Interspeech 2024
1-5 September 2024, Kos, Greece

SOMSRED: Sequential Output Modeling for Joint Multi-talker Overlapped Speech Recognition and Speaker Diarization

Naoki Makishima, Naotaka Kawata, Mana Ihori, Tomohiro Tanaka, Shota Orihashi, Atsushi Ando, Ryo Masumura

NTT Corporation, Japan

naoki.makishima@ntt.com

Abstract

This paper proposes SOMSRED, which jointly models the multi-talker automatic speech recognition (ASR) and speaker diarization (SD) for fully overlapped speech of unknown speakers. The conventional method that jointly estimates ASR and SD requires non-overlapping speech and a separate clustering-based SD component for accurately identifying speakers. However, the speech is often overlapped, which deteriorates speaker identification performance, and the separate model makes the whole system sub-optimal. To address this problem, our idea is to build a sequential output model that outputs transcriptions, timestamps, and newly introduced speaker identifiers recursively from overlapped speech. Since speaker identifier do not fully represent the speaker characteristics of unknown speakers, SOMSRED utilizes the intermediate feature as speaker embeddings. Experimental results show the efficacy of the proposed method in speaker recognition, SD, and multi-talker ASR.
Index Terms: speech recognition, speaker diarization, speaker recognition

1. Introduction

Speaker diarization (SD) and automatic speech recognition (ASR) are tasks to figure out who spoke when and what, respectively [1–4]. Since our natural conversations often include overlapping speech, where several people speak simultaneously, studies on these tasks have been conducted to address overlapping speech.

One of the most common frameworks is to use the pipeline system that comprises SD, speech separation, and ASR [5–10]. Although existing models can be used in this approach, the system becomes complex, and the whole model is difficult to optimize jointly. Recently, serialized output training (SOT) [11] was proposed to handle multi-talker ASR without using speech separation. Instead of having independent multiple output layers that output transcriptions of each speaker, SOT generates the transcription of multiple speakers recursively one after another with a single output layer, which enables simple and natural modeling of multi-talker ASR.

Various studies have extended SOT to reduce the components of the SD and ASR pipeline and estimate who spoke when and what simultaneously. In [3], end-to-end-neural-based diarization that estimates frame-wise speaker activity [12, 13] is combined with multi-talker ASR using SOT. Although they jointly model ASD and SD of known speakers, to avoid the permutation problem of end-to-end-neural-based diarization, they require a speaker inventory, which cannot be prepared for unknown speakers. A common approach to avoid permutation while handling unknown speakers is to utilize clustering-based diarization [14,15]. In clustering-based diarization, speaker embedding extraction from non-overlapping speech is followed by clustering the embeddings to assign the speaker identity. To perform clustering-based diarization, non-overlapping speech regions need to be detected for high accuracy clustering [16]. Various studies have addressed the problem of detecting non-overlapping speech regions [15, 17–19]. Among them, predicting the quantized timestamp token [20] achieves the promising performance [18] with the same architecture as the simple single-talker ASR. In [18], SOT is extended to estimate utterance-level timestamps of each speaker as well as transcriptions in the overlapped speech, which reveals the non-overlapping speech regions.

However, the limitation of these studies is that a cascaded pipeline of the timestamp prediction and speaker embedding extraction is required, which makes the system complex. Moreover, non-overlapping speech regions are required for extracting speaker embeddings, but such regions are often short or do not exist in overlapped speech. Our key idea to solve these problems is to build a sequential output model that outputs transcriptions, timestamps, and newly introduced speaker identifiers recursively from overlapped speech. We introduce speaker identifier tokens representing speaker characteristics before overlap, which we call speaker tokens. The introduced speaker tokens are estimated in the same way as transcriptions of multi-talker ASR with the SOT framework.

In this paper, we propose the sequential output modeling for multi-talker ASR and SD that handles fully overlapped speeches with a single model, which we call SOMSRED (sequential output modeling of speech recognition and speaker diarization). SOMSRED estimates not only transcriptions and timestamps but also speaker tokens as the single output sequence. The newly introduced speaker tokens correspond to the specific speaker embeddings, and the model is trained to estimate the closest speaker tokens whose embeddings are close to that of each utterance. Although the model is trained to output the speaker tokens, the tokens are obtained from training data, and their accuracy would be degraded if applied to unknown speakers during inference. Thus, we utilize the intermediate feature before the classification layer as speaker embeddings, which is expected to reflect speaker characteristics like speaker recognition model [21, 22]. We experimentally show the efficacy of this approach in Section 4. The advantages of SOMSRED are three-fold. First, the same simple architecture as the conventional single-talker ASR is utilized for jointly modeling multi-talker ASR, timestamp prediction, and speaker embedding extraction by extending the SOT framework. Second, although SOMSRED conducts clustering-based diarization, the model estimates speaker embeddings of each speaker considering speech overlap. Third, the estimated speaker embeddings are applicable to many applications besides SD such as speaker

10.21437/Interspeech.2024-436

recognition. We experimentally valid SOMSRED in speaker recognition, speaker diarization, and multi-talker ASR.

2. Conventional methods

2.1. Multi-talker ASR with autoregressive modeling

We denote the acoustic feature of the input speech as $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_T)$, where $\boldsymbol{x}_t \in \mathbb{R}^F$ denotes the tth 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 $\boldsymbol{W}^{1:K} = (\boldsymbol{W}^1, \dots, \boldsymbol{W}^K)$, where K denotes the number of speakers in the overlapped speech, $\boldsymbol{W}^k = (w_1^k, \dots, w_{N^k}^k)$ denotes the kth speaker's textual tokens, N^k denotes the length of the token, $w_n^k \in \mathcal{V}$ denotes the nth textual token of the kth speaker, and $\mathcal{V}$ denotes the vocabulary set. To avoid permutation ambiguities in the order of k when predicting $\boldsymbol{W}^k$, the first-in, first-out approach [11, 23] is adopted in SOT; the transcription is estimated in the order of the speakers' utterance start time. Moreover, to recognize multiple utterances with a single output layer, $\boldsymbol{W}^{1:K}$ is serialized into a single token sequence with a special symbol [sep] representing speaker change. The serialized token $\boldsymbol{S} \in \{\mathcal{V} \cup \mathcal{O}\}$ is given as

$$\begin{aligned}\boldsymbol{S} = (&w_1^1, \dots, w_{N^1}^1, [\text{sep}], w_1^2, \dots, w_{N^2}^2, [\text{sep}], \\ &\dots, w_{N^{K-1}}^{K-1}, [\text{sep}], w_1^K, \dots, w_{N^K}^K, [\text{eos}]),\end{aligned} \tag{1}$$

where [eos] denotes the end of a sentence, $\mathcal{O} = \{[\text{sep}], [\text{eos}]\}$, and we assume that $\boldsymbol{W}^{1:K}$ is sorted in order of utterance start times for simplicity.

Multi-talker ASR with SOT estimates generation probability of $\boldsymbol{S}$ given $\boldsymbol{X}$ as follows:

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

where s_l denotes the lth token of $\boldsymbol{S}$, $\boldsymbol{s}_{1:l-1} = (s_1, \dots, s_{l-1})$, $|\boldsymbol{S}|$ denotes the length of $\boldsymbol{S}$, and $\boldsymbol{\Theta}_{\text{MT}}$ denotes the parameter of the multi-talker ASR model. The parameter $\boldsymbol{\Theta}_{\text{MT}}$ is optimized with the following cross-entropy function:

$$L_{\text{MT}} = -\log P(\boldsymbol{S}|\boldsymbol{X}; \boldsymbol{\Theta}_{\text{MT}}). \tag{3}$$

2.2. Joint modelinng of multi-talker ASR and timestamp prediction

Multi-talker ASR with SOT has been extended to estimate not only transcriptions but also timestamps of utterances [14, 18, 19]. The predicted timestamps are useful for speech segmentation to obtain speaker-uniform regions in SD. In [18], quantized timestamp tokens [20] are used to efficiently model the joint generation probability of transcriptions and utterances. The serialized label sequences $\tilde{\boldsymbol{S}} \in \{\mathcal{V} \cup \mathcal{O} \cup \mathcal{T}\}$ to estimate are denoted as

$$\begin{aligned}\tilde{\boldsymbol{S}} = (&[t_{\text{s}}^1], [t_{\text{e}}^1], w_1^1, \dots, w_{N^1}^1, [\text{sep}], \\ &[t_{\text{s}}^2], [t_{\text{e}}^2], w_1^2, \dots, w_{N^2}^2, [\text{sep}], \\ &\dots, [t_{\text{s}}^K], [t_{\text{e}}^K], w_1^K, \dots, w_{N^K}^K, [\text{eos}]),\end{aligned} \tag{4}$$

where $\mathcal{T}$ denotes the quantized time token label set, and $[t_{\text{s}}^k] \in \mathcal{T}$ and $[t_{\text{e}}^k] \in \mathcal{T}$ denote the kth speaker's start time token and end time token, respectively. We denote the start time token and the end time token of multiple speakers as $\boldsymbol{T}_{\text{s}}^{1:K} = ([t_{\text{s}}^1], \dots, [t_{\text{s}}^K])$ and $\boldsymbol{T}_{\text{e}}^{1:K} = ([t_{\text{e}}^1], \dots, [t_{\text{e}}^K])$, respectively. The

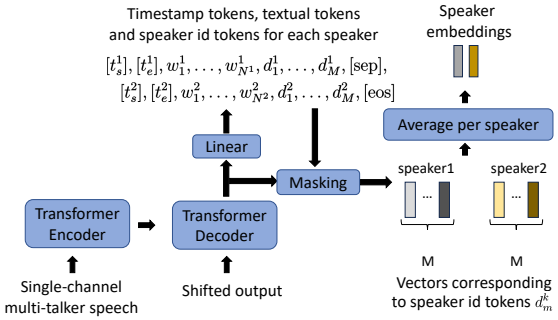

Figure 1: *Overview of SOMSRED.*

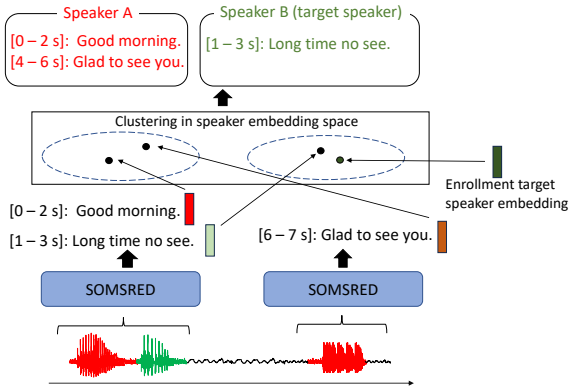

Figure 2: *Speaker diarization and speaker verification with SOMSRED.*

joint generation probability of $\boldsymbol{W}^{1:K}$, $\boldsymbol{T}_{\text{s}}^{1:K}$, and $\boldsymbol{T}_{\text{e}}^{1:K}$ given multi-talker overlapped speech $\boldsymbol{X}$ is obtained as

$$P(\tilde{\boldsymbol{S}}|\boldsymbol{X}; \boldsymbol{\Theta}) = \prod_{l=1}^{|\tilde{\boldsymbol{S}}|} P(\tilde{s}_l | \tilde{\boldsymbol{s}}_{1:l-1}, \boldsymbol{X}; \boldsymbol{\Theta}_{\text{MTT}}), \tag{5}$$

where $\tilde{s}_l$ denotes the lth token of $\tilde{\boldsymbol{S}}$, $\tilde{\boldsymbol{s}}_{1:l-1} = (\tilde{s}_1, \dots, \tilde{s}_{l-1})$, $|\tilde{\boldsymbol{S}}|$ denotes the length of $\tilde{\boldsymbol{S}}$, and $\boldsymbol{\Theta}_{\text{MTT}}$ denotes the parameter of the model in the conventional method.

3. Proposed method

3.1. Strategy

Figure 1 shows an overview of SOMSRED. SOMSRED predicts the joint generation probability of multiple transcriptions, timestamps, and speaker tokens from single-channel overlapped speech. Although SOMSRED estimates speaker tokens, we do not use these tokens for speaker identification because their accuracy would be greatly degraded if applied to unknown speakers during inference. Thus, we utilize vectors corresponding to speaker tokens before the classification layer.

Speaker tokens are obtained by applying typical clustering methods such as k-means clustering to non-overlapping speech. Although a large k value captures speaker characteristics precisely, the vocabulary size including speaker tokens also becomes large. Thus, we use multiple speaker tokens to represent the speaker embeddings of each speaker with a small k value; i.e., the average of the multiple speaker embeddings becomes close to the target speaker embedding.

3.2. Formulation

We denote the multiple speaker tokens as $\boldsymbol{d}^{1:K} = (\boldsymbol{d}^1, \ldots, \boldsymbol{d}^K)$, where $\boldsymbol{d}^k = (d_1^k, \ldots, d_M^k)$ denotes the kth speaker's speaker tokens, $d_m^k \in \mathcal{D}$ denotes the mth speaker token, $\mathcal{D}$ denotes the speaker token label set, and M denotes the number of tokens per speaker. To efficiently model the joint generation probability of $\boldsymbol{W}^{1:K}$, $\boldsymbol{T}_{\mathrm{s}}^{1:K}$, $\boldsymbol{T}_{\mathrm{e}}^{1:K}$, and $\boldsymbol{d}^{1:K}$, we serialize them into a single label sequence as SOT [11]. The serialized label sequence $\bar{\boldsymbol{S}} \in \{\mathcal{V} \cup \mathcal{O} \cup \mathcal{T} \cup \mathcal{D}\}$ is obtained as

$$\begin{aligned}\bar{\boldsymbol{S}} = ([t_{\mathrm{s}}^1], [t_{\mathrm{e}}^1], w_1^1, \ldots, w_{N^1}^1, d_1^1, \ldots, d_M^1, [\mathrm{sep}], \\ \ldots, [t_{\mathrm{s}}^K], [t_{\mathrm{e}}^K], \ldots, w_{N^K}^K, d_1^K, \ldots, d_M^K, [\mathrm{eos}]).\end{aligned} \quad (6)$$

The joint generation probability of $\boldsymbol{W}^{1:K}$, $\boldsymbol{T}_{\mathrm{s}}^{1:K}$, $\boldsymbol{T}_{\mathrm{e}}^{1:K}$, and $\boldsymbol{d}^{1:K}$ is autoregressively estimated as

$$P(\bar{\boldsymbol{S}}|\boldsymbol{X}; \boldsymbol{\Theta}) = \prod_{l=1}^{|\bar{\boldsymbol{S}}|} P(\bar{s}_l | \bar{\boldsymbol{s}}_{1:l-1}, \boldsymbol{X}; \boldsymbol{\Theta}), \quad (7)$$

where $\bar{s}_l$ denotes the lth token of $\bar{\boldsymbol{S}}$, $\bar{\boldsymbol{s}}_{1:l-1} = (\bar{s}_1, \ldots, \bar{s}_{l-1})$, $|\bar{\boldsymbol{S}}|$ denotes the length of $\bar{\boldsymbol{S}}$, and $\boldsymbol{\Theta}$ denotes the parameter of SOMSRED.

3.3. Modeling

We use a Transformer-based ASR model [24, 25]. The joint generation probability is obtained as follows:

$$\boldsymbol{H} = \mathrm{TransformerEnc}(\boldsymbol{X}; \boldsymbol{\theta}_{\mathrm{enc}}), \quad (8)$$

$$\boldsymbol{E} = \mathrm{TransformerDec}(\boldsymbol{H}, \bar{\boldsymbol{s}}_{1:v-1}; \boldsymbol{\theta}_{\mathrm{dec}}), \quad (9)$$

$$P(\bar{s}_l | \bar{\boldsymbol{s}}_{1:l-1}, \boldsymbol{X}; \boldsymbol{\Theta}) = \mathrm{Linear}(\boldsymbol{E}; \boldsymbol{\theta}_{\mathrm{linear}}), \quad (10)$$

where $\mathrm{TransformerEnc}(\cdot)$ is a Transformer encoder, $\boldsymbol{\theta}_{\mathrm{enc}}$ denotes its parameters, $\mathrm{TransformerDec}(\cdot)$ is a Transformer decoder, $\boldsymbol{\theta}_{\mathrm{dec}}$ denotes its parameters, $\mathrm{Linear}(\cdot)$ denotes a linear layer with softmax activation, and $\boldsymbol{\theta}_{\mathrm{linear}}$ denotes its parameter. We describe the detailed architecture of the model in Section 4.2. Instead of speaker tokens, the speaker embeddings are used for speaker identification in inference. The speaker embedding is obtained by averaging the features before the classification layer as follows:

$$\boldsymbol{e}_k = \frac{1}{|I(k)|} \sum_{v \in I(k)} \boldsymbol{E}_v, \quad (11)$$

where $\boldsymbol{e}_k$ denotes the speaker embedding of the kth speaker, $I(k)$ denotes the decoding steps corresponding to the kth speaker tokens estimation, $|I(k)|$ denotes its size, and $\boldsymbol{E}_v$ denotes the vth element of $\boldsymbol{E}$. The parameter $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_{\mathrm{enc}}, \boldsymbol{\theta}_{\mathrm{dec}}, \boldsymbol{\theta}_{\mathrm{linear}}\}$, is optimized with the cross-entropy function that is defined as

$$L_{\mathrm{CE}} = -\log P(\bar{\boldsymbol{S}}|\boldsymbol{X}; \boldsymbol{\Theta}). \quad (12)$$

The obtained speaker embeddings can be used in various applications including SD and speaker recognition as shown in Fig. 2. For example, SD is performed by clustering the embeddings with a typical clustering algorithm such as agglomerative hierarchical clustering (AHC), which reveals who spoke when and what simultaneously along with the ASR results of SOMSRED. Moreover, the target speaker is verified by calculating the similarity score of the speaker embeddings of the output of SOMSRED and that of the pre-registered speakers. We experimentally verify SOMSRED in a SD task and speaker recognition task in Section 4.

Table 1: *EER (%) of speaker embeddings obtained with speaker tokens and proposed method.*

Method	Number of speakers		
	1	2	3
Speaker tokens	22.9	22.9	23.1
SOMSRED	8.0	9.3	10.3

4. Experiment

4.1. Dataset

We evaluated SOMSRED by conducting multi-talker ASR, speaker recognition, and SD. We used the Corpus of Spontaneous Japanese (CSJ) [26] for our experiments. First, we divided CSJ into training, validation, and test data. Training data consists of 1,388 speakers, and its size is 522 h. Validation data consists of 10 speakers, and its size is 1.3 h, and test data consists of 10 speakers, and its size is 1.9 h. Since the CSJ is a dataset for single-talker ASR, 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. As for the delay applied to each utterance, the delay values were randomly chosen under the same constraints as in [11]. First, the start times of individual utterances differed by 0.5 s or longer. Second, 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 %. We prepared test data for speaker recognition with ASR and SD with ASR. In the speaker recognition task, we prepared test data in the same way as [18]. In the SD task, we created a two-speaker test set and three-speaker test set. We created a two-speaker mixture from six randomly selected speakers so that about one-third of the speech was the mixture. The average overlap ratio is about 9.2%, and total duration is about 42 minutes, which contains 625 speech. We created a three-speaker mixture from six randomly selected speakers so that about one-fourth of the speech was the mixture. The average overlap ratio of at least two speakers is about 18.2%, and total duration is about 24 minutes, which contains 238 speech. We used 80 log mel-scale filterbank coefficients as acoustic features. We rounded continuous timestamps every 0.5 s.

4.2. Implementation

We used a Transformer-based ASR model [24,25]. 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 256 dimensions. Then, we stacked 10-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 256, and the dimensions of the inner output in the position-wise feed-forward networks were set to 1,024. For decoder layers, we stacked two-layer Transformer decoder blocks, where the settings were the same as for the encoder blocks.

4.3. Settings

We compared three methods: conventional multi-talker ASR [11], conventional multi-talker ASR with timestamp pre-

Table 2: *Evaluation results for speaker recognition and ASR.*

Method	2 speaker				3 speaker			
	CER (%)	TER (%)	EER (%)	SCA (%)	CER (%)	TER (%)	EER (%)	SCA (%)
Multi-talker ASR	8.9	-	-	92.8	13.0	-	-	92.8
Multi-talker ASR + timestamp	8.0	2.33	17.9	96.3	12.2	3.76	24.6	98.8
SOMSRED	8.4	2.73	9.3	98.9	11.8	3.16	10.3	99.4

Table 3: *Evaluation results for SD and ASR.*

Method	2 speaker		3 speaker	
	DER (%)	cpCER (%)	DER (%)	cpCER (%)
Multi-talker ASR + timestamp	4.29	14.5	22.42	32.3
SOMSRED	2.41	11.7	3.11	15.7

diction [18], and SOMSRED. Moreover, we compared two approaches for speaker embedding estimation. One uses features before the classification layer for speaker embeddings as explained in Section 3.2, and the other directly uses speaker tokens to look up the speaker embedding.

To prepare speaker ids for training data, we conducted the following procedure. First, we trained the speaker classification model for speaker embedding extraction. This model is trained to classify 9332 speakers with over two million utterances including CSJ, VoxCeleb2 [27], and an internal dataset. The model consists of four Transformer encoder blocks, the average layer that calculates the weighted mean of the frame-level features [28], and the classification layer. We trained the model with ArcFace loss function [29], and the last classification layer is removed in inference. Second, we estimated speaker embeddings of training data before mixing with the speaker model. Third, the speaker embeddings were clustered with k-means clusterings and 1000 centroids were obtained. Finally, we assigned the three closest clusters' ids that most reconstruct the embeddings of each training data before mixing; i.e., we set M as three in this experiment.

In the conventional method with timestamp prediction, we used the same speaker model as the one used to obtain speaker tokens. We calculated the speaker embeddings with the estimated specific speaker speech. When non-overlapping speech do not exist for the specific speaker, we calculated the speaker embeddings with the overlapped speech.

All models were optimized by using the RAdam [30] algorithm with a minibatch 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 10 successive epochs. We applied label smoothing with the smoothing weight of 0.1 [31].

For speaker recognition and ASR task, we used the character error rate (CER), time stamp error rate (TER), equal error rate (EER), and speaker count accuracy (SCA) to evaluate the performance. TER was calculated as the sum of the missed speaker rate and false alarm rate; i.e., it equals the diarization error rate (DER) except speaker error rate. When comparing hypothesized boundaries to references, we used 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 excluding the special token $\mathcal{O}$, the time-stamp token $\mathcal{T}$, and the speaker token $\mathcal{D}$. For SD and ASR task, we used the concatenated minimum permutation CER (cpCER) [32] and DER for evaluation. We used a tolerance of ± 250 ms for calculating DER. We used AHC with the threshold of cosine similarity 0.35 to cluster speaker embeddings.

4.4. Results

Table 1 shows EER of two different approaches for estimating speaker embeddings. Results show that SOMSRED using features before the classification layer as speaker embeddings improves EER over the method that directly uses speaker tokens. This suggests that the features before the classification layer keep the speaker characteristics of unknown speakers although the speaker tokens cannot fully represent it.

Table 2 shows evaluation results for the speaker recognition and ASR task. Compared to the conventional method, SOMSRED improves EER by 8.6% and 14.3 % when the number of speakers is two and three, respectively, while achieving comparative CER, TER and SCA. This suggests that SOMSRED can estimate more discriminative speaker embeddings compared to the conventional method that fails to estimate discriminative speaker embeddings especially when the overlap ratio is high.

Table 3 shows the evaluation results for the SD and ASR task. The results show that SOMSRED outperforms the conventional method in both two-speaker and three-speaker cases. The performance of the conventional method degrades when the number of speakers and overlap ratio increases because of the decrease in the non-overlapping speech to estimate speaker embeddings. On the other hand, the results show that SOMSRED improves this problem, which shows the effectiveness of the training with speaker tokens and using speaker vectors before the classification layer.

5. Conclusions

In this paper, we proposed SOMSRED, which jointly models the multi-talker ASR and SD for fully overlapped speech of unknown speakers. SOMSRED outputs serialized transcriptions of multiple speakers, their quantized timestamp tokens, and speaker tokens one after another with a single layer, which enables simple and natural modeling of multi-talker ASR and SD without changing the simple autoregressive modeling of the conventional multi-talker ASR. Experimental results show that SOMSRED outperforms the conventional methods in terms of speaker recognition and SD while it performs comparably in terms of ASR.

6. References

[1] L. E. Shafey, H. Soltau, and I. Shafran, "Joint speech recognition and speaker diarization via sequence transduction," in *Proc. Interspeech*, 2019, pp. 396–400.

[2] H. H. Mao, S. Li, J. J. McAuley, and G. W. Cottrell, "Speech recognition and multi-speaker diarization of long conversations," in *Proc. Interspeech*, 2020, pp. 691–695.

[3] N. Kanda, Y. Gaur, X. Wang, Z. Meng, Z. Chen, T. Zhou, and T. Yoshioka, "Joint speaker counting, speech recognition, and speaker identification for overlapped speech of any number of speakers," in *Proc. Interspeech*, 2020, pp. 36–40.

[4] A. Khare, E. Han, Y. Yang, and A. Stolcke, "ASR-aware end-to-end neural diarization," in *Proc. ICASSP*, 2022, pp. 8092–8096.

[5] 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.

[6] 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.

[7] 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.

[8] C. Böddeker, A. S. Subramanian, G. Wichern, R. Haeb-Umbach, and J. L. Roux, "TS-SEP: joint diarization and separation conditioned on estimated speaker embeddings," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 32, pp. 1185–1197, 2024.

[9] F. Yu, S. Zhang, P. Guo, Y. Fu, Z. Du, S. Zheng, W. Huang, L. Xie, Z.-H. Tan, D. Wang, Y. Qian, K. A. Lee, Z. Yan, B. Ma, X. Xu, and H. Bu, "Summary on the ICASSP 2022 multi-channel multi-party meeting transcription grand challenge," in *Proc. ICASSP*, 2022, pp. 9156–9160.

[10] S. Cornell, M. Wiesner, S. Watanabe, D. Raj, X. Chang, P. Garcia, Y. Masuyama, Z.-Q. Wang, S. Squartini, and S. Khudanpur, "The CHiME-7 DASR challenge: Distant meeting transcription with multiple devices in diverse scenarios," *arXiv preprint arXiv:2306.13734*, 2023.

[11] 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.

[12] 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.

[13] 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.

[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] 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.

[16] 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.

[17] T. Cord-Landwehr, C. Böddeker, 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.

[18] 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.

[19] S. Cornell, J.-w. Jung, S. Watanabe, and S. Squartini, "One model to rule them all? Towards end-to-end joint speaker diarization and speech recognition," *arXiv preprint arXiv:2310.01688*, 2023.

[20] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.

[21] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. ICASSP*, 2014, pp. 4052–4056.

[22] 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.

[23] A. Tripathi, H. Lu, and H. Sak, "End-to-end multi-talker overlapping speech recognition," in *Proc. ICASSP*, 2020, pp. 6129–6133.

[24] 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.

[25] 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.

[26] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of Japanese," in *Proc. LREC*, 2000.

[27] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech*, pp. 1086–1090.

[28] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech*, 2018, pp. 2252–2256.

[29] 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.

[30] 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.

[31] 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.

[32] S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj *et al.*, "CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," *arXiv preprint arXiv:2004.09249*, 2020.