Joint Speaker Counting, Speech Recognition, and Speaker Identification for Overlapped Speech of Any Number of Speakers

Naoyuki Kanda, Yashesh Gaur, Xiaofei Wang, Zhong Meng,
Zhuo Chen, Tianyan Zhou, Takuya Yoshioka

Microsoft Corp.

{Naoyuki.Kanda,Yashesh.Gaur,Xiaofei.Wang,Zhong.Meng,zhuc,tizhou,tayoshio}@microsoft.com

Abstract

We propose an end-to-end speaker-attributed automatic speech recognition model that unifies speaker counting, speech recognition, and speaker identification on monaural overlapped speech. Our model is built on serialized output training (SOT) with attention-based encoder-decoder, a recently proposed method for recognizing overlapped speech comprising an arbitrary number of speakers. We extend SOT by introducing a speaker inventory as an auxiliary input to produce speaker labels as well as multi-speaker transcriptions. All model parameters are optimized by speaker-attributed maximum mutual information criterion, which represents a joint probability for overlapped speech recognition and speaker identification. Experiments on LibriSpeech corpus show that our proposed method achieves significantly better speaker-attributed word error rate than the baseline that separately performs overlapped speech recognition and speaker identification.

Index Terms: multi-speaker speech recognition, speaker counting, speaker identification, serialized output training

1. Introduction

Speaker-attributed automatic speech recognition (SA-ASR) from overlapped speech has been an active research area towards meeting transcription [1, 2, 3]. It requires to count the number of speakers, transcribe utterances that are sometimes overlapped, and also diarize or identify the speaker of each utterance. While significant progress has been made especially for multi-microphone settings (e.g., [4]), SA-ASR remains very challenging when we can only access monaural audio.

A significant amount of research has been conducted to achieve the goal of SA-ASR. One approach is applying speech separation (e.g., [5, 6, 7]) before ASR and speaker diarization/identification. However, a speech separation module is often designed with a signal-level criterion, which is not necessarily optimal for succeeding modules. To overcome this suboptimality, researchers have investigated approaches for jointly modeling multiple modules. For example, there are a number of studies concerning joint modeling of speech separation and ASR (e.g., [8, 9, 10, 11, 12, 13]). Several methods were also proposed for integrating speaker identification and speech separation [14, 15, 16]. However, little research has yet been done to address SA-ASR by combining *all* these modules.

Only a limited number of studies have tackled the joint modeling of multi-speaker ASR and speaker diarization/identification. [17] proposed to generate transcriptions of different speakers interleaved by speaker role tags to recognize two-speaker conversations based on a recurrent neural network transducer (RNN-T). Although promising results were shown for two-speaker conversation data, the method cannot deal with speech overlaps due to the monotonicity constraint of RNN-T. Furthermore, their method is difficult to be extended to an arbitrary number of speakers because the speaker role tag needs to be uniquely defined for each speaker (e.g., a doctor and a patient). [18] proposed a joint decoding framework for overlapped speech recognition and speaker diarization, in which speaker embedding estimation and target-speaker ASR were applied alternately. Although this method is extendable to many speakers in theory, it assumes that the speaker counting is conducted during the speaker embedding estimation process, which is challenging in practice.

In this paper, we propose an end-to-end SA-ASR model that unifies speaker counting, overlapped speech recognition, and speaker identification. Our model is built on serialized output training (SOT) [19] with attention-based encoder-decoder (AED) [20, 21, 22, 23], which was recently proposed for recognizing overlapped speech consisting of an arbitrary number of speakers. We extend the SOT model by introducing a speaker inventory as an auxiliary input to produce speaker labels as well as multi-speaker transcriptions. All model parameters are optimized by maximizing a joint probability for overlapped speech recognition and speaker identification. Our model can recognize overlapped speech of *any number of speakers* while identifying the speaker of each utterance among *any number of speaker profiles*. We show that the proposed model achieves significantly better speaker-attributed word error rate (SA-WER) over the model consisting of separate modules.

2. Overlapped Speech Recognition with Serialized Output Training

2.1. ASR based on Attention-based Encoder Decoder

Given input $X = \{x_1, ..., x_T\}$, an AED model produces a posterior probability of output sequence $Y = \{y_1, ..., y_n, ..., y_N\}$ as follows. Firstly, an encoder converts the input sequence X into a sequence, H^{enc}, of embeddings, i.e.,

$$H^{enc} = \{h_1^{enc}, ..., h_T^{enc}\} = \text{AsrEncoder}(X). \quad (1)$$

Secondly, at each decoder step n, the attention module outputs attention weight $\alpha_n = \{\alpha_{n,1}, ..., \alpha_{n,T}\}$ as

$$\alpha_n = \text{Attention}(u_n, \alpha_{n-1}, H^{enc}), \quad (2)$$

$$u_n = \text{DecoderRNN}(y_{n-1}, c_{n-1}, u_{n-1}), \quad (3)$$

where u_n is a decoder state vector at n-th step, and c_{n-1} is the context vector at the previous time step. Then, context vector c_n for the current time step n is generated as a weighted sum of the encoder embeddings as follows.

$$c_n = \sum_{t=1}^{T} \alpha_{n,t} h_t^{enc}. \quad (4)$$

Finally, the output distribution for y_n is estimated given the context vector c_n and decoder state vector u_n as follows:

$$Pr(y_n|y_{1:n-1}, X) \sim \text{DecoderOut}(c_n, u_n)$$
$$= \text{Softmax}(W_{out} \cdot \text{LSTM}(c_n + u_n)). \quad (5)$$

Here, we are assuming that c_n and u_n have the same dimensionality. Variable W_{out} is the affine matrix of the final layer. Note that DecoderOut normally consists of a single affine transform with a softmax output layer. However, it was found in [19] that inserting one LSTM just before the affine transform effectively improves the SOT model, so we follow that architecture.

2.2. Serialized Output Training

With the SOT framework, the references for multiple overlapped utterances are concatenated to form a single token sequence by inserting a special symbol $\langle sc \rangle$ representing a speaker change. For example, for the three-speaker case, the reference label will be given as $R = \{r_1^1, .., r_{N^1}^1, \langle sc \rangle, r_1^2, .., r_{N^2}^2, \langle sc \rangle, r_1^3, .., r_{N^3}^3, \langle eos \rangle\}$, where r_i^j represents i-th token of j-th utterance. Note that $\langle eos \rangle$, a token for sequence end, is used only at the end of the entire sequence.

Because there are multiple permutations in the order of reference labels to form R, some trick is needed to calculate the loss for AED. One simple yet effective approach in [19] is sorting the reference labels by their start times, which is called "first-in, first-out" (FIFO) training. This training scheme works with complexity of $O(S)$ with respect to the number of speakers and outperforms a scheme that exhaustively considers all possible permutations [19]. In this paper, we always use this FIFO training scheme.

Unlike permutation invariant training [7, 8], in which the number of the output branches that a model has constrains the number of recognizable speakers, SOT has no such limitation. Refer to [19] for a detailed description of the SOT model.

3. Proposed method

3.1. Overview

Suppose that we have a speaker inventory $\mathcal{D} = \{d_1, ..., d_K\}$, where K is the number of speakers in the inventory and d_k is a speaker profile vector (e.g., d-vector [24]) of the k-th speaker. The goal of the proposed method is to estimate a serialized multi-speaker transcription Y accompanied by the speaker identity of each token $S = \{s_1, ..., s_N\}$ given input X and $\mathcal{D}$.

In this work, we assume that the profiles of all speakers involved in the input speech are included in $\mathcal{D}$. In other words, we assume there is no "unknown" speaker for speaker identification. As long as this condition holds, the speaker inventory may include any number of irrelevant speakers' profiles. This is a typical setup in scheduled office meetings, where meeting organizers invite attendees whose voice profiles are pre-registered.

3.2. Model Architecture

We start with the conventional AED represented by the blue blocks in Fig. 1. Firstly, we introduce one more encoder to represent the speaker characteristics of input X as follows.

$$H^{spk} = \{h_1^{spk}, ..., h_T^{spk}\} = \text{SpeakerEncoder}(X), \quad (6)$$

On top of this, for every decoder step n, we apply the attention weight α_n generated by the attention module of AED to extract
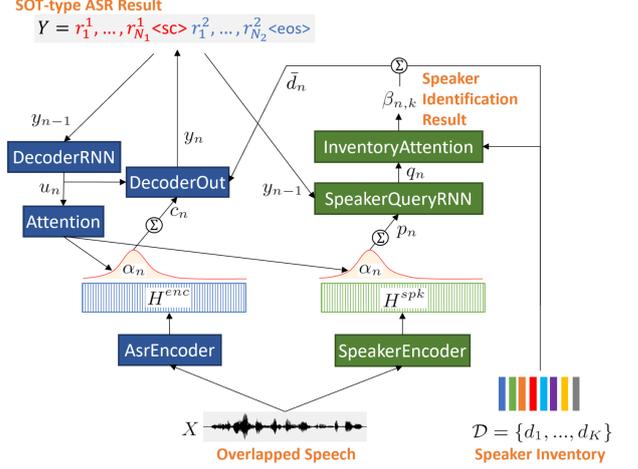

Figure 1: *Proposed model.*

attention-weighted vector of speaker embeddings p_n.

$$p_n = \sum_{t=1}^{T} \alpha_{n,t} h_t^{spk}. \quad (7)$$

Note that p_n could be contaminated by interfering speech because some time frames include two or more speakers.

The speaker query RNN in Fig. 1 then generates a speaker query q_n given the speaker embedding p_n, previous output y_{n-1}, and previous speaker query q_{n-1}.

$$q_n = \text{SpeakerQueryRNN}(p_n, y_{n-1}, q_{n-1}). \quad (8)$$

Based on the speaker query q_n, an attention module for speaker inventory (shown as InventoryAttention in the diagram) estimates the attention weight $\beta_{n,k}$ for each profile in $\mathcal{D}$.

$$b_{n,k} = \frac{q_n \cdot d_k}{|q_n||d_k|}, \quad (9)$$

$$\beta_{n,k} = \frac{\exp(b_{n,k})}{\sum_j^K \exp(b_{n,j})}. \quad (10)$$

Here, we use the softmax function (Eq. (10)) on the cosine similarity between the speaker query and speaker profile (Eq. (9)), which was found to be the most efficient in our preliminary experiment. The attention weight $\beta_{n,k}$ can be seen as a posterior probability of speaker k speaking the n-th token given all previous tokens and speakers as well as X and $\mathcal{D}$.

$$Pr(s_n = k|y_{1:n-1}, s_{1:n-1}, X, \mathcal{D}) \sim \beta_{n,k}. \quad (11)$$

Finally, we calculate the attention-weighted speaker profile $\bar{d}_n$ based on $\beta_{n,k}$ and d_k as followings.

$$\bar{d}_n = \sum_{k=1}^{K} \beta_{n,k} d_k. \quad (12)$$

This weighted profile $\bar{d}_n$ is appended to the input of DecoderOut(). Specifically, we replace Eq. (5) by

$$Pr(y_n|y_{1:n-1}, s_{1:n}, X, \mathcal{D}) \sim \text{DecoderOut}(c_n, u_n, \bar{d}_n)$$
$$= \text{Softmax}(W_{out} \cdot \text{LSTM}(c_n + u_n + W_d \bar{d}_n)), \quad (13)$$

where W_d is a matrix to change the dimension of $\bar{d}_n$ to that of c_n. Terms u_n and c_n are obtained from Eq. (3) and (4), respectively. Note that the output distribution is now conditioned on $s_{1:n}$ and $\mathcal{D}$ because of the addition of the weighted profile $\bar{d}_n$.

3.3. Training

During training, all the network parameters are optimized by maximizing $\log Pr(Y, S|X, \mathcal{D})$ as follows. We call it *speaker-attributed maximum mutual information (SA-MMI)* training.

$$\mathcal{F}^{\text{SA-MMI}} = \log Pr(Y, S|X, \mathcal{D}) \tag{14}$$

$$= \log \prod_{n=1}^{N} \{ Pr(y_n|y_{1:n-1}, s_{1:n}, X, \mathcal{D}) $$
$$\cdot Pr(s_n|y_{1:n-1}, s_{1:n-1}, X, \mathcal{D})^{\gamma} \} \tag{15}$$

$$= \sum_{n=1}^{N} \log Pr(y_n|y_{1:n-1}, s_{1:n}, X, \mathcal{D}) $$
$$+ \gamma \cdot \sum_{n=1}^{N} \log Pr(s_n|y_{1:n-1}, s_{1:n-1}, X, \mathcal{D}). \tag{16}$$

From Eq. (14) to Eq. (15), the chain rule is applied for y_n and s_n alternately. Here, we introduce a scaling parameter γ to adjust the scale of the speaker estimation probability to that of ASR. Equation (16) shows that our training criterion can be factorized into two conditional probabilities defined in Eqs. (13) and (11), respectively. Note that the speaker identity of the token $\langle sc \rangle$ or $\langle eos \rangle$ is set the same as that of the preceding token.

3.4. Decoding

An extended beam search algorithm is used for decoding with the proposed method. In the conventional beam search for AED, each hypothesis contains estimated tokens accompanied by the posterior probability of the hypothesis. In addition to these, a hypothesis for the proposed method contains speaker estimation $\beta_{n,k}$. Each hypothesis expands until $\langle eos \rangle$ is detected, and the estimated tokens in each hypothesis are grouped by $\langle sc \rangle$ to form multiple utterances. For each utterance, the average of $\beta_{n,k}$ values, including the last token corresponding to $\langle sc \rangle$ or $\langle eos \rangle$, is calculated for each speaker. The speaker with the highest average $\beta_{n,k}$ score is selected as the predicted speaker of that utterance. Finally, when the same speaker is predicted for multiple utterances, those utterances are concatenated to form a single utterance.

4. Experiments

4.1. Evaluation settings

4.1.1. Evaluation data

We evaluated the effectiveness of the proposed method by simulating multi-speaker signals based on the LibriSpeech corpus [25]. Following the Kaldi [26] recipe, we used the 960 hours of LibriSpeech training data ("train_960") for model learning, the "dev_clean" set for adjusting hyper-parameter values, and the "test_clean" set for testing.

Our training data were generated as follows. For each utterance in train_960, randomly chosen $(S-1)$ train_960 utterances were added after being shifted by random delays. 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 constraints that (1) the start times of the individual utterances differed by 0.5 sec or longer and that (2) every utterance in each mixed audio sample had at least one speaker-overlapped region with other utterances. For each training sample, speaker profiles were gen-

erated as follows. First, the number of profiles was randomly selected from S to 8. Among those profiles, S profiles were for the speakers involved in the overlapped speech. The utterances for creating the profiles of these speakers were different from those constituting the input overlapped speech. The rest of the profiles were randomly extracted from the other speakers in train_960. Each profile was extracted by using 10 utterances. We generated data for $S = \{1, 2, 3\}$ and combined them to use for training.

The development and evaluation sets were generated from dev_clean or test_clean, respectively, in the same way as the training set except that constraint (1) was not imposed. Therefore, multiple utterances were allowed to start at the same time in evaluation. Also, each profile was extracted from 2 utterances (15 sec on average) instead of 10, unless otherwise stated.

4.1.2. Evaluation metric

We evaluated the model with respect to speaker error rate (SER), WER, and SA-WER. **SER** is defined as the total number of model-generated utterances with speaker misattribution divided by the number of reference utterances. All possible permutations of the hypothesized utterances were examined by ignoring the ASR results, and the one that yielded the smallest number of errors (including the speaker insertion and deletion errors) was picked for the SER calculation. Similarly, **WER** was calculated by picking the best permutation in terms of word errors (i.e., speaker labels were ignored). Finally, **SA-WER** was calculated by comparing the ASR hypothesis and the reference transcription of each speaker.

4.1.3. Model settings

In our experiments, we used a 80-dim log mel filterbank, extracted every 10 msec, for the input feature. We stacked 3 frames of features and applied the model on top of the stacked features. For the speaker profile, we used a 128-dim d-vector [24], whose extractor was separately trained on VoxCeleb Corpus [27, 28]. The d-vector extractor consisted of 17 convolution layers followed by an average pooling layer, which was a modified version of the one presented in [29].

Our AsrEncoder consisted of 5 layers of 1024-dim bidirectional long short-term memory (BLSTM), interleaved with layer normalization [30]. The DecoderRNN consisted of 2 layers of 1024-dim unidirectional LSTM, and the DecoderOut consisted of 1 layer of 1024-dim unidirectional LSTM. We used a conventional location-aware content-based attention [22] with a single attention head. The SpeakerEncoder had the same architecture as the d-vector extractor except for not having the final average pooling layer. Our SpeakerQueryRNN consisted of 1 layer of 512-dim unidirectional LSTM. We used 16k subwords based on a unigram language model [31] as a recognition unit. We apppplied volume perturbation to the mixed audio to increase the training data variability. Note that we applied neither an additional language model (LM) nor any other forms of data augmentation [32, 33, 34, 35] for simplicity.

Model training was performed as follows. In our preliminary experiment, training models from fully random parameters showed poor convergence due to the difficulty in attention module training. Therefore, we initialized the parameters of AsrEncoder, Attention, DecoderRNN, and DecoderOut by SOT-ASR parameters trained on simulated mixtures of LibriSpeech utterances as reported in [19]. We pre-trained the SOT-model with 640k iterations. We also initialized the SpeakerEncoder parameters by using those of the d-vector extractor. After the initial-

Table 1: *SER (%), WER (%), and SA-WER (%) for baseline systems and proposed method. The number of profiles per test audio was 8. Each profile was extracted by using 2 utterances (15 sec on average). For random speaker assignment experiment (3rd row), averages of 10 trials were computed. No LM was used in the evaluation.*

Eval Set / Model	1-speaker			2-speaker-mixed			3-speaker-mixed			Total		
	SER	WER	**SA-WER**	SER	WER	**SA-WER**	SER	WER	**SA-WER**	SER	WER	**SA-WER**
Single-speaker ASR	-	4.7	-	-	66.9	-	-	90.7	-	-	68.4	-
SOT-ASR	-	4.5	-	-	10.3	-	-	19.5	-	-	13.9	-
SOT-ASR + random speaker assignment	87.4	4.5	**175.2**	82.8	23.4	**169.7**	76.1	39.1	**165.1**	80.2	28.1	**168.3**
SOT-ASR + d-vec speaker identification	0.4	4.5	**4.8**	6.4	10.3	**16.5**	13.1	19.5	**31.7**	8.7	13.9	**22.2**
Proposed Model												
SOT-ASR + Spk-Enc + Inv-Attn	0.3	4.3	**4.7**	5.5	10.4	**12.2**	14.8	23.4	**26.7**	9.3	15.9	**18.2**
↳ + SpeakerQueryRNN	0.4	4.2	**4.6**	3.0	9.1	**10.9**	11.6	21.5	**24.7**	6.9	14.5	**16.7**
↳ + Weighted Profile ($\bar{d}_n$)	0.2	4.2	**4.5**	2.5	8.7	**9.9**	10.2	20.2	**23.1**	6.0	13.7	**15.6**

Table 2: *Speaker counting accuracy (%) of the proposed model.*

Actual # of Speakers in Test Data	Estimated # of Speakers (%)			
	1	2	3	>4
1	**99.96**	0.04	0.00	0.00
2	2.56	**97.44**	0.00	0.00
3	0.31	25.34	**74.35**	0.00

Table 3: *SER (%) / SA-WER (%) for different numbers of profiles in the speaker inventory.*

# of Profiles in Inventory	# of Speakers in Test Data			Total
	1	2	3	
4	0.1 / 4.5	1.8 / 9.4	8.8 / 22.3	**5.0 / 15.0**
8	0.2 / 4.5	2.5 / 9.9	10.2 / 23.1	**6.0 / 15.6**
16	0.8 / 5.1	2.9 / 10.6	11.3 / 23.8	**6.8 / 16.3**
32	0.9 / 5.4	4.6 / 11.9	11.6 / 24.0	**7.5 / 16.9**

Table 4: *Impact of the number of profile extraction utterances on SER (%) and SA-WER (%) for 8-profile setting. The average utterance duration was 7.5 sec.*

# of Utterances per Profile	# of Speakers in Test Data			Total
	1	2	3	
1	0.9 / 5.6	3.8 / 11.5	11.2 / 24.8	**7.0 / 17.2**
2	0.2 / 4.5	2.5 / 9.9	10.2 / 23.1	**6.0 / 15.6**
5	0.04 / 4.2	2.1 / 9.5	9.7 / 22.6	**5.6 / 15.2**
10	0.08 / 4.3	2.0 / 9.4	9.5 / 22.3	**5.4 / 15.0**

ization, we updated the entire network based on $\mathcal{F}^{\mathrm{SA-MMI}}$ with $\gamma = 0.1$ by using an Adam optimizer with a learning rate of 0.00002. We used 8 GPUs, each of which worked on 6k frames of minibatch. We report the results of the dev_clean-based best models found after 160k of training iterations.

4.2. Evaluation results

4.2.1. Baseline results

We built 4 different baseline systems, whose results are shown in the first 4 rows of Table 1. The first row corresponds the conventional single-speaker ASR based on AED. As expected, the WER was significantly degraded for overlapped speech. The second row shows the result of the SOT-ASR system that was used for initializing the proposed method in training. SOT-ASR significantly improved the WER for all evaluation settings. The lower WER for the 1-speaker case could be attributed to the data augmentation effect resulting from the use of overlapped speech for training, which was also observed in [19].

The third row shows the result of randomly assigning a speaker label for each utterance generated by SOT-ASR. Note that the speaker identification may affect WER as well as SA-WER. This is because multiple SOT-ASR-generated utterances were mereged when their speaker labels were the same.

The fourth row shows the result of combining SOT-ASR and d-vector based speaker identification. In this baseline system, for each utterance, we calculated a weighted average of frame-level d-vectors by using the attention weights from SOT-ASR. The estimated d-vectors were then compared with each profile contained in the speaker inventory in terms of cosine similarity. The best scored speaker was selected one-by-one with a constraint that the same speaker could not be selected for multiple utterances. This method gave us reasonable results as can be seen in the table although the SA-WERs were not sufficient for overlapped speech.

4.2.2. Results of the proposed method

The last 3 rows of Table 1 shows the results of the proposed method while the first two of them were the results of an ablation study. "SOT-ASR + Spk-Enc + Inv-Attn" is the result of a variant of the proposed model where SpeakerEncoder output p_n was directly used for InventoryAttention (Eq. (9)) instead of using SpeakerQueryRNN output q_n, and $\bar{d}_n$ was not used in Eq. (13). Due to SA-MMI training, even this model achieved a lower SA-WER than the baseline while the SER and WER were degraded. Then, the entire performance was significantly boosted by introducing SpeakerQueryRNN as shown in the next row. Finally, by introducing the weighted profile $\bar{d}_n$ in Eq. (13), the proposed method outperformed the baseline in all three evaluation metrics, resulting in 29% reduction of the SA-WER.

Table 2 shows the speaker counting accuracy of the proposed method. We can see that the speakers were counted very accurately especially for the 1-speaker (99.96%) and 2-speaker cases (97.44%) while it sometimes underestimated the number of the speakers for the 3-speaker mixtures.

4.2.3. Evaluation with different profile settings

In the previous experiments, we used the inventory comprising 8 profiles, each of which was extracted from 2 utterances. We then evaluated the proposed method with different numbers of profiles. As shown in Table 3, our proposed method showed only minor degradation in terms of the SER and SA-WER even with 32 profiles. This demonstrates the robustness of our method against the increase of profiles.

Finally, we also evaluated the impact of the number of utterances used for speaker profile extraction. As shown in Table 4, using more utterances for a profile yielded lower error rates.

5. Conclusions

In this paper, we proposed a joint model for SA-ASR that can recognize overlapped speech of any number of speakers while identifying the speaker of each utterance among any number of speaker profiles. In the experiments on LibriSpeech, the proposed model achieved significantly better SA-WER than the baseline that consists of separated modules.

6. References

[1] J. G. Fiscus, J. Ajot, and J. S. Garofolo, "The rich transcription 2007 meeting recognition evaluation," in *Multimodal Technologies for Perception of Humans*. Springer, 2007, pp. 373–389.

[2] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke *et al.*, "The ICSI meeting corpus," in *Proc. ICASSP*, vol. 1, 2003, pp. I–I.

[3] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, "The AMI meeting corpus: A pre-announcement," in *International workshop on machine learning for multimodal interaction*. Springer, 2005, pp. 28–39.

[4] T. Yoshioka, I. Abramovski, C. Aksoylar, Z. Chen, M. David, D. Dimitriadis, Y. Gong, I. Gurvich, X. Huang, Y. Huang *et al.*, "Advances in online audio-visual meeting transcription," in *Proc. ASRU*, 2019, pp. 276–283.

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

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

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

[8] D. Yu, X. Chang, and Y. Qian, "Recognizing multi-talker speech with permutation invariant training," *Proc. Interspeech 2017*, pp. 2456–2460, 2017.

[9] H. Seki, T. Hori, S. Watanabe, J. Le Roux, and J. R. Hershey, "A purely end-to-end system for multi-speaker speech recognition," in *Proc. ACL*, 2018, pp. 2620–2630.

[10] 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.

[11] X. Chang, W. Zhang, Y. Qian, J. L. Roux, and S. Watanabe, "MIMO-SPEECH: End-to-end multi-channel multi-speaker speech recognition," in *Proc. ASRU*, 2019, pp. 237–244.

[12] N. Kanda, Y. Fujita, S. Horiguchi, R. Ikeshita, K. Nagamatsu, and S. Watanabe, "Acoustic modeling for distant multi-talker speech recognition with single-and multi-channel branches," in *Proc. ICASSP*, 2019, pp. 6630–6634.

[13] N. Kanda, S. Horiguchi, R. Takashima, Y. Fujita, K. Nagamatsu, and S. Watanabe, "Auxiliary interference speaker loss for target-speaker speech recognition," in *Proc. Interspeech*, 2019, pp. 236–240.

[14] P. Wang, Z. Chen, X. Xiao, Z. Meng, T. Yoshioka, T. Zhou, L. Lu, and J. Li, "Speech separation using speaker inventory," in *Proc. ASRU*, 2019, pp. 230–236.

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

[16] K. Kinoshita, M. Delcroix, S. Araki, and T. Nakatani, "Tackling real noisy reverberant meetings with all-neural source separation, counting, and diarization system," *arXiv preprint arXiv:2003.03987*, 2020.

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

[18] N. Kanda, S. Horiguchi, Y. Fujita, Y. Xue, K. Nagamatsu, and S. Watanabe, "Simultaneous speech recognition and speaker diarization for monaural dialogue recordings with target-speaker acoustic models," in *Proc. ASRU*, 2019.

[19] N. Kanda, Y. Gaur, X. Wang, Z. Meng, and T. Yoshioka, "Serialized output training for end-to-end overlapped speech recognition," *arXiv preprint arXiv:2003.12687*, 2020.

[20] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[21] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: First results," in *NIPS Workshop on Deep Learning*, 2014.

[22] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NIPS*, 2015, pp. 577–585.

[23] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016, pp. 4960–4964.

[24] 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.

[25] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.

[26] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *ASRU*, 2011.

[27] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.

[28] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018, pp. 1086–1090.

[29] T. Zhou, Y. Zhao, J. Li, Y. Gong, and J. Wu, "CNN with phonetic attention for text-independent speaker verification," in *Proc. ASRU*, 2019, pp. 718–725.

[30] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[31] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," *arXiv preprint arXiv:1804.10959*, 2018.

[32] N. Kanda, R. Takeda, and Y. Obuchi, "Elastic spectral distortion for low resource speech recognition with deep neural networks," in *Proc. ASRU*, 2013, pp. 309–314.

[33] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech*, 2015, pp. 3586–3589.

[34] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.

[35] C. Wang, Y. Wu, Y. Du, J. Li, S. Liu, L. Lu, S. Ren, G. Ye, S. Zhao, and M. Zhou, "Semantic mask for transformer based end-to-end speech recognition," *arXiv preprint arXiv:1912.03010*, 2019.