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

Serialized Output Training for End-to-End Overlapped Speech Recognition

Naoyuki Kanda, Yashesh Gaur, Xiaofei Wang, Zhong Meng, Takuya Yoshioka

Microsoft Corp.

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

Abstract

This paper proposes serialized output training (SOT), a novel framework for multi-speaker overlapped speech recognition based on an attention-based encoder-decoder approach. Instead of having multiple output layers as with the permutation invariant training (PIT), SOT uses a model with only one output layer that generates the transcriptions of multiple speakers one after another. The attention and decoder modules take care of producing multiple transcriptions from overlapped speech. SOT has two advantages over PIT: (1) no limitation in the maximum number of speakers, and (2) an ability to model the dependencies among outputs for different speakers. We also propose a simple trick that allows SOT to be executed in $O(S)$, where S is the number of the speakers in the training sample, by using the start times of the constituent source utterances. Experimental results on LibriSpeech corpus show that the SOT models can transcribe overlapped speech with variable numbers of speakers significantly better than PIT-based models. We also show that the SOT models can accurately count the number of speakers in the input audio.

Index Terms: multi-speaker speech recognition, attention-based encoder-decoder, permutation invariant training, serialized output training

1. Introduction

Thanks to the advancement in deep neural network (DNN)-based automatic speech recognition (ASR) [1, 2], the word error rate (WER) for single speaker recordings has reached to the level of human transcribers even for tasks that were once thought very challenging (e.g., Switchboard [3, 4], LibriSpeech [5, 6, 7, 8, 9, 10]) . Nonetheless, ASR for multi-speaker speech remains to be a very difficult problem especially when multiple utterances significantly overlap in monaural recordings. For example, an ASR system that achieves a WER of 5.5% for single speaker speech can yield a WER of 84.7% for two-speaker overlapped speech as reported in [11].

Researchers have made tremendous efforts towards the multi-speaker ASR for handling overlapped speech. One of the early works with DNN-based ASR is to train two ASR models, one of which recognizes a speech with higher instantaneous energy and another one of which recognizes a speech with lower instantaneous energy [12]. This method has a limitation that the model can handle only two speakers. A more sophisticated method for multi-speaker ASR is the permutation invariant training (PIT) in which the model has multiple output layers corresponding to different speakers, and the model is trained by considering all possible permutations of speakers. PIT was proposed for speech separation [13] and multi-speaker ASR [14], and worked well for both of them. Despite this, PIT has several limitations. Firstly, the number of the output layers in the model constrains the maximum number of speakers that the model can output. Secondly, it cannot handle the dependency among utterances of multiple speakers because the output layers are independent from each other. Because of this, it is possible that the duplicated hypotheses are generated from different output layers, and extra treatment is necessary to reduce such errors [15]. Thirdly, the computational complexity of PIT is at the order of $O(S^3)$, where S represents the number of speakers. Because of these limitations, most previous works using PIT [13, 14, 15, 16, 17] only addressed the two-speaker case although real recordings often contain even more speakers.

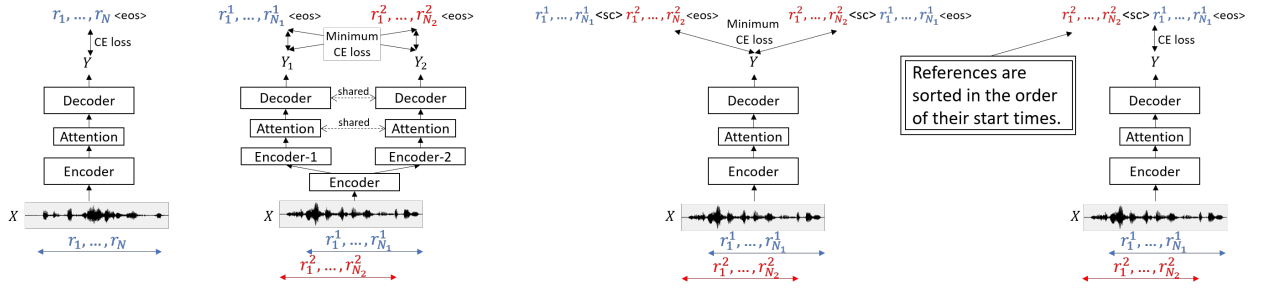
In this paper, we propose a novel framework for overlapped speech recognition, named Serialized Output Training (SOT), based on an attention-based encoder-decoder (AED) approach [18, 19, 20, 21]. Instead of having multiple output layers as with the PIT-based ASR, our proposed model has only one output layer that generates the transcriptions of multiple speakers one after another. By using the single output layer for the modeling, we avoid having the maximum speaker number constraint. In addition, the proposed method can naturally model the dependency among the outputs for multiple speakers, which could help avoid duplicate hypotheses from being generated. We also propose a simple trick that allows SOT to be executed in $O(S)$ by using the start times of the source utterances. We show that the proposed method can better transcribe utterances of multiple speakers from monaural overlapped speech than PIT, and can count the number of speakers with good accuracy.

2. Related Work

The most relevant work we are aware of would be the joint speech recognition and diarization with a recurrent neural network transduer (RNN-T) [22]. In the paper, the authors proposed to generate transcriptions of different speakers interleaved by speaker role tags to recognize two-speaker conversations. Another related piece of work is the AED-based multilingual ASR for mixed-language speech [23, 24]. They used language tags as additional output symbols to transcribe the mixed-language speech with a single model. These methods are similar to ours in that both approaches decode multiple utterances that are separated by a special symbol. However, all the aforementioned methods did not deal with speech overlaps.

It should be noted that, although AED was originally proposed for machine translation to cope with word order differences between the source and target languages [18], the previous studies on the AED-based ASR attempted to incorporate a monotonic alignment constraint to reduce errors in attention estimation. For example, [19] used a penalty for encouraging monotonic alignment, and [25] proposed jointly training connectionist temporal classification (CTC) and AED models. Other popular ASR frameworks, such as the hybrid of DNN and hidden Markov model (HMM), CTC, and RNN-T, also impose the monotonic alignment assumption. By contrast, our attention module scans the encoder embedding sequence back and forth along the time dimension to transcribe utterances of multiple speakers, which is the key difference from the previous studies.

http://dx.doi.org/10.21437/Interspeech.2020-999



(a) Single-speaker ASR (b) Permutation Invariant Training (c) Serialized Output Training (minimum-loss speaker order) (d) Serialized Output Training (first-in, first-out)

Figure 1: *Architectures of (a) the conventional single-speaker ASR, (b) the conventional multi-speaker ASR using PIT, and (c),(d) the proposed serialized output training (SOT) with different schemes. The special symbol $\langle sc \rangle$ represents the speaker change, and is inserted in between the utterances.*

3. Review: Multi-Speaker ASR Based on AED with PIT

3.1. AED-based single-speaker ASR

The AED-based ASR consists of encoder, attention, and decoder modules as shown in Fig. 1 (a). Given input $X = \{x_1, ..., x_T\}$, the AED produces the output sequence $Y = \{y_1, ..., y_n, ...\}$ as follows. Firstly, the encoder converts the input sequence X into a sequence, H^{enc}, of embeddings, i.e.,

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

Then, for every decoder step n, the attention module outputs context vector c_n with attention weight α_n given decoder state vector q_n, the previous attention weight α_{n-1}, and the encoder embeddings H^{enc} as

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

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

$$q_n = \text{DecoderRNN}(y_{n-1}, c_{n-1}, q_{n-1}), \quad (3)$$
$$y_n = \text{DecoderOut}(c_n, q_n). \quad (4)$$

Here, DecoderRNN consists of multiple RNN layers while DecoderOut consists of an affine transform with a softmax output layer. The model is trained to minimize the cross entropy loss given Y and reference label $R = \{r_1, ..., r_N, r_{N+1} = \langle eos \rangle\}$. Specifically, the loss function is defined as

$$\mathcal{L}^{CE} = \sum_{n=1}^{N+1} \text{CE}(y_n, r_n), \quad (5)$$

where CE() represents the cross entropy given the output distribution and the reference label, and N is the number of symbols in the reference R. $\langle eos \rangle$ is the special symbol that represents the end of the sentence.

3.2. PIT-based ASR with multiple output layers

With the conventional multi-speaker ASR with PIT, the model has multiple output branches as shown in Fig. 1 (b). Thus, we have

$$H^{enc_s} = \text{Encoder}_s(H^{enc}) \quad (6)$$
$$c_n^s, \alpha_n^s = \text{Attention}(q_n^s, \alpha_{n-1}^s, H^{enc_s}) \quad (7)$$
$$q_n^s = \text{DecoderRNN}(y_{n-1}^s, c_{n-1}^s, q_{n-1}^s), \quad (8)$$
$$y_n^s = \text{DecoderOut}(c_n^s, q_n^s). \quad (9)$$

Here, s is the index of each output branch, where $1 \leq s \leq S$ with S being the number of speakers. Parameters for the attention and decoder modules are shared across s. Given the set of the outputs, $\{Y^1, \cdots, Y^S\}$, and the set of the references, $\{R^1, \cdots, R^S\}$, where R^s denotes the sth reference defined as $R^s = \{r_1^s, .., r_{N^s}^s, r_{N^s+1}^s = \langle eos \rangle\}$, the PIT-CE loss is calculated by considering all possible speaker permutations as

$$\mathcal{L}^{PIT} = \min_{\phi \in \Phi(1,...,S)} \sum_{s=1}^{S} \sum_{n=1}^{N^s+1} \text{CE}(y_n^{\phi[s]}, r_n^s). \quad (10)$$

Here, $\Phi()$ is the function that generates all possible permutations of a given sequence.

There are three theoretical limitations in PIT. Firstly, the number of the output layers in the model constrains the maximum number of speakers that the model can handle. Secondly, it cannot represent the dependency between the utterances of multiple speakers because each output Y^s has no direct dependency on the other outputs. Because of this, it might be possible that the duplicated hypotheses are generated from each output layer. Thirdly, even with the Hungarian algorithm [26], it requires a training cost of $O(S^3)$ which hinders its application to a large number of speakers.

4. Serialized Output Training

4.1. Overview

Instead of having multiple output layers as with PIT, we propose to use the original form of AED (Eq. (1)-(4)), which has only one output branch, for the multi-speaker ASR. To recognize multiple utterances, we serialize multiple references into a single token sequence. Specifically, we introduce a special symbol $\langle sc \rangle$ to represent the speaker change and simply concatenate the reference labels by inserting $\langle sc \rangle$ between utterances. For example, for a three-speaker case, the reference label will be given as $R = \{r_1^1, .., r_{N^1}^1, \langle sc \rangle, r_1^2, .., r_{N^2}^2, \langle sc \rangle, r_1^3, .., r_{N^3}^3, \langle eos \rangle\}$. Note that $\langle eos \rangle$ is used only at the end of the entire sequence. We call our proposed approach serialized output training (SOT).

Because there are multiple permutations in the order of reference labels to form R, some trick is needed to calculate the loss between the output Y and the concatenated reference label R. For example, in the case of two speakers, the reference label can be either $R = \{r_1^1, .., r_{N^1}^1, \langle sc \rangle, r_1^2, .., r_{N^2}^2, \langle eos \rangle\}$ or $R = \{r_1^2, .., r_{N^2}^2, \langle sc \rangle, r_1^1, .., r_{N^1}^1, \langle eos \rangle\}$. One possible way to determine the order is to calculate the loss for all possible concatenation patterns to form R and select the best one, similarly

to PIT, as (Fig. 1 (c))

$$\mathcal{L}^{SOT-1} = \min_{\phi \in \Phi(1,...,S)} \sum_{n=1}^{N^{sot}} \mathrm{CE}(y_n, r_n^{\phi}), \quad (11)$$

where $N^{sot} = \sum_{s=1}^{S}\{N^s + 1\}$, and r_n^{ϕ} is the n-th token in the concatenated reference given permutation ϕ. We call this method as SOT with minimum-loss speaker order. This method has the problem that requires $O(S!)$ training cost.

To reduce the computational cost to $O(S)$, we alternatively propose to sort the reference labels by their start times (Fig. 1 (d)) as follows:

$$\mathcal{L}^{SOT-2} = \sum_{n=1}^{N^{sot}} \mathrm{CE}(y_n, r_n^{\Psi(1,...,S)}). \quad (12)$$

Here, Ψ is the function that outputs the sorted index of $\{1, ..., S\}$ according to the start time of each speaker. The term $r_n^{\Psi(1,...,S)}$ is the n-th token in the concatenated reference given $\Psi(1, ..., S)$. As a result, the AED is trained to recognize the utterances of multiple speakers in the order of their start times, separated by a special symbol $\langle sc \rangle$. We call this method as SOT based on first-in, first-out order[1]. The only assumption to perform this first-in, first-out training is that the two utterances do not start at exactly the same time. If that is the case, we randomly determine the utterance order. That said, it rarely happens in real recordings, and thus its impact should be marginal.

4.2. Separation after attention (SAA)

With PIT, speech separation is explicitly modeled by the multiple encoder branches. In the proposed SOT framework, the attention module operates on the encoder embeddings that could be contaminated by overlapped speech. Thus, the context vector generated by the attention module may also be contaminated, resulting in potential degracation in accuracy.

We found that simply inserting one unidirectional LSTM layer in DecoderOut() in Eq (4) could solve the problem effectively. Unlike PIT, where the speech separation is performed *before* the attention module, this additional LSTM works as a separation module, taking place *after* the attention module. In our experiment, we removed one encoder layer when we applied this "Separation after Attention (SAA)" method for the sake of fairness in terms of the model size.

4.3. Advantages of the proposed method

There are two key advantages of using single output branch instead of multiple branches as with PIT. Firstly, there is no longer a limitation on the maximum number of speakers that the model can handle. Secondly, the proposed model can represent the dependency among multiple utterances, which prevents duplicated hypotheses from being generated.

Furthermore, the proposed model can even predict the number of speakers if the model is trained on a data set including various numbers of speakers. At the inference time, the decoder module will not stop until $\langle eos \rangle$ is predicted. Therefore, it can automatically count the number of speakers in the recording just by counting the occurrences of $\langle sc \rangle$ and $\langle eos \rangle$.

[1] A similar idea to use the start times was proposed in [27] during this paper was reviewed.

Table 1: *WER(%) of 512-dim models for* **2-speaker-mixed speech**. *Note that the WERs for single speaker speech by the single-speaker ASR were 5.4% and 5.7% for dev_clean and test_clean, respectively.*

Model (all 512-dim)	WER (%) dev_clean	test_clean
Single-speaker ASR	67.9	68.5
SOT (minimum loss speaker order)	17.4	17.1
SOT (first-in, first-out)	17.0	16.5

5. Experiments

5.1. Evaluation settings

5.1.1. Training and evaluation data

In this work, we used the LibriSpeech corpus [28] to simulate multi-speaker signals and evaluate the proposed method. LibriSpeech consists of about 1,000 hours of audio book data. We followed the Kaldi [29] recipe to generate the dataset, and used the 960 hours of training data ("train_960") for training the ASR models, and used "dev_clean" and "test_clean" for the evaluation.

Our training data were generated as follows. For each training example, we firstly determined the number of speakers S to be included in the sample. For each utterance in "train_960", $(S-1)$ utterances were randomly picked up from "train_960" and added with 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 differ by 0.5 s or longer and that (2) every utterance in each mixed audio sample has at least one speaker-overlapped region with other utterances.

The evaluation set was generated from "dev_clean" or "test_clean" in the same way except that Constraint (1) mentioned above was not imposed. Therefore, multiple utterances were allowed to start at the same time in the evaluation data.

5.1.2. Evaluation metric

Our trained models were evaluated with respect to WER. In multi-talker ASR, a system may produce a different number of hypotheses than references (i.e., speakers). To cope with this, all possible permutations of the hypothesis order were examined, and the one that yielded the lowest WER was picked.

5.1.3. Model settings

In our experiments, we used 6 layers of M-dim bidirectional long short-term memory (BLSTM) for the encoder, where M was set to 512, 724 or 1024. Layer normalization [30] was applied after every BLSTM. For PIT-based baseline systems, the first 5 encoder layers were shared across the output branches, and each branch had its own last encoder layer. The decoder module consists of 2 layers of M-dim unidirectional LSTM without layer normalization. We used a conventional location-aware content-based attention [20] with a single head.

As for the input feature, we used an 80-dim log mel filterbank extracted every 10 msec. We stacked 3 frames of features and applied the encoder on top of the stacked features. We used 16k subwords based on a unigram language model [31] as a recognition unit. We applied the speed and volume perturbation [32] to the mixed audio to enhance the model training.

Table 2: *WER (%) of 512-dim models for* **various mixtures** *of training and test data. Test data were generated by mixing "test_clean". Evaluation results with unmatched training/testing conditions were shown with parenthesis.*

Model (all 512-dim)	# of Speakers in Training Data	# of Speakers in Test Data 1	2	3
Single-spk ASR	1	5.7	(68.5)	(92.7)
SOT (fifo)	2	(16.7)	16.5	(55.2)
SOT (fifo)	1,2	5.7	18.6	(59.7)
SOT (fifo)	1,2,3	5.4	17.3	34.3

Table 3: *WER (%) for different numbers of parameters and architectures for SOT model trained with the mixture of 1, 2, and 3 speakers. Test data were generated by mixing "test_clean."*

Model Dim	SAA (Sec 4.2)	# of Params	# of Speakers in Test Data 1	2	3	**Total**
512		44.7M	5.4	17.3	34.3	**23.8**
724		79.2M	4.5	12.0	26.5	**18.0**
1024		143.9M	4.8	10.9	25.8	**17.3**
1024	$\surd$	135.6M	4.6	11.2	24.0	**16.5**

We used the Adam optimizer with a learning rate schedule similar to that described in [9]. We firstly linearly increased the learning rate from 0 to 0.0002 by using the initial 1k iterations and kept the learning rate until the 160k-th iteration. We then started decaying the learning rate exponentially at a rate of 1/10 per 240k iterations. In this paper, we report the results of the"dev_clean"-based best models found after 320k of training iterations. We used minibatch consists of 9k, 7.5k, and 6k frames of input for M=512, 724, 1024, respectively. 8 GPUs were used for all training.

Note that we applied neither an additional language model nor SpecAugment [9] for simplicity. Our results for the standard LibriSpeech "test_clean" (4.6% of WER in Table 4) was on par with recently reported results without these techniques (eg, 4.1%–4.6% of WER was reported in [8, 9, 33, 34, 35]).

5.2. Evaluation results

5.2.1. Evaluation on two-speaker mixed speech

First, we evaluated the SOT model for two-speaker overlapped speech. In this experiment, both the training and evaluation data consisted only of two-speaker overlapped utterances. We used a 512-dim AED model without SAA.

As shown in Table 1, the SOT model significantly outperformed a normal single-speaker ASR. Surprisingly, the "first-in, first-out" training achieved a better WER than the "minimum loss speaker order" training although the former approach can be executed with a computational cost of $O(S)$ with respect to the number of speakers. Based on this finding, we used the "first-in, first-out" training for the rest of the experiments.

5.2.2. Evaluation on variable numbers of overlapped speakers

We then evaluated the same 512-dim model with a training set consisting of various numbers of speakers. The results are shown in Table 2, where the SOT model is shown to be able to recognize overlapped speech with variable numbers of speakers very well. It should be emphasized that the SOT model did not show any degradation for the single-speaker case, sometimes even outperformed the single-speaker ASR model. This might be because of the data augmentation effect resulting from mixing training utterances.

Table 4: *WER (%) comparison of PIT and SOT. Test data were generated by mixing "test_clean". Evaluation results with unmatched training/testing conditions were shown with parenthesis. Note that we trained the PIT model with up to 2 speakers since 3-output PIT required impractical training time.*

Model (1024-dim)	# of Speakers in Training Data	# of Params.	# of Speakers in Test Data 1	2	3
2-output PIT	2	160.7M	(80.6)	11.1	(52.1)
2-output PIT	1,2	160.7M	6.7	11.9	(52.3)
SOT	1,2,3	135.6M	4.6	11.2	24.0

Table 5: *Speaker counting accuracy (%) for the 1024-dim SOT model trained by the mixture of 1, 2, 3 speakers.*

Actual # of Speakers in Test Data	Estimated # of Speakers (%) 1	2	3	>4
1	**99.8**	0.2	0.0	0.0
2	1.9	**97.0**	1.1	0.0
3	0.2	24.0	**74.2**	1.5

We also investigated the impact of the number of parameters, the results of which are shown in Table 3. We found that the large model size was essential for SOT to achieve good results. This is because the attention and decoder modules in the SOT framework are required to work on contaminated embeddings as we discussed in Sec 4.2. Applying SAA further improved the performance as shown in the last row of Table 3. Note that the SOT model size of with SAA is smaller than the naive SOT because we removed one 1024-dim *bidirectional* LSTM layer from the encoder instead of simply inserting one 1024-dim *unidirectional* LSTM after the attention module.

5.2.3. Comparison with PIT

Table 4 compares SOT with PIT. In this experiment, we used 1024-dim model for both PIT and SOT. As shown in the table, PIT showed severe WER degradation for the 1-speaker case even when the training data contained many 1-speaker examples. SOT achieved significantly better results than the PIT model trained on 1- and 2-speaker mixtures for both 1- and 2-speaker evaluation data while recognizing 3-speaker evaluation data, with fewer parameters. For reference, the training speed for the 3-speaker SOT was roughly 30% faster (including I/O) than the 2-output PIT. The difference of the training speed would become much larger if we use PIT with more output branches.

5.2.4. Speaker counting accuracy

Finally, we analyzed the speaker counting accuracy of the 1024-dim SOT. As shown in Table 5, we found that the model could count the number of speakers very accurately especially for 1-speaker (99.8%) and 2-speaker cases (97.0%) while it sometimes underestimated the speakers for the 3-speaker mixtures.

6. Conclusions

In this paper, we proposed SOT that can recognize overlapped speech consisting of any number of speakers. We also proposed a simple trick to execute SOT in $O(S)$ by using the start time of each utterance. Our experiments on LibriSpeech showed that the proposed model could transcribe utterances from monaural overlapped speech significantly more effectively than PIT while being able to accurately count the number of speakers as well.

7. References

[1] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech*, 2011, pp. 437–440.

[2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[3] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," *arXiv preprint arXiv:1610.05256*, 2016.

[4] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim *et al.*, "English conversational telephone speech recognition by humans and machines," in *Proc. Interspeech*, 2017, pp. 132–136.

[5] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *Proc. ICML*, 2016, pp. 173–182.

[6] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Proc. Interspeech*, 2016, pp. 2751–2755.

[7] N. Kanda, Y. Fujita, and K. Nagamatsu, "Lattice-free state-level minimum Bayes risk training of acoustic models." in *Proc. Interspeech*, vol. 2018, 2018, pp. 2923–2927.

[8] C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, "RWTH ASR systems for LibriSpeech: Hybrid vs attention," in *Proc. Interspeech*, 2019, pp. 231–235.

[9] 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.

[10] K. J. Han, R. Prieto, K. Wu, and T. Ma, "State-of-the-art speech recognition using multi-stream self-attention with dilated 1d convolutions," *arXiv preprint arXiv:1910.00716*, 2019.

[11] 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.

[12] C. Weng, D. Yu, M. L. Seltzer, and J. Droppo, "Deep neural networks for single-channel multi-talker speech recognition," *IEEE/ACM Trans. on TASLP*, vol. 23, no. 10, pp. 1670–1679, 2015.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

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

[23] S. Watanabe, T. Hori, and J. R. Hershey, "Language independent end-to-end architecture for joint language identification and speech recognition," in *Proc. ASRU*, 2017, pp. 265–271.

[24] H. Seki, S. Watanabe, T. Hori, J. Le Roux, and J. R. Hershey, "An end-to-end language-tracking speech recognizer for mixed-language speech," in *Proc. ICASSP*, 2018, pp. 4919–4923.

[25] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. ICASSP*, 2017, pp. 4835–4839.

[26] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

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

[28] 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.

[29] 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.

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

[33] K. Irie, A. Zeyer, R. Schlüter, and H. Ney, "Language modeling with deep transformers," in *Proc. Interspeech*, 2019, pp. 3905–3909.

[34] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney, "A comparison of transformer and LSTM encoder decoder models for ASR," in *Proc. ASRU*, 2019.

[35] A. Rosenberg, B. Ramabhadran, P. Moreno, Y. Jia, Y. Wu, Y. Zhang, and Z. Wu, "Speech recognition with augmented synthesized speech," 2019, pp. 996–1002.