



SOMSRED-SVC: Sequential Output Modeling with Speaker Vector Constraints for Joint Multi-Talker Overlapped ASR and Speaker Diarization

Naoki Makishima, Naotaka Kawata, Taiga Yamane, Mana Ihori, Tomohiro Tanaka, Satoshi Suzuki, Shota Orihashi, Ryo Masumura

NTT Corporation, Japan

naoki.makishima@ntt.com

Abstract

We have developed a sequential output model with speaker vector constraints for the joint modeling of multi-talker automatic speech recognition (ASR) and speaker diarization. The conventional approach to joint modeling of multi-talker ASR and speaker diarization, called SOMSRED, enables the estimation of speaker embeddings from fully overlapped speech by discretizing the speaker embedding space and treating the speaker embeddings as tokens. However, the predicted speaker embeddings become less distinctive compared to the ones directly obtained from non-overlapping speech due to the discretization. To address this problem, we add a new training objective that optimizes speaker embeddings in continuous space without discretization. Experimental results show that the proposed method avoids overfitting to the discretized speaker tokens and outperforms SOMSRED in both ASR performance and speaker embedding performance.

Index Terms: speech recognition, speaker diarization, overlapped speech, multi-talker speech

1. Introduction

Figuring out who spoke “when” and “what” is important for various applications such as meeting records and conversation robots. Speaker diarization and automatic speech recognition (ASR) are tasks to estimate who spoke “when” and “what”, respectively. Since multi-talker conversations such as meetings and chats often include overlapped speech of multiple people, addressing overlapped speech is important in practical terms.

ASR and speaker diarization have been developed independently. Several studies have explored multi-talker ASR to estimate “what” from overlapped speech, where the task is to transcribe each utterance of overlapped speech into text. Some studies combine speech separation and the conventional single-talker ASR [1–7], while others have tackled multi-talker ASR without utilizing speech separation [8]. On the other hand, to address overlapped speech in speaker diarization, some studies have utilized an overlap detector [9, 10], while others have utilized end-to-end speaker diarization models that estimate frame-level speaker activity [11, 12].

Several recent studies have explored the joint modeling of ASR and speaker diarization [13–18]. The goal of joint modeling is to transcribe each utterance of the overlapped speech along with their timestamps and speaker tags, enabling an optimization of the whole model. One of the most popular approaches for estimating speaker tags is to estimate speaker embeddings representing speaker characteristics of the speech and cluster them to distinguish whether the embeddings belong to the same cluster or not [15, 16, 19–23]. Although most approaches first estimate non-overlapping speech seg-

ments and then estimate speaker embeddings from these segments, a recently proposed model enables the simultaneous estimation of multi-talker ASR and speaker embeddings from fully overlapped speech without non-overlapping speech segments, which is named SOMSRED [18].

The key point of SOMSRED is to discretize the speaker embedding space and treat the speaker embeddings as tokens. The speaker embedding tokens are serialized along with transcriptions to be the target sequence of the model. This enables the model to autoregressively estimate the serialized label including both transcriptions and speaker embedding tokens in the same way as the conventional ASR model with a single output layer. Compared to the conventional method that estimates speaker embeddings from predicted non-overlapping speech segments, SOMSRED estimates more distinctive speaker embeddings utilizing both overlapping and non-overlapping speech segments, as demonstrated in [18].

However, the speaker embedding obtained with SOMSRED becomes less distinctive compared to the speaker embedding directly obtained from non-overlapping speech. We assume this is because SOMSRED is trained in discretized speaker embedding space: although this enables the model to autoregressively estimate discretized speaker tokens as well as transcriptions, poorly fitting speaker tokens are assigned when the speaker embedding of an utterance is far from any point in a discrete speaker embedding space. This makes the speaker embeddings less distinctive and the training becomes more difficult, as discussed in Section 4.4.

To address this problem, our idea is to add a new training criterion that directly optimizes speaker embeddings in continuous space without discretization. In this paper, we propose SOMSRED-SVC, which is a sequential output model trained with newly introduced speaker vector constraints loss for the joint modeling of multi-talker overlapped speech recognition and speaker diarization. The speaker vector constraints loss enables the model to predict speaker embeddings trained in continuous space. SOMSRED-SVC utilizes speaker embeddings obtained from non-overlapping speech as the speaker embedding target and trains the model so that the predicted speaker embedding becomes close to the target. Experimental results demonstrate that SOMSRED-SVC outperforms SOMSRED in both ASR performance and speaker embeddings performance.

2. Conventional Method

2.1. Sequential output model for multi-talker ASR

We denote the acoustic feature of the input speech as $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$, where $\mathbf{x}_t \in \mathbb{R}^F$ denotes the t th frame of the feature, F denotes its dimension, and T denotes the length of acoustic features. We denote utterance-level textual tokens

of multiple speakers as $\mathbf{W}^{1:K} = (\mathbf{W}^1, \dots, \mathbf{W}^K)$, where K denotes the number of speakers in the overlapped speech, $\mathbf{W}^k = (w_1^k, \dots, w_{N^k}^k)$ denotes the k th speaker’s textual tokens, N^k denotes the length of the token, $w_n^k \in \mathcal{V}$ denotes the n th textual token of the k th speaker, and \mathcal{V} denotes the vocabulary set. In serialized output training (SOT) [8], $\mathbf{W}^{1:K}$ is estimated with a single output layer recursively following the first-in, first-out order; the transcriptions of multiple speakers are estimated in the order of their utterance start time with a special symbol [sep] representing speaker change [8, 24]. The serialized token label $\mathbf{S} \in \{\mathcal{V} \cup \mathcal{O}\}$ is given as

$$\mathbf{S} = (w_1^1, \dots, w_{N^1}^1, [\text{sep}], w_1^2, \dots, w_{N^2}^2, [\text{sep}], \dots, w_{N^{K-1}}^{K-1}, [\text{sep}], w_1^K, \dots, w_{N^K}^K, [\text{eos}]), \quad (1)$$

where [eos] denotes the end of a sentence, $\mathcal{O} = \{[\text{sep}], [\text{eos}]\}$, and we assume that $\mathbf{W}^{1:K}$ is sorted in order of utterance start times for simplicity. Multi-talker ASR with SOT estimates the generation probability of \mathbf{S} given \mathbf{X} as follows:

$$P(\mathbf{S}|\mathbf{X}; \Theta_{\text{MT}}) = \prod_{l=1}^{|\mathbf{S}|} P(s_l | \mathbf{s}_{1:l-1}, \mathbf{X}; \Theta_{\text{MT}}), \quad (2)$$

where s_l denotes the l th token of \mathbf{S} , $\mathbf{s}_{1:l-1} = (s_1, \dots, s_{l-1})$, $|\mathbf{S}|$ denotes the length of \mathbf{S} , and Θ_{MT} denotes the parameter of the multi-talker ASR model.

2.2. SOMSRED

The start time token and the end time token of multiple speakers are denoted as $\mathbf{T}_s^{1:K} = ([t_s^1], \dots, [t_s^K])$ and $\mathbf{T}_e^{1:K} = ([t_e^1], \dots, [t_e^K])$, respectively, where $[t_s^k] \in \mathcal{T}$ and $[t_e^k] \in \mathcal{T}$ denote the k th speaker’s start time token and end time token, respectively, and \mathcal{T} denotes the quantized time token label set. The quantized time token is obtained by rounding the continuous timestamp values to the nearest quantized value every Q seconds. We denote the multiple speaker tokens as $\mathbf{d}^{1:K} = (\mathbf{d}^1, \dots, \mathbf{d}^K)$, where $\mathbf{d}^k = (d_1^k, \dots, d_M^k)$ denotes the k th speaker’s speaker tokens, $d_m^k \in \mathcal{D}$ denotes the m th token of the k th speaker, \mathcal{D} denotes the speaker token sets, and M denotes the number of tokens per speaker.

In SOMSRED, the simulated mixture for training is created by mixing the non-overlapping speeches. We denote the k th non-overlapping speech comprising the simulated mixture as $\mathbf{Y}^k = (\mathbf{y}_1^k, \dots, \mathbf{y}_{T'}^k)$, where $\mathbf{y}_{t'}^k \in \mathbb{R}^F$ denotes the t' th frame of the feature. SOMSRED utilizes \mathbf{Y}^k and a pre-trained speaker model that extracts the speaker embeddings to obtain speaker tokens \mathbf{d}^k as follows. First, the speaker model estimates a speaker embedding of each non-overlapping speech as $\mathbf{u}^k = \text{SpeakerModel}(\mathbf{Y}^k)$, where $\mathbf{u}^k \in \mathbb{R}^G$ denotes the speaker embedding of the k th speaker, G denotes its dimension, and $\text{SpeakerModel}(\cdot)$ denotes the speaker model. Second, \mathbf{u}^k are discretized with vector quantization. Finally, the corresponding centroid indices are assigned as speaker tokens \mathbf{d}^k . Note that multiple speaker tokens are used to represent a single speaker embedding, i.e., the average of the M discretized speaker embeddings becomes close to \mathbf{u}^k .

To efficiently model the joint generation probability of $\mathbf{W}^{1:K}$, $\mathbf{T}_s^{1:K}$, $\mathbf{T}_e^{1:K}$, and $\mathbf{d}^{1:K}$, they are serialized into a single label sequence. When the speaker tokens are estimated after the transcription, the serialized label sequence $\bar{\mathbf{S}} \in \{\mathcal{V} \cup \mathcal{O} \cup \mathcal{T} \cup \mathcal{D}\}$

is obtained as

$$\bar{\mathbf{S}} = ([t_s^1], [t_e^1], w_1^1, \dots, w_{N^1}^1, d_1^1, \dots, d_M^1, [\text{sep}], [t_s^2], [t_e^2], w_1^2, \dots, w_{N^2}^2, d_1^2, \dots, d_M^2, [\text{sep}], \dots, [t_s^K], [t_e^K], w_1^K, \dots, w_{N^K}^K, d_1^K, \dots, d_M^K, [\text{eos}]). \quad (3)$$

The joint generation probability of $\mathbf{W}^{1:K}$, $\mathbf{T}_s^{1:K}$, $\mathbf{T}_e^{1:K}$, and $\mathbf{d}^{1:K}$ is autoregressively estimated in the same way as Eq. (2).

A Transformer-based ASR model [25, 26] is used to estimate the joint generation probability, as

$$\mathbf{H} = \text{TransformerEnc}(\mathbf{X}; \theta_{\text{enc}}), \quad (4)$$

$$\mathbf{E}_l = \text{TransformerDec}(\mathbf{H}, \bar{\mathbf{s}}_{1:l-1}; \theta_{\text{dec}}), \quad (5)$$

$$P(\bar{\mathbf{s}}_l | \bar{\mathbf{s}}_{1:l-1}, \mathbf{X}; \Theta) = \text{Linear}(\mathbf{E}_l; \theta_{\text{linear}}), \quad (6)$$

where $\bar{\mathbf{s}}_l$ denotes the l th token of $\bar{\mathbf{S}}$, $\bar{\mathbf{s}}_{1:l-1} = (\bar{\mathbf{s}}_1, \dots, \bar{\mathbf{s}}_{l-1})$, $\text{TransformerEnc}(\cdot)$ is a Transformer encoder, θ_{enc} denotes its parameters, $\text{TransformerDec}(\cdot)$ is a Transformer decoder, θ_{dec} denotes its parameters, $\text{Linear}(\cdot)$ denotes a linear layer with softmax activation, and θ_{linear} denotes its parameter. The parameter $\Theta_{\text{SOMSRED}} = \{\theta_{\text{enc}}, \theta_{\text{dec}}, \theta_{\text{linear}}\}$ is optimized with the cross-entropy function defined as

$$L_{\text{CE}} = -\log P(\bar{\mathbf{S}}|\mathbf{X}; \Theta_{\text{SOMSRED}}), \quad (7)$$

where Θ_{SOMSRED} denotes the parameter of the model in SOMSRED.

During inference, the speaker embedding is utilized to assign speaker tags because the speaker tokens $\mathbf{d}^{1:K}$ fit poorly when it is applied to unknown speakers. The speaker embedding is obtained by averaging the features before the classification layer, as

$$\mathbf{e}_k = \frac{1}{|T(k)|} \sum_{v \in T(k)} \mathbf{E}_v, \quad (8)$$

where \mathbf{e}_k denotes the speaker embedding of the k th speaker, $T(k)$ denotes the decoding steps corresponding to the k th speaker token estimation, and $|T(k)|$ denotes its size.

3. Proposed Method

3.1. Strategy

As discussed in Section 1, the discretization of the speaker embedding space in SOMSRED makes the speaker embedding less distinctive and the training more difficult. This leads to a degradation of both the ASR performance and the speaker embedding performance, as discussed later in Section 4.4. To address this problem, our idea is to directly optimize speaker embeddings in continuous space without discretization.

The overview of SOMSRED-SVC is shown in Fig. 1. In addition to the cross-entropy loss (7), we introduce the speaker vector constraints loss that directly optimizes the speaker embeddings so that the estimated speaker embeddings and the target speaker embeddings become close in continuous space. While only the discretized speaker tokens after vector quantization are used as targets to estimate speaker embeddings in SOMSRED, the speaker embeddings before vector quantization is also used as targets in SOMSRED-SVC. The speaker embeddings are intermediate features corresponding to the speaker token, as in [18].

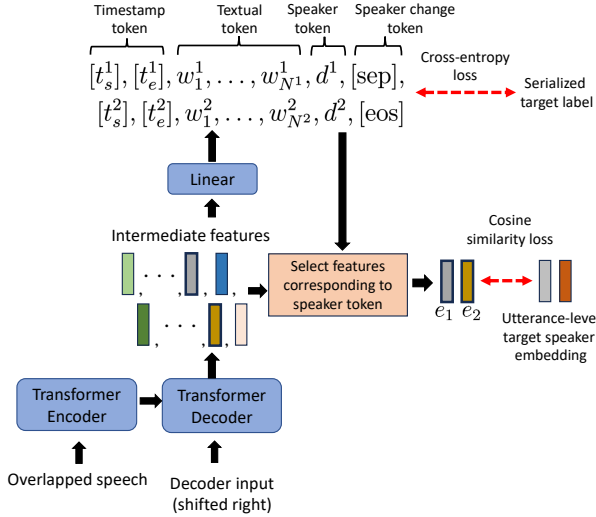


Figure 1: Overview of SOMSRED-SVC.

3.2. Formulation

SOMSRED-SVC utilizes the speaker embeddings before vector quantization \mathbf{u}^k in Section 2.2. Recall that SOMSRED utilizes multiple speaker tokens to model the residual components of the discretized speaker embedding so that the speaker embedding becomes distinctive in discretized embedding space. In contrast, SOMSRED-SVC uses a single speaker token per utterance because it is optimized in continuous space, i.e., $M = 1$ in Eq. (3) and $|T(k)| = 1$ in Eq. (8). The speaker embedding e_k and generation probability $P(\bar{s}_l | \bar{s}_{1:l-1}, \mathbf{X}; \Theta)$ are obtained in the same way as Eqs. (4)–(8). The speaker vector constraints loss is calculated as cosine similarity between \mathbf{u}_k and e_k defined as follows:

$$L_{\text{SVC}} = - \sum_k \frac{\mathbf{u}_k \cdot e_k}{\|\mathbf{u}_k\| \|e_k\|}, \quad (9)$$

where $\|\cdot\|$ denotes the L2 norm of the vector. The loss function of the model is defined as

$$L = L_{\text{CE}} + \alpha L_{\text{SVC}}, \quad (10)$$

where α is the hyperparameter representing the loss weight.

4. Experiment

4.1. Dataset

We used the Corpus of Spontaneous Japanese (CSJ) [27] for our experiments. First, we divided CSJ into training, validation, and test data: 1,388 speakers for the training data (522 h), ten speakers for the validation data (1.3 h), and ten speakers for the test data (1.9 h). Since CSJ is a dataset of single-talker’s speech, we created two-speaker and three-speaker simulated mixtures for training data and validation data by mixing the utterances of different speakers. When mixing the audio signals, the original volume of each utterance was kept unchanged, resulting in an average signal-to-interference ratio of about 0 dB. Delay values for each utterance were randomly chosen under the constraints provided in [8]. The start times of individual utterances differed by 0.5 s or longer so that every utterance in each mixed audio sample had at least one speaker-overlapped region with other utterances. The average overlap rate of the

mixture was about 35%. To train the speaker model for extracting the target speaker embeddings \mathbf{u}_k , we used data containing 9,332 speakers with over two million utterances from CSJ, VoxCeleb2 [28], and an internal dataset. We utilized 80 log mel-scale filterbank coefficients as acoustic features, which were extracted using a 20-ms-long Hann window with a 10-ms-long shift. We used characters as textual tokens. Continuous timestamps were rounded every 0.5 s; i.e., $Q = 0.5$.

4.2. Implementation

We used a Transformer-based ASR model [25, 26] for the experiments. The acoustic feature was first passed to layers composed of two 1×1 convolutions with 1×1 strides, two max pooling with a stride of 2, two 3×3 depthwise convolutions with 1×1 strides, and two long-short term memory layers with outputs of 512 dimensions. Then, we stacked ten-layer Transformer encoder blocks, where the number of heads in the multi-head attention was set to 4, the dimensions of the output continuous representations were set to 512, and the dimensions of the inner output in the position-wise feed-forward networks were set to 1,024. For the decoder layers, we stacked two-layer Transformer decoder blocks, where the settings were the same as for the encoder blocks.

The speaker model to extract the target speaker embeddings \mathbf{u}_k consists of four Transformer encoder blocks, an average layer that calculates the weighted mean of the frame-level features [29], and two linear layers. In the encoder blocks, the number of heads in the multi-head attention was set to 4, the dimensions of the output continuous representations were set to 512, and the dimensions of the inner output in the position-wise feed-forward networks were set to 2,048. We trained the model with the arcFace loss function [30] and the last linear layer was removed during inference, resulting in speaker embeddings of 512 dimensions.

4.3. Settings

We compared SOMSRED-SVC to SOMSRED. We prepared the speaker tokens for SOMSRED following the procedure in [18]. First, we trained the speaker classification model for speaker embedding extraction. Second, we estimated the speaker embeddings of non-overlapping speeches with over 410,000 utterances including an internal dataset. Third, the speaker embeddings were clustered with k-means clustering and 1,000 centroids were obtained. Finally, we assigned the M closest centroid indices that best reconstructs the embeddings of each training data before mixing. All models were optimized using the RAdam [31] algorithm with a mini-batch size of 32. We set the learning rate of the algorithm to $1e - 4$. The training steps were stopped if the loss on the validation set did not decrease for ten epochs in succession. We applied label smoothing with the smoothing weight of 0.1 [32].

We used the character error rate (CER), time stamp error rate (TER), equal error rate (EER), and speaker count accuracy (SCA) to evaluate the overall performance. TER was calculated as the sum of the missed speaker rate and false alarm rate; i.e., it equals to diarization error rate excluding the speaker error rate. When comparing hypothesized boundaries to references, we utilized a tolerance of ± 250 ms. SCA was calculated as the ratio of the number of test samples for which each method correctly counted the speaker to the total number of test samples. In multi-talker overlapped ASR settings, we compared hypotheses with references while considering the order of utterances. When calculating CER, we only evaluated textual tokens

Table 2: Evaluation results on multi-talker ASR training.

Number of speakers in test dataset	Methods	CER (%)	TER (%)	EER (%)	SCA (%)
1	SOMSRED ($M = 1$)	6.2	2.1	11.3	99.7
	SOMSRED ($M = 3$)	6.4	2.7	8.6	100.0
	SOMSRED ($M = 5$)	7.3	2.5	7.9	99.6
	SOMSRED-SVC	6.0	2.7	7.0	99.8
2	SOMSRED ($M = 1$)	9.1	3.4	13.0	98.0
	SOMSRED ($M = 3$)	10.2	4.0	10.0	98.0
	SOMSRED ($M = 5$)	11.1	3.3	10.3	97.4
	SOMSRED-SVC	8.8	3.6	9.5	98.6
3	SOMSRED ($M = 1$)	13.2	4.1	14.6	93.4
	SOMSRED ($M = 3$)	15.5	5.2	10.7	92.3
	SOMSRED ($M = 5$)	16.2	4.3	10.9	93.8
	SOMSRED-SVC	12.9	4.4	9.8	94.9

Table 1: CER and EER on validation data w.r.t. loss weight α .

Method	CER (%)	EER (%)
SOMSRED-SVC ($\alpha = 0.1$)	5.4	7.8
SOMSRED-SVC ($\alpha = 1$)	5.3	6.4
SOMSRED-SVC ($\alpha = 10$)	5.3	5.1
SOMSRED-SVC ($\alpha = 100$)	6.6	4.0

Table 3: Prediction accuracy (%) of speaker tokens on training data.

Methods	1 speaker	2 speakers	3 speakers
SOMSRED ($M = 1$)	74.4	71.3	66.2
SOMSRED-SVC	58.3	54.0	52.2

excluding the special token \mathcal{O} , time-stamp token \mathcal{T} , and speaker tokens \mathcal{D} .

For SOMSRED-SVC, we experimented with the loss weight $\alpha = 0.1, 1, 10, 100$. Table 1 shows the CER and EER results on single-talker’s speech of validation data trained with different α . We can see that a large loss weight leads to a better EER, but setting it too high deteriorates the ASR performance. Considering the balance between CER and EER, we set $\alpha = 10$.

4.4. Results

Table 2 shows the evaluation results. For reference, the EER of the speaker model used to generate the target u_k was 4.8%, which represents the lower bound. As we can see, SOMSRED-SVC outperforms SOMSRED in terms of both CER and EER while achieving a comparable performance in terms of TER and SCA, which means that SOMSRED-SVC improves the performance of ASR while making the speaker embeddings more distinctive. Moreover, although SOMSRED-SVC uses a single speaker token ($M = 1$), the speaker embedding becomes more distinctive than that of SOMSRED, which uses five speaker tokens ($M = 5$).

There is clearly a trade-off between CER and EER regarding the number of speaker tokens M in SOMSRED: as M increases, EER improves while CER degrades. We assume this is because although the large M represents the speaker embedding more precisely in discretized speaker embedding space, it is difficult for the model to estimate the residual part, i.e., $M > 1$, of the discretized speaker embedding, which makes the training difficult and degrades the ASR performance. In

contrast, since SOMSRED-SVC directly optimizes speaker embeddings in continuous space, the speaker embeddings become more distinctive. Moreover, comparing SOMSRED ($M = 1$) to SOMSRED-SVC, we can see that the direct optimization improves the ASR performance even when the same number of speaker token is used. We assume this is because the introduced loss helps avoid overfitting to the speaker tokens of the training data. Table 3 shows the accuracy of the predicted speaker token on the training dataset. Note that the speaker token is not used for inference in either SOMSRED or SOMSRED-SVC, where the speaker embeddings obtained from intermediate features are utilized to assign speaker tags. We can see that the accuracy of SOMSRED is higher than that of SOMSRED-SVC, which is presumably what degrades the ASR performance and speaker embedding performance. Since speaker tokens correspond to the point of the discretized speaker embeddings space, poorly fitting speaker tokens are assigned when the utterance is far from any point in a discrete speaker embedding space. The proposed training reduces the weight to estimate these speaker tokens while increasing the weight to estimate speaker embeddings in continuous space, which helps avoid the overfitting to the speaker token and improves the ASR and speaker embedding performance.

5. Conclusion

In this paper, we proposed SOMSRED-SVC, which is a sequential output model trained with speaker vector constraints loss for the joint modeling of multi-talker overlapped speech recognition and speaker diarization. The conventional method, SOMSRED, discretizes the speaker embedding space to treat the speaker embeddings as tokens, and although this enables SOMSRED to estimate speaker embeddings even from fully overlapped speeches and outperform other methods that utilize only non-overlapped speech segments for speaker embedding estimation, the speaker embedding obtained with SOMSRED becomes less distinctive compared to the speaker embedding directly obtained from non-overlapping speeches. To address this problem, we introduced a new training criterion that directly optimizes speaker embeddings in continuous space without discretization. The model is trained with the weighted sum of the conventional cross-entropy loss and the newly introduced loss for speaker embedding. Experimental results on the CSJ dataset demonstrate that SOMSRED-SVC outperforms SOMSRED in terms of both ASR performance and speaker embedding performance.

6. References

- [1] Y. Z. Isik, J. L. Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-channel multi-speaker separation using deep clustering," in *Proc. Interspeech*, 2016, pp. 545–549.
- [2] H. Seki, T. Hori, S. Watanabe, J. L. Roux, and J. R. Hershey, "A purely end-to-end system for multi-speaker speech recognition," in *Proc. ACL*, 2018, pp. 2620–2630.
- [3] S. Settle, J. L. Roux, T. Hori, S. Watanabe, and J. R. Hershey, "End-to-end multi-speaker speech recognition," in *Proc. ICASSP*, 2018, pp. 4819–4823.
- [4] X. Chang, Y. Qian, K. Yu, and S. Watanabe, "End-to-end monaural multi-speaker ASR system without pretraining," in *Proc. ICASSP*, 2019, pp. 6256–6260.
- [5] D. Raj, P. Denisov, Z. Chen, H. Erdogan, Z. Huang, M. He, S. Watanabe, J. Du, T. Yoshioka, Y. Luo, N. Kanda, J. Li, S. Wisdom, and J. R. Hershey, "Integration of speech separation, diarization, and recognition for multi-speaker meetings: System description, comparison, and analysis," in *Proc. SLT*, 2021, pp. 897–904.
- [6] S. Chen, Y. Wu, Z. Chen, J. Wu, J. Li, T. Yoshioka, C. Wang, S. Liu, and M. Zhou, "Continuous speech separation with conformer," in *Proc. ICASSP*, 2021, pp. 5749–5753.
- [7] I. Sklyar, A. Piunova, and Y. Liu, "Streaming multi-speaker ASR with RNN-T," in *Proc. ICASSP*, 2021, pp. 6903–6907.
- [8] N. Kanda, Y. Gaur, X. Wang, Z. Meng, and T. Yoshioka, "Serialized output training for end-to-end overlapped speech recognition," in *Proc. Interspeech*, 2020, pp. 2797–2801.
- [9] K. Boakye, B. Trueba-Hornero, O. Vinyals, and G. Friedland, "Overlapped speech detection for improved speaker diarization in multiparty meetings," in *Proc. ICASSP*, 2008, pp. 4353–4356.
- [10] V. Andrei, H. Cucu, and C. Burileanu, "Detecting overlapped speech on short timeframes using deep learning," in *Proc. Interspeech*, 2017, pp. 1198–1202.
- [11] 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, pp. 4300–4304.
- [12] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and P. García, "Encoder-decoder based attractors for end-to-end neural diarization," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 30, pp. 1493–1507, 2022.
- [13] L. E. Shafey, H. Soltan, and I. Shafran, "Joint speech recognition and speaker diarization via sequence transduction," in *Proc. Interspeech*, 2019, pp. 396–400.
- [14] N. Kanda, X. Xiao, Y. Gaur, X. Wang, Z. Meng, Z. Chen, and T. Yoshioka, "Transcribe-to-diarize: Neural speaker diarization for unlimited number of speakers using end-to-end speaker-attributed ASR," in *Proc. ICASSP*, 2022, pp. 8082–8086.
- [15] A. Khare, E. Han, Y. Yang, and A. Stolcke, "ASR-aware end-to-end neural diarization," in *Proc. ICASSP*, 2022, pp. 8092–8096.
- [16] N. Makishima, K. Suzuki, S. Suzuki, A. Ando, and R. Masumura, "Joint autoregressive modeling of end-to-end multi-talker overlapped speech recognition and utterance-level timestamp prediction," in *Proc. Interspeech*, 2023, pp. 2913–2917.
- [17] S. Cornell, J. Jung, S. Watanabe, and S. Squartini, "One model to rule them all ? towards end-to-end joint speaker diarization and speech recognition," in *Proc. ICASSP*, 2024, pp. 11 856–11 860.
- [18] N. Makishima, N. Kawata, M. Ihori, T. Tanaka, S. Orihashi, A. Ando, and R. Masumura, "SOMSRED: Sequential output modeling for joint multi-talker overlapped speech recognition and speaker diarization," in *Proc. Interspeech*, 2024, pp. 1660–1664.
- [19] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.
- [20] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Proc. SLT*, 2016, pp. 165–170.
- [21] N. Ryant, P. Singh, V. Krishnamohan, R. Varma, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, "The third DI-HARD diarization challenge," in *Proc. Interspeech*, 2021, pp. 3570–3574.
- [22] K. Kinoshita, M. Delcroix, and N. Tawara, "Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds," in *Proc. ICASSP*, pp. 7198–7202.
- [23] T. Cord-Landwehr, C. Bøddeker, C. Zorila, R. Doddipatla, and R. Haeb-Umbach, "Frame-wise and overlap-robust speaker embeddings for meeting diarization," in *Proc. ICASSP*, 2023, pp. 1–5.
- [24] A. Tripathi, H. Lu, and H. Sak, "End-to-end multi-talker overlapping speech recognition," in *Proc. ICASSP*, 2020, pp. 6129–6133.
- [25] 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, pp. 5998–6008.
- [26] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. ICASSP*, 2018, pp. 5884–5888.
- [27] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of Japanese," in *Proc. LREC*, 2000.
- [28] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech*, pp. 1086–1090.
- [29] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech*, 2018, pp. 2252–2256.
- [30] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. CVPR*, 2019, pp. 4690–4699.
- [31] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," in *Proc. ICLR*, 2020.
- [32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. CVPR*, 2016, pp. 2818–2826.