x-vectors [10, 11] and d-vectors [12, 13], have enabled accurate clustering-based diarization. However, most clustering-based approaches (except for a few studies, *e.g.*, [14]) cannot deal with speaker overlap because each time slot is assigned to one speaker.

End-to-end speaker diarization called EEND [15, 16] has been proposed to overcome this situation. The EEND is optimized to calculate diarization results for every speaker in a mixture from input audio features using permutation invariant training (PIT) [17]. The EEND, especially self-attentive EEND (SA-EEND), showed the effectiveness of end-to-end training of the diarization model by outperforming conventional clustering-based methods. One drawback it has is that the maximum number of speakers is pre-determined by the network architecture, and it cannot deal with a case where the number of speakers is higher. On this point, EEND is less flexible than clustering-based methods, where the number of speakers can be easily changed by setting the number of clusters during inferences.

This paper proposes an encoder-decoder based attractor calculation method called EDA. It determines a flexible number of—and theoretically an infinite number of attractors—from a speech embedding sequence. We applied it to SA-EEND to enable diarization with a flexible number of speakers. Then, the diarization results are calculated using dot products between all pairs of attractors and embeddings. Evaluation results on both simulated mixtures and real recordings showed that our method achieved better results with both fixed and unknown numbers of speakers than the x-vector-based clustering method and conventional SA-EEND.

2. Related work

Several methods in the context of speech separation can process speech mixtures of a flexible number of speakers. One series of methods involve applying the one-vs-rest approach iteratively [18, 19, 20, 21]. However, it has a major drawback in that the calculation is conducted until all the speakers are extracted, so the computational time increases linearly as the number of speakers increases. Another series involve attractor-based approaches including Deep Attractor Network (DANet) [22]. It does not limit the number of speakers in the inference phase; however, the number of speakers has to be known a priori. Anchored DANet [23] successfully solved the aforementioned problems, but it always requires calculating dot products between all the possible selections of anchors and extracted embeddings even in the inference phase. Thus, it is not scalable in terms of the number of speakers.

Several efforts have been made to calculate representatives from an embedding sequence in an end-to-end manner. Lee *et al*. proposed Set Transformer to implement set-to-set transformation [24], but the number of outputs has to be defined beforehand. Meier *et al*. implemented end-to-end clustering by estimating the distribution for every possible number of clusters $K \in \{1, \ldots, K_{\max}\}$ [25] so that the maximum number is limited by the network architecture. Li *et al*. proposed encoder-decoder based clustering for speaker diarization [26], which is the most related to EDA. However, the output is a sequence of cluster numbers of each input, so each time slot is assigned to one cluster; therefore, it cannot deal with speaker overlap. Our proposed EDA, in contrast, determines a flexible number of attractors from an embedding sequence without prior knowledge about the number of clusters.

3. End-to-end neural diarization: Review

Here we briefly introduce our end-to-end diarization framework named EEND [15, 16]. The EEND takes a T-length se-

http://dx.doi.org/10.21437/Interspeech.2020-1022

quence of log-scaled Mel-filterbank based features as an input, and processes it using bi-directional long short-term memory (BLSTM) [15] or Transformer encoders [16] to obtain an embedding $\mathbf{e}_t \in \mathbb{R}^D$ at each time slot. After that, a linear transformation $f : \mathbb{R}^D \mapsto \mathbb{R}^S$ with an element-wise sigmoid function is applied to calculate posteriors $\hat{\mathbf{y}}_t = [\hat{y}_{t,1}, \ldots, \hat{y}_{t,S}]^\mathsf{T} \in (0,1)^S$ of S speakers at time slot t. In the training phase, the EEND is optimized using the PIT scheme [17], *i.e.*, the loss is calculated between $\hat{\mathbf{y}}_t$ and the groundtruth labels $\mathbf{y}_t = [y_{t,1}, \ldots, y_{t,S}]^\mathsf{T} \in \{0,1\}^S$ as follows:

$$L_d = \frac{1}{TS} \underset{\phi \in \mathrm{perm}(1,\ldots,S)}{\arg\min} \sum_{t=1}^{T} H\left(\mathbf{y}_t^\phi, \hat{\mathbf{y}}_t\right), \tag{1}$$

where $\mathrm{perm}(1,\ldots,S)$ is the set of all the possible permutation of speakers, $\mathbf{y}_t^\phi \in \{0,1\}^S$ is the permuted labels at t, and $H(\mathbf{y}_t, \hat{\mathbf{y}}_t)$ is the binary cross entropy determined as follows:

$$H\left(\mathbf{y}_t, \hat{\mathbf{y}}_t\right) \coloneqq \sum_s -y_{t,s} \log \hat{y}_{t,s} - (1 - y_{t,s}) \log\left(1 - \hat{y}_{t,s}\right). \tag{2}$$

4. Proposed method

The EEND has a critical problem, in that the output size is limited by the network architecture; the linear transformation f restricts the number of speakers S during inference. Therefore, it cannot deal with a case where the input mixture contains a higher number of speakers than the capacity. Therefore, we utilized an attractor-based method. To make our method end-to-end trainable, we designed Encoder-Decoder based Attractor calculation (EDA) to determine attractors from an embedding sequence. The overview of our proposed method is shown in Figure 1. We used the same self-attentive network in [16] as a backbone to obtain an embedding $\mathbf{e}_t$ at each time slot. In this section, we explain how we calculate a flexible number of attractors from the embeddings and obtain diarization results using the attractors.

4.1. Encoder-decoder based attractor calculation

To calculate a flexible number of attractor points from variable lengths of embedding sequences, we utilize LSTM-based encoder-decoder [27]. A sequence of D-dimensional embeddings $(\mathbf{e}_t)_{t=1}^T$ is fed into the unidirectional LSTM encoder, obtaining the final hidden state embedding $\mathbf{h}_0 \in \mathbb{R}^D$ and the cell state $\mathbf{c}_0 \in \mathbb{R}^D$:

$$\mathbf{h}_0, \mathbf{c}_0 = \mathrm{Encoder}\left(\mathbf{e}_1, \cdots, \mathbf{e}_T\right). \tag{3}$$

Next, time-invariant D-dimensional attractors $(\mathbf{a}_s)_s$ are calculated using an unidirectional LSTM decoder with the initial states $\mathbf{h}_0$ and $\mathbf{c}_0$ as follows.

$$\mathbf{h}_s, \mathbf{c}_s, \mathbf{a}_s = \mathrm{Decoder}\left(\mathbf{h}_{s-1}, \mathbf{c}_{s-1}, \mathbf{0}\right) \tag{4}$$

We use a D-dimensional zero vector $\mathbf{0}$ as the input for the decoder at each decoding step. Theoretically infinite numbers of attractors can be calculated using the LSTM decoder. The probability of whether or not the attractor $\mathbf{a}_s$ exists to determine when to stop the attractor calculation is computed using a fully-connected layer with a sigmoid function as

$$p_s = \frac{1}{1 + \exp\left(-\left(\mathbf{w}^\mathsf{T}\mathbf{a}_s + b\right)\right)}, \tag{5}$$

where $\mathbf{w}$ and b are the trainable weights and bias of the fully-connected layer, respectively.

We note that the output attractors $(\mathbf{a}_s)_s$ depend on the order of the input embeddings $(\mathbf{e}_t)_{t=1}^T$ because we use LSTMs for the EDA. To investigate the effect of the input order, we used two types of embedding order. One was a chronological order, *i.e.*, the embeddings were sorted by time slot indexes. The other was a shuffled order. In this case, we used a shuffled order of embeddings, namely $(\mathbf{e}_{\psi(t)})_{t=1}^T$, where $(\psi(1), \ldots, \psi(T))$ is one of the permutations of $(1, \ldots, T)$, for the input to the EDA.

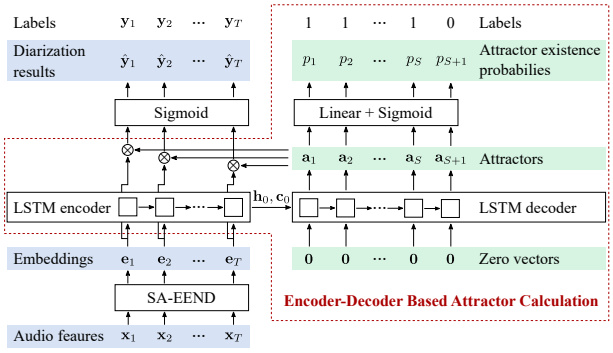

Figure 1: *SA-EEND with encoder-decoder based attractor calculation.*

In the training phase, we defined the groundtruth labels $\mathbf{l} = [l_1, \ldots, l_{S+1}]^\mathsf{T}$ using the actual number of speakers S as follows:

$$l_s = \begin{cases} 1 & (s \in \{1, \ldots, S\}) \\ 0 & (s = S+1)\,. \end{cases} \tag{6}$$

Also the attractor existence loss L_a between the labels and the estimated probabilities $\mathbf{p} = [p_1, \ldots, p_{S+1}]^\mathsf{T}$ were calculated using the binary cross entropy in Equation 2 as

$$L_a = \frac{1}{1+S} H\left(\mathbf{l}, \mathbf{p}\right). \tag{7}$$

In the inference phase, if the number of speakers S was given, we use the first S attractors, which were the output from the EDA. If the number of speakers was unknown, we first estimated it using

$$\hat{S} = \max\left\{s \mid s \in \mathbb{Z}_+ \wedge p_s \geq \tau\right\} \tag{8}$$

with a given threshold τ and then used the first $\hat{S}$ attractors.

4.2. Speaker diarization using EDA

We respectively define the matrix formulations of the embeddings extracted from the SA-EEND and the attractors from the EDA as follows.

$$E \coloneqq [\mathbf{e}_1, \ldots, \mathbf{e}_T] \in \mathbb{R}^{D \times T} \tag{9}$$

$$A \coloneqq [\mathbf{a}_1, \ldots, \mathbf{a}_S] \in \mathbb{R}^{D \times S} \tag{10}$$

The posterior probabilities can be calculated using the inner product of every embedding-attractor pair as follows:

$$\hat{Y} = \sigma(A^\mathsf{T} E) \in (0,1)^{S \times T}, \tag{11}$$

where $\sigma(\cdot)$ is the element-wise sigmoid function. Note that the output size was determined using the number of attractors so that our method could output the diarization results of a flexible number of speakers. Finally, diarization loss was calculated in the same way as SA-EEND using the PIT found in Equation 1.

The total loss is defined by the diarization loss in Equation 1 and the attractor existence loss in Equation 7 as follows:

$$L = L_d + \alpha L_a, \tag{12}$$

where α is the weighting parameter. In this study, α was set to

Table 1: *Dataset to train and test our diarization models.*

(a) *Simulated datasets.*

Dataset	#Spk	#Mixtures	Overlap ratio ρ (%)
Train			
Sim1spk	1	100,000	0.0
Sim2spk	2	100,000	34.1
Sim3spk	3	100,000	34.2
Sim4spk	4	100,000	31.5
Test			
Sim1spk	1	500	0.0
Sim2spk	2	500/500/500	34.4/27.3/19.6
Sim3spk	3	500/500/500	34.7/27.4/19.1
Sim4spk	4	500	32.0

(b) *Real datasets.*

Dataset	#Spk	#Mixtures	Overlap ratio ρ (%)
Train			
CALLHOME [30]	2	155	14.0
CALLHOME [30]	3	61	19.6
CALLHOME [30]	2-7	249	17.0
DIHARD dev [32]	1-10	192	9.8
Test			
CALLHOME [30]	2	148	13.1
CALLHOME [30]	3	74	17.0
CALLHOME [30]	2-6	250	16.7
CSJ [31]	2	54	20.1
DIHARD eval [32]	1-9	194	8.9

1.0 when the simulated data were used for training and 0.01 for adaptation on real datasets.

5. Experiments

5.1. Data

For the training and evaluation, we used simulated mixtures created from Switchboard-2 (Phase I & II & III), Switchboard Cellular (Part 1 & 2), and the NIST Speaker Recognition Evaluation (2004 & 2005 & 2006 & 2008) for speech and the MUSAN corpus [28] for noise with simulated room impulse responses used in [29] following the procedure in [16]. We note that the speaker sets for the training and test datasets were not overlapped. In [16], only the 2-speaker dataset was constructed. In this study, we created 1-, 3-, and 4-speaker datasets with similar overlap ratios ρ to the 2-speaker mixtures. We also used the telephone conversation dataset CALLHOME (CH) [30], the dialogue recordings from the Corpus of Spontaneous Japanese (CSJ) [31], and the dataset used for the second DIHARD challenge [32] to evaluate the performance on real recordings. The statistics of the datasets used are summarized in Table 1.

5.2. Experimental settings

We basically followed the training protocol of the best model described in [33][1]. We used SA-EEND with four-stacked Transformer encoders as a baseline and a backbone of our method. The inputs for the SA-EEND were 345-dimensional log-scaled Mel-filterbank based features, which were also the same as those used in the original paper. For our method, we extracted a sequence of 256-dimensional embeddings after the last layer normalization [34] of the SA-EEND, and fed them into the EDA

[1]SA-EEND is available at `https://github.com/hitachi-speech/EEND`. We will release the source code of SA-EEND with EDA at the same repository.

Table 2: *DERs (%) on 2-speaker datasets.*

	Sim2spk			Real	
Method	$\rho = 34.4\,\%$	$27.3\,\%$	$19.6\,\%$	CH	CSJ
i-vector clustering	33.74	30.93	25.96	12.10	27.99
x-vector clustering	28.77	24.46	19.78	11.53	22.96
BLSTM-EEND [15]	12.28	14.36	19.69	26.03	39.33
SA-EEND [16]	4.56	4.50	3.85	9.54	20.48
SA-EEND + EDA (Chronol.)	3.07	2.74	3.04	8.24	18.89
SA-EEND + EDA (Shuffled)	**2.69**	**2.44**	**2.60**	**8.07**	**16.27**

Table 3: *DERs (%) on 3-speaker datasets.*

	Sim3spk			Real
Method	$\rho = 34.7\,\%$	$27.4\,\%$	$19.1\,\%$	CH
x-vector clustering	31.78	26.06	19.55	19.01
SA-EEND	8.69	7.64	6.92	14.00
SA-EEND + EDA (Chronol.)	13.02	11.65	10.41	15.86
SA-EEND + EDA (Shuffled)	**8.38**	**7.06**	**6.21**	**13.92**

to calculate attractors. The threshold τ in Equation 8 to determine whether or not the attractor existed was set to 0.5. As we explained in subsection 4.1, we used two types of input order for the EDA: chronological order and shuffled order. Unless otherwise noted, we used the same type of order in the training and inference phases.

In this paper, we evaluated our method under the following two conditions: a fixed number of speakers and a flexible number of speakers. For the fixed number of speakers, we first trained our model using Sim2spk with $\rho = 34.1\,\%$ or Sim3spk with $\rho = 34.2\,\%$ for 100 epochs. We used the Adam optimizer [35] with the learning rate schedule proposed in [36] with warm-up steps of 100,000. We also finetuned those models using subsets of corresponding numbers of speakers from CALLHOME data to evaluate the performance on the real recordings. For comparison, the performance on i-vectors or x-vectors using agglomerative hierarchical clustering with probabilistic linear discriminate analysis (PLDA) scoring according to Kaldi's pretrained model [37] was also evaluated. In these cases, TDNN-based speech activity detection [38] and the oracle number of speakers were used for the evaluation. For experiments on the flexible speaker condition, we finetuned the 2-speaker model trained on Sim2spk on the concatenation of Sim1spk, Sim2spk, Sim3spk, and Sim4spk for 25 epochs. We finetuned the model using CALLHOME or DIHARD dev to evaluate the performance on real datasets. The x-vector-based methods based on the oracle number of speakers and the clustering threshold determined using the training set were also evaluated.

For the evaluation metric, we used the diarization error rate (DER). The 0.25 s of collar tolerance was defined at the start and end of each segment for the evaluation on the simulated datasets and the CALLHOME dataset. For the DIHARD dataset, we also used the Jaccard error rate (JER), and we did not use collar tolerance, following the regulation of the second DIHARD challenge [32].

5.3. Results on a fixed number of speakers

First, we evaluated our method on the 2-speaker condition like the one in [15, 16]. The results are shown in Table 2. The best DERs were attained using EDA trained on shuffled embeddings. When the model was trained using embeddings in chronological order, the DERs slightly degraded. We also show the results on the 3-speaker condition in Table 3. Our method with shuffled embeddings achieved better DERs compared with the conventional x-vector clustering and vanilla SA-EEND.

Table 4: *DERs on Sim2spk ($\rho = 34.4\,\%$) using various types of sequences.*

	Use whole sequence		Subsample $1/N$					Use the last $1/N$				
Method	Chronol.	Shuffled	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$
SA-EEND + EDA (Chronol.)	3.07	30.04	3.54	7.32	14.48	21.13	27.18	3.67	4.97	5.40	6.11	7.68
SA-EEND + EDA (Shuffled)	2.69	2.69	2.70	2.68	2.79	3.09	5.08	3.36	5.92	7.46	8.59	10.65

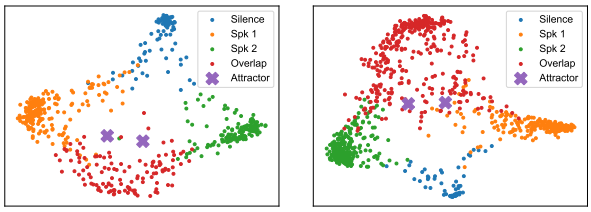

Figure 2: *Visualization of embeddings and attractors on 2-speaker mixtures in Sim2spk ($\rho = 34.4\,\%$).*

Effect of the input order: To better understand the EDA, we evaluated the diarization performance on both chronologically-ordered sequences and shuffled sequences. We also tried to reduce the length of sequences by subsampling embeddings or using the last $1/N$ of the sequences. The results on Sim2spk ($\rho = 34.4\,\%$) are shown in Table 3. When the EDA was trained on chronologically-ordered embeddings, it worked better on chronologically-ordered embeddings but degraded shuffled embeddings. If the embeddings were subsampled, the performance degradation was also severe even if the samples were ordered chronologically, while using the last $1/N$ could suppress the performance degradation. These results were that the model captured speech length tendency to output attractors. However, when the EDA was trained on shuffled embeddings, the model was not affected very much by the order and subsampling. These results show that the EDA could capture the overall sequence successfully.

Visualization: In Figure 2, we visualized embeddings and attractors of 2-speaker mixtures by applying PCA to reduce their dimensionality. The embeddings of two speakers were well separated from the silent region, and those of overlapping regions were distributed between two clusters. Attractors were successfully calculated for each of the two speakers.

5.4. Results on a flexible number of speakers

We also evaluated our method on a condition involving a flexible number of speakers. In this case, the order of the embeddings was always shuffled. The model was first finetuned from the weights trained on Sim2spk and evaluated on simulated mixtures of a flexible number of speakers. The results are shown in Table 5. Our method achieved better DERs than the x-vector clustering-based method. It achieved 4.33 % and 8.94 % DERs on two- and three-speaker mixtures, which showed only 1.64 and 0.56 point degradation from two- or three-speaker specific models, respectively. Furthermore, our method further improved performance when the actual number of speakers was given, while x-vector clustering worsened performance in most cases using the oracle number of speakers.

We also evaluated our method with real conversations using the CALLHOME. In this case, the model was finetuned again using the CALLHOME training set and evaluated on the test set. The results are shown in Table 6. Our method achieved a 15.29 % DER, which outperformed the clustering-based method. However, it did not perform well when the number of speakers was higher than four. This is because the CALLHOME contains only ten recordings that include more than four speakers.

Table 5: *DERs (%) on simulated mixtures of a flexible number of speakers.*

	Sim1spk	Sim2spk	Sim3spk	Sim4spk
Method	$\rho = 0.0\,\%$	$34.4\,\%$	$34.7\,\%$	$32.0\,\%$
x-vector clustering				
Threshold	37.42	7.74	11.46	22.45
Oracle #Spk	1.67	28.77	31.78	35.76
SA-EEND + EDA				
Estimated #Spk	0.39	4.33	8.94	13.76
Oracle #Spk	0.16	4.26	8.63	13.31

Table 6: *DERs (%) on CALLHOME of a flexible number of speakers.*

	#Spk					
Method	2	3	4	5	6	All
x-vector clustering						
Threshold	15.45	18.01	22.68	31.40	34.27	19.43
Oracle #Spk	8.93	19.01	24.48	32.14	34.95	18.98
SA-EEND + EDA						
Estimated #Spk	8.50	13.24	21.46	33.16	40.29	**15.29**
Oracle #Spk	8.35	13.20	21.71	33.00	41.07	15.43

Finally, we evaluated our method on the DIHARD dataset. The evaluation follows the DIHARD 2019 track 2, where speech activity detection has to be conducted from single channel audio. Because utilizing a high number of speakers with PIT is difficult, our system was only trained to output the most dominant seven speakers even if the input contained more than seven speakers. The results are shown in Table 7. Our SA-EEND with EDA achieved a DER of 32.59 %, which outperformed the baseline [39] and the best pre-is2019-deadline system by the DI-IT team [40], but it could not beat the best post-is2019-deadline system by the BUT team [41]. We note that our system is based on 8 kHz audio, while others use 16 kHz audio with additional training data from VoxCeleb datasets [42]. Evaluations on high-resolution audio with additional data are left for future work.

6. Conclusions

In this paper, we proposed EDA to calculate attractors from a sequence of embeddings, and we applied it to SA-EEND to implement end-to-end speaker diarization for speech mixtures of a flexible number of speakers. Our method achieved state-of-the-art DERs on conditions including both a fixed and a flexible number of speakers.

Table 7: *DERs and JERs (%) on DIHARD eval.*

Method	DER	JER
DIHARD-2 baseline [39]	40.86	66.60
Best pre-is2019-deadline [40]	35.10	57.11
Best post-is2019-deadline [41]	27.11	49.07
SA-EEND + EDA (Estimated #Speakers)	32.59	55.99

7. References

[1] P. Kenny, D. Reynolds, and F. Castaldo, "Diarization of telephone conversations using factor analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 6, pp. 1059–1070, 2010.

[2] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE TASLP*, vol. 15, no. 7, pp. 2011–2022, 2007.

[3] X. Zhu, C. Barras, L. Lamel, and J.-L. Gauvain, "Multi-stage speaker diarization for conference and lecture meetings," in *Multimodal technologies for perception of humans*. Springer, 2007, pp. 533–542.

[4] F. Vallet, S. Essid, and J. Carrive, "A multimodal approach to speaker diarization on TV talk-shows," *IEEE TMM*, vol. 15, no. 3, pp. 509–520, 2012.

[5] I. Kapsouras, A. Tefas, N. Nikolaidis, G. Peeters, L. Benaroya, and I. Pitas, "Multimodal speaker clustering in full length movies," *Multimedia Tools and Applications*, vol. 76, no. 2, pp. 2223–2242, 2017.

[6] N. Kanda, C. Boeddeker, J. Heitkaemper, Y. Fujita, S. Horiguchi, K. Nagamatsu, and R. Haeb-Umbach, "Guided source separation meets a strong ASR backend: Hitachi/Paderborn University joint investigation for dinner party scenario," in *INTERSPEECH*, 2019, pp. 1248–1252.

[7] C. Zorilă, C. Boeddeker, R. Doddipatla, and R. Haeb-Umbach, "An investigation into the effectiveness of enhancement in ASR training and test for CHiME-5 dinner party transcription," in *ASRU*, 2019, pp. 47–53.

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

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

[10] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *ICASSP*, 2019, pp. 5796–5800.

[11] M. Diez, L. Burget, S. Wang, J. Rohdin, and J. Černockỳ, "Bayesian HMM based x-vector clustering for speaker diarization," in *INTERSPEECH*, 2019, pp. 346–350.

[12] Q. Wang, W. L. Downey, Carlton, P. Andrew Mansfield, and I. Lopez Moreno, "Speaker diarization with LSTM," in *ICASSP*, 2018, pp. 5239–5243.

[13] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully supervised speaker diarization," in *ICASSP*, 2019, pp. 6301–6305.

[14] Z. Huang, S. Watanabe, Y. Fujita, P. García, Y. Shao, D. Povey, and S. Khudanpur, "Speaker diarization with region proposal network," in *ICASSP*, 2020, pp. 6514–6518.

[15] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," in *INTERSPEECH*, 2019, pp. 4300–4304.

[16] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," in *ASRU*, 2019, pp. 296–303.

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

[18] K. Kinoshita, L. Drude, M. Delcroix, and T. Nakatani, "Listening to each speaker one by one with recurrent selective hearing networks," in *ICASSP*, 2018, pp. 5064–5068.

[19] J. Shi, J. Xu, G. Liu, and B. Xu, "Listen, think, and listen again: Captureing top-down auditory attention for speaker-independent speech separation," in *IJCAI*, 2018, pp. 4353–4360.

[20] T. von Neumann, K. Kinoshita, M. Delcroix, S. Araki, T. Nakatani, and R. Haeb-Umback, "All-neural online source separation, counting, and diarization for meeting analysis," in *ICASSP*, 2019, pp. 91–95.

[21] N. Takahashi, S. Parthasaarathy, N. Goswami, and Y. Mitsufuji, "Recursive speech separation for unknown number of speakers," in *INTERSPEECH*, 2019, pp. 1348–1352.

[22] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *ICASSP*, 2017, pp. 246–250.

[23] Y. Luo, Z. Chen, and N. Mesgarani, "Speaker-independent speech separation with deep attractor network," *IEEE/ACM TASLP*, vol. 26, no. 4, pp. 787–796, 2018.

[24] J. Lee, Y. Lee, J. Kim, A. R. Kosiorek, S. Choi, and Y. W. Teh, "Set Transformer: A framework for attention-based permutation-invariant neural networks," in *ICML*, 2019, pp. 3744–3753.

[25] B. B. Meier, I. Elezi, M. Amirian, O. Dürr, and T. Stadelmann, "Learning neural models for end-to-end clustering," in *ANNPR*, 2018, pp. 126–138.

[26] Q. Li, F. L. Kreyssig, C. Zhang, and P. C. Woodland, "Discriminative neural clustering for speaker diarisation," arXiv:1910.09703, 2019.

[27] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NeurIPS*, 2014, pp. 3104–3112.

[28] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," arXiv:1510.08484, 2015.

[29] 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 *ICASSP*, 2017, pp. 5220–5224.

[30] "2000 NIST Speaker Recognition Evaluation," https://catalog.ldc.upenn.edu/LDC2001S97.

[31] K. Maekawa, "Corpus of spontaneous Japanese: Its design and evaluation," in *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[32] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, "The Second DIHARD Diarization Challenge: Dataset, task, and baselines," in *INTERSPEECH*, 2019, pp. 978–982.

[33] Y. Fujita, S. Watanabe, S. Horiguchi, Y. Xue, and K. Nagamatsu, "End-to-end neural diarization: Reformulating speaker diarization as simple multi-label classification," arXiv:2003.02966, 2020.

[34] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," in *NIPS 2016 Deep Learning Symposium*, 2016.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.

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

[38] V. Peddinti, G. Chen, V. Manohar, T. Ko, D. Povey, and S. Khudanpur, "JHU ASpIRE system: Robust LVCSR with TDNNs, iVector adaptation and RNN-LMS," in *ASRU*, 2015, pp. 539–546.

[39] 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 *INTERSPEECH*, 2018, pp. 2808–2812.

[40] S. Novoselov, A. Gusev, A. Ivanov, T. Pekhovsky, A. Shulipa, A. Avdeeva, A. Gorlanov, and A. Kozlov, "Speaker diarization with deep speaker embeddings for DIHARD Challenge II," in *INTERSPEECH*, 2019, pp. 1003–1007.

[41] F. Landini, S. Wang, M. Diez, L. Burget, P. Matějka, K. Žmolíková, L. Mošner, A. Silnova, O. Plchot, O. Novotnỳ, H. Zeinali, and S. Rohdin, "BUT system for the Second DIHARD Speech Diarization Challenge," in *ICASSP*, 2020, pp. 6529–6533.

[42] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "VoxCeleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.