END-TO-END MULTI-SPEAKER SPEECH RECOGNITION WITH TRANSFORMER

Xuankai Chang[1], *Wangyou Zhang*[2], *Yanmin Qian*[2], *Jonathan Le Roux*[3], *Shinji Watanabe*[1]

[1]Center for Language and Speech Processing, Johns Hopkins University, USA
[2]MoE Key Lab of Artificial Intelligence &
SpeechLab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
[3]Mitsubishi Electric Research Laboratories (MERL), USA

xchang14@jhu.edu, {wyz-97, yanminqian}@sjtu.edu.cn, leroux@merl.com, shinjiw@jhu.edu

ABSTRACT

Recently, fully recurrent neural network (RNN) based end-to-end models have been proven to be effective for multi-speaker speech recognition in both the single-channel and multi-channel scenarios. In this work, we explore the use of Transformer models for these tasks by focusing on two aspects. First, we replace the RNN-based encoder-decoder in the speech recognition model with a Transformer architecture. Second, in order to use the Transformer in the masking network of the neural beamformer in the multi-channel case, we modify the self-attention component to be restricted to a segment rather than the whole sequence in order to reduce computation. Besides the model architecture improvements, we also incorporate an external dereverberation preprocessing, the weighted prediction error (WPE), enabling our model to handle reverberated signals. Experiments on the spatialized wsj1-2mix corpus show that the Transformer-based models achieve 40.9% and 25.6% relative WER reduction, down to 12.1% and 6.4% WER, under the anechoic condition in single-channel and multi-channel tasks, respectively, while in the reverberant case, our methods achieve 41.5% and 13.8% relative WER reduction, down to 16.5% and 15.2% WER.

Index Terms— Transformer, end-to-end, overlapped speech recognition, neural beamforming, speech separation.

1. INTRODUCTION

Deep learning techniques have dramatically improved the performance of separation and automatic speech recognition (ASR) tasks related to the cocktail party problem [1], where the speech from multiple speakers overlaps. Two main scenarios are typically considered, single-channel and multi-channel. In single-channel speech separation, various methods have been proposed, among which deep clustering (DPCL) based methods [2] and permutation invariant training (PIT) based methods [3] are the dominant ones. For ASR, methods combining separation with single-speaker ASR as well as methods skipping the explicit separation step and building directly a multi-speaker speech recognition system have been proposed, using either the hybrid ASR framework [4–6] or the end-to-end ASR framework [7–9]. In the multi-channel condition, the spatial information derived from the inter-channel differences can help distinguish between speech sources from different directions, which makes the problem easier to solve. Several methods have been proposed for multi-channel speech separation, including DPCL-based

methods using integrated beamforming [10] or inter-channel spatial features [11], and a PIT-based method using a multi-speaker mask-based beamformer [12]. For multi-channel multi-speaker speech recognition, an end-to-end system was proposed in [13], called MIMO-Speech because of the multi-channel input (MI) and multi-speaker output (MO). This system consists of a mask-based neural beamformer frontend, which explicitly separates the multi-speaker speech via beamforming, and an end-to-end speech recognition model backend based on the joint CTC/attention-based encoder-decoder [14] to recognize the separated speech streams. This end-to-end architecture is optimized via only the connectionist temporal classification (CTC) and cross-entropy (CE) losses in the backend ASR, but is nonetheless able to learn to develop relatively good separation abilities.

Recently, Transformer models [15] have shown impressive performance in many tasks, such as pretrained language models [16,17], end-to-end speech recognition [18,19], and speaker diarization [20], surpassing the long short-term memory recurrent neural networks (LSTM-RNNs) based models. One of the key components in the Transformer model is self-attention, which computes the contribution information of the whole input sequence and maps the sequence into a vector at every time step. Even though the Transformer model is powerful, it is usually not computationally practical when the sequence length is very long. It also needs adaptation for specific tasks, such as the subsampling operation in encoder-decoder based end-to-end speech recognition. However, for signal-level processing tasks such as speech separation and enhancement, subsampling is usually not a good option, because these tasks need to maintain the original time resolution.

In this paper, we explore the use of Transformer models for end-to-end multi-speaker speech recognition in both the single-channel and multi-channel scenarios. First, we replace the LSTMs in the encoder-decoder network of the speech recognition module with Transformers for both scenarios. Second, in order to also apply Transformers in the masking network of the neural beamforming module in the multi-channel case, we modify the self-attention layers to reduce their memory consumption in a time-restricted (or local) manner, as used in [5, 21, 22]. To the best of our knowledge, this work is the first attempt to use the Transformer model for tasks such as speech enhancement/separation with such very long sequences. Another contribution of this paper is to improve the robustness of our model in reverberant environments. To do so, we incorporate an external dereverberation method, the weighed prediction error (WPE) [23], to preprocess the reverberated speech. The experiments show that this straightforward method can lead to a performance boost for reverberant speech.
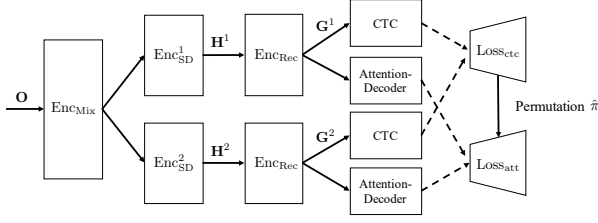
Fig. 1. End-to-end single-channel multi-speaker model in the 2-speaker case. The speaker-differentiating encoder (Enc$_{SD}$), recognition encoder (Enc$_{Rec}$), and decoder are either RNNs or Transformers.
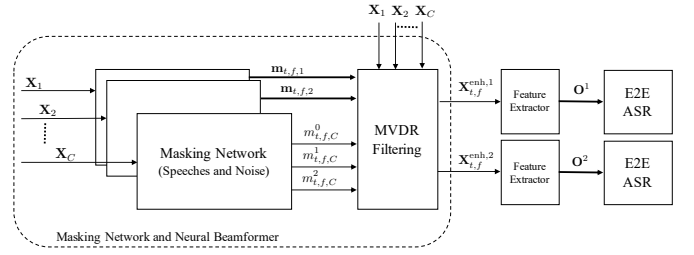

Fig. 2. End-to-end multi-channel multi-speaker model in the 2-speaker case. The masking network and end-to-end ASR network are based on either RNNs or Transformers.

2. END-TO-END MULTI-SPEAKER ASR

In this section, we review the end-to-end speech recognition models for both the single-channel [8, 9] and multi-channel [13] tasks. For both tasks, we denote by J the number of speakers in the input speech mixture.

2.1. Single-channel Multi-speaker ASR

In this subsection, we briefly introduce the end-to-end single-channel multi-speaker speech recognition model proposed in [8, 9], shown in Fig.1. The model is an extension of the joint CTC/attention-based encoder-decoder framework [14] to recognize multi-speaker speech. The input $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$ is the single-channel mixed speech feature. In the encoder, the input feature is separated and encoded as hidden states $\mathbf{G}^j, j = \{1, \ldots, J\}$ for each speaker. The computation of the encoder can be divided into three submodules:

$$\mathbf{H} = \text{Encoder}_{\text{Mix}}(\mathbf{O}), \tag{1}$$

$$\mathbf{H}^j = \text{Encoder}_{\text{SD}}^j(\mathbf{H}), \ j = 1, \ldots, J, \tag{2}$$

$$\mathbf{G}^j = \text{Encoder}_{\text{Rec}}(\mathbf{H}^j), \ j = 1, \ldots, J. \tag{3}$$

Encoder$_{\text{Mix}}$ first maps the input $\mathbf{O}$ to some high dimensional representation $\mathbf{H}$. Then J speaker-differentiating encoders Encoder$_{\text{SD}}^j$ extract each speaker's speech $\mathbf{H}^j$. Finally, Encoder$_{\text{Rec}}$ transforms each $\mathbf{H}^j$ into the embeddings $\mathbf{G}^j = \{\mathbf{g}_1^j, \ldots, \mathbf{g}_L^j\}, L \leq T$ with subsampling. The attention-based decoder then takes these hidden representations to generate the corresponding output token sequences $\mathbf{Y}^j = \{y_1^j, \ldots, y_N^j\}$. For each embedding sequence $\mathbf{G}^j$, the recognition process is formalized as follows:

$$\mathbf{c}_n^j = \text{Attention}(\mathbf{e}_{n-1}^j, \mathbf{G}^j), \tag{4}$$

$$\mathbf{e}_n^j = \text{Update}(\mathbf{e}_{n-1}^j, \mathbf{c}_{n-1}^j, y_{n-1}^j), \tag{5}$$

$$y_n^j \sim \text{Decoder}(\mathbf{e}_n^j, y_{n-1}^j), \tag{6}$$

in which $\mathbf{c}_n^j$ denotes the context vector and $\mathbf{e}_n^j$ is the hidden state of the decoder at step n. To determine the permutation of the reference sequences $\mathbf{R}^j$, permutation invariant training (PIT) is performed on the CTC loss right after the encoder [8, 9]:

$$\hat{\pi} = \underset{\pi \in \mathcal{P}}{\text{argmin}} \sum_j \text{Loss}_{\text{ctc}}(\mathbf{Z}^j, \mathbf{R}^{\pi(j)}), \ j = 1, \ldots, J, \tag{7}$$

where $\mathbf{Z}^j$ is the sequence obtained from $\mathbf{G}^j$ by linear transform to compute the label posterior distribution, $\mathcal{P}$ is the set of all permutations on $\{1, \ldots, J\}$, and $\pi(i)$ is the i-th element of permutation π. The model is optimized with both CTC and cross-entropy losses:

$$\mathcal{L} = \sum_j \left(\lambda \text{Loss}_{\text{ctc}}(\mathbf{Z}^j, \mathbf{R}^{\hat{\pi}(j)}) + (1-\lambda) \text{Loss}_{\text{att}}(\mathbf{Y}^j, \mathbf{R}^{\hat{\pi}(j)}) \right), \tag{8}$$

where $0 \leq \lambda \leq 1$ is an interpolation factor, and Loss$_{\text{att}}$ is the cross-entropy loss of the attention-decoder.

2.2. Multi-channel Multi-speaker ASR

In this subsection, we review the model architecture of the MIMO-Speech end-to-end multi-channel multi-speaker speech recognition system [13], shown in Fig.2. The model takes as input the microphone-array signals from an arbitrary number C of sensors. The model can be roughly divided into two modules, namely the frontend and the backend. The frontend is a mask-based multi-source neural beamformer. For simplicity of notation, we denote the noise as the 0-th source in the mixture signals. First, the monaural masking network estimates the masks $\mathbf{M}_c^j$ for every source $j = 0, 1, \ldots, J$ on each channel $c = 1, \ldots, C$ from the complex STFT of the multi-channel mixture speech, $\mathbf{X}_c = (x_{t,f,c})_{t,f} \in \mathbb{C}^{T \times F}$, where $1 \leq t \leq T$ and $1 \leq f \leq F$ represent the time and frequency indices, as follows:

$$\mathbf{M}_c = \text{MaskNet}(\mathbf{X}_c), \tag{9}$$

where $\mathbf{M}_c = (m_{t,f,c}^j)_{t,f,j} \in [0,1]^{T \times F \times (J+1)}$. Second, the multi-source neural beamformer separates each source from the mixture based on the MVDR formalization [24]. The estimated masks of each source are used to compute the corresponding power spectral density (PSD) matrices $\mathbf{\Phi}^j$ for $j \in \{0, \ldots, J\}$ [25–27]:

$$\mathbf{\Phi}^j(f) = \frac{1}{\sum_{t=1}^T m_{t,f}^j} \sum_{t=1}^T m_{t,f}^j \mathbf{x}_{t,f} \mathbf{x}_{t,f}^H \in \mathbb{C}^{C \times C}, \tag{10}$$

where $\mathbf{x}_{t,f} = (x_{t,f,c})_c \in \mathbb{C}^C$, $m_{t,f}^j = \frac{1}{C} \sum_{c=1}^C m_{t,f,c}^j$ and ^{H} represents the conjugate transpose. The time-invariant filter coefficients $\mathbf{g}^j(f)$ for each speaker j are then computed from the PSD matrices:

$$\mathbf{g}^j(f) = \frac{(\sum_{i \neq j} \mathbf{\Phi}^i(f))^{-1} \mathbf{\Phi}^j(f)}{\text{Tr}((\sum_{i \neq j} \mathbf{\Phi}^i(f))^{-1} \mathbf{\Phi}^j(f))} \mathbf{u} \in \mathbb{C}^C, \tag{11}$$

where $1 \leq j \leq J$, and $\mathbf{u} \in \mathbb{R}^C$ is a vector representing the reference microphone derived from an attention mechanism [28]. The beamforming filters $\mathbf{g}^j$ can be used to obtain the enhanced signal $\hat{s}_{t,f}^j$ for speaker j, which is further processed to get the log mel-filterbank with global mean and variance normalization (LMF($\cdot$)):

$$\hat{s}_{t,f}^j = (\mathbf{g}^j(f))^H \mathbf{x}_{t,f} \in \mathbb{C}, \tag{12}$$

$$\mathbf{O}^j = \text{LMF}(|\hat{\mathbf{S}}^j|), \tag{13}$$

where $\hat{\mathbf{S}}^j$ is the short-time Fourier transform (STFT) of $\hat{s}^j$.

The backend ASR module maps the speech feature $\mathbf{O}^j = \{\mathbf{o}_1^j, \ldots, \mathbf{o}_T^j\}$ of each speaker j into the output token sequences $\mathbf{Y}^j = \{y_1^j, \ldots, y_N^j\}$. The computation of the speech recognition is very similar to the process for the single-channel case described in Sec. 2.1, except that the encoder is a single path network and does not have to separate the input feature using Encoder$_{\text{SD}}$.

Similar to the single-channel model, the permutation order of the reference sequences $\mathbf{R}^j$ is determined by (7). The whole MIMO-Speech model is optimized only with ASR loss as in (8).

3. TRANSFORMER WITH TIME-RESTRICTED SELF-ATTENTION

In this section, we describe one of the key components in the Transformer architecture, the multi-head self-attention [15], and the time-restricted modification [22] for its application in the masking network of the frontend.

Transformers employ the dot-product self-attention for mapping a variable-length input sequence to another sequence of the same length, making them different from RNNs. The input consists of queries Q, keys K, and values V of dimension d^{att}. The weights of the self-attention are obtained by computing the dot-product between the query and all keys and normalizing with softmax. A scaling factor $\sqrt{d^{att}}$ is used to smooth the distribution:

$$\text{Attention}(Q, K, V) = \text{softmax}\Big(\frac{QK^T}{\sqrt{d^{att}}}\Big)V. \quad (14)$$

To capture information from different representation subspaces, multi-head attention (MHA) is used by multiplying the original queries, keys, and values by different weight matrices:

$$\text{MHA}(Q, K, V) = \text{Concat}([H_h]_{h=1}^{d^{head}})W^{head}, \quad (15)$$

$$\text{where } H_h = \text{Attention}(QW_h^q, KW_h^k, V_h^v W_h^v), \quad (16)$$

where d^{head} is the number of heads, and $W^{head} \in \mathbb{R}^{(d^{head} d^{att}) \times d^{att}}$ and $W_h^q, W_h^k, W_h^v \in \mathbb{R}^{d^{att} \times d^{att}}$ are learnable parameters.

In general, the speech sequence length can be considerably long, making self-attention computationally difficult. For tasks like speech separation and enhancement, the technique of subsampling is not practical as in speech recognition. Inspired by [21, 22], we adjust the self-attention of the Transformers in the masking network to be performed on a local segment of the speech, because those frames have higher correlation. This time-restricted self-attention for the query at time step t is formalized as:

$$\text{Attention}(Q, K', V') = \text{softmax}\Big(\frac{QK'^T}{\sqrt{d^{att}}}\Big)V', \quad (17)$$

where the corresponding keys and values are $K' = K_{t-l:t+r}$ and $V' = V_{t-l:t+r}$, respectively, with l and r here denoting the left and right context window sizes.

4. EXPERIMENTS

The proposed methods were evaluated on the same dataset as in [13], referred to as the spatialized wsj1-2mix dataset, where the number of speakers in an utterance is $J = 2$. The multi-channel speech signals were generated[1] from the monaural wsj1-2mix speech used in [8,9]. The room impulse responses (RIR) for the spatialization were randomly generated[2], characterizing the room dimensions, speaker locations, and microphone geometry. The final spatialized dataset contains two different environment conditions, anechoic and reverberant. In the anechoic condition, the room is assumed to be anechoic

---

[1]The spatialization toolkit is available at http://www.merl.com/demos/deep-clustering/spatialize_wsj0-mix.zip

[2]The RIR generator script is available online at https://github.com/ehabets/RIR-Generator

and only the delays and decays due to the propagation are considered when generating the signals. In the reverberant condition, reverberation is also considered, with randomly drawn T60s from $[0.2, 0.6]$ s. In total, the spatialized corpus under each condition contains 98.5 hr, 1.3 hr, and 0.8 hr in training, development, and evaluation sets respectively.

In the single-channel multi-speaker speech recognition task, we used the 1st channel of the training, development, and evaluation set to train, validate, and evaluate our model respectively. The input features are 80-dimensional log mel-filterbank coefficients with pitch features and their delta and delta delta features. In the multi-channel multi-speaker speech recognition task, we also followed [13] in including the WSJ train_si284 in the training set to improve the performance. The model takes the raw waveform audio signal as input and converts it to its STFT using a 25 ms-long Hann window with stride 10 ms. The spectral feature dimension is $F = 257$ due to zero-padding. After the frontend computation, 80-dimensional log filterbank features are extracted for each separated speech signal and global mean-variance normalization is applied, using the statistics of the single-speaker WSJ1 training set. All the multi-channel experiments were performed with $C = 2$ channels. However, the model can be extended to an arbitrary number of input channels as described in [28].

4.1. Experimental Setup

All the proposed end-to-end multi-speaker speech recognition models are implemented with the ESPnet framework [29] using the Pytorch backend. Some basic parts are the same for all the models. The interpolation factor λ of the loss function in (8) is set to 0.2. The word-level language model [30] used during decoding was trained with the official text data included in the WSJ corpus. The configurations of the RNN-based models are the same as in [9] and [13] for single-channel and multi-channel experiments, respectively.

In the Transformer-based multi-speaker encoder-decoder ASR model, there is a total of 12 layers in the encoder and 6 layers in the decoder as in [18]. Before the Transformer encoder, the log mel-filterbank features are encoded by two CNN blocks. The CNN layers have a kernel size of 3×3 and the number of feature maps is 64 in the first block and 128 in the second block. For the single-channel multi-speaker model in Sec. 2.1, Encoder$_{\text{Mix}}$ is the same as the CNN embedding layer, and Encoder$_{\text{SD}}$ and Encoder$_{\text{Rec}}$ contain 4 and 8 Transformer layers, respectively. For all the tasks, the configuration of each encoder-decoder layer is $d^{att} = 256$, $d^{ff} = 2048$, $d^{head} = 4$. The masking network in the frontend has 3 layers similar to the encoder-decoder layer except $d^{ff} = 768$. The training stage of Transformer runs with the Adam optimizer and Noam learning rate decay as in [15]. Note that the backend ASR module is currently initialized with a pretrained model from the ESPnet recipe of WSJ corpus and kept frozen for the first 15 epochs, for training stability.

4.2. Performance in Anechoic Condition

We first provide in Table 1 the performance in anechoic condition of the single-channel multi-speaker end-to-end ASR models trained and evaluated on the original single-channel wsj1-2mix corpus used in [9, 30]. All the layers are randomly initialized. The result shows that using the Transformer model leads to a 40.9% relative word error rate (WER) improvement on the evaluation set, decreasing from 20.43% to 12.08% compared with the RNN-based model in [9].

The multi-channel multi-speaker speech recognition performance is shown in Table 2 using the spatialized anechoic wsj1-2mix dataset. The baseline multi-channel system is the RNN-based

Table 1. Performance in terms of average WER [%] on the **single-channel anechoic** wsj1-2mix corpus.

Model	dev	eval
RNN-based 1-channel Model [9]	24.90	20.43
Transformer-based 1-channel Model	**17.11**	**12.08**

Table 2. Performance in terms of average WER [%] on the spatialized **two-channel anechoic** wsj1-2mix corpus.

Model	dev	eval
RNN-based MIMO-Speech [13]	13.54	8.62
+ Transformer backend	**10.73**	6.85
++ Transformer frontend	11.75	**6.41**

model from our previous study [13]. Before we move to the fully Transformer-based MIMO-Speech model, we first replace the RNNs with Transformers in the backend ASR only. We see that using Transformers for the ASR backend can achieve 20.5% relative improvement against the RNN-based model in anechoic conditions.

We then also apply Transformers in the masking network of the frontend. Considering the feasibility of computing, in this preliminary study, the left and right context window sizes of the self-attention are set to $l = 14$ and $r = 15$. The parameters of the frontend are randomly initialized. Compared with using a Transformer-based model only for the backend, the fully Transformer-based model leads to a further improvement, achieving a WER of 6.41%. Compared against the whole sequence information available in the RNN-based model, such a small context window greatly limits the power of our model but shows its potential. Overall, the proposed fully Transformer-based model achieves a 25.6% relative WER improvement against the RNN-based model in the multi-channel case. We also see that the multi-channel system is better than the single-channel system, thanks to the availability of spatial information.

4.3. Performance in Reverberant Condition

Even though our model can perform very well in anechoic condition, such ideal environments are rarely encountered in practice. It is thus crucial to investigate whether the model can be applied in more realistic environments. In this subsection, we describe preliminary efforts to process the reverberated signal.

We first used a straightforward multi-conditioned training by adding reverberated utterances into the training set. The results of multi-speaker speech recognition on the multi-channel reverberant datasets are shown in Table 3. It can be observed that only using the Transformers for the backend is 6.6% better than the RNN-based model. In addition, the fully Transformer-based model achieves 13.2% relative WER improvement on the evaluation set, which is consistent with the anechoic case. However, comparing with the numbers for the anechoic condition in Table 2, a large performance degradation can be observed.

To alleviate this, we turned to an existing external dereverberation method to preprocess the input signals as a simple yet effective solution. Nara-WPE [31] is a widely used open source software for blind dereverberation of acoustic signals. The dereverberation is performed on the reverberated speech before it is added to the training dataset with anechoic data. Similarly, the reverberant test set is also preprocessed. Speech recognition performance on the multi-channel reverberant speech after Nara-WPE is shown in Table 4. In general, the WERs are dramatically decreased with the dereverbera-

Table 3. Performance in terms of average WER [%] on the spatialized **two-channel reverberant** wsj1-2mix corpus.

Model	dev	eval
RNN-based MIMO-Speech [13]	34.98	29.99
+ Transformer backend	32.95	28.01
++ Transformer frontend	**31.93**	**26.02**

Table 4. Performance in terms of average WER [%] on the spatialized **two-channel reverberant** wsj1-2mix corpus **after Nara-WPE**.

Model	dev	eval
RNN-based MIMO-Speech	24.45	17.67
+ Transformer backend	**19.17**	**15.24**
++ Transformer frontend	20.55	15.46

tion method. For the RNN-based model, the WER on the evaluation set decreased by 41.1% relative, from 29.99% to 17.67%. Similar to the experiments under other conditions, the model with backend Transformer only is better than the RNN-based baseline model on the reverberant evaluation set by 13.8% relative WER. However, the Transformer-based frontend slightly degraded the performance. This may be due the window size of the attention being too small, as it only covers about 0.3 s of speech. Note that our systems are not trained through Nara-WPE, which is left for future work.

At last, we show results in the single-channel task with the 1st channel of the reverberated speech after Nara-WPE dereverberation in Table 5. Using the RNN-based model, the WER of the evaluation set is high, at 28.21%, which is influenced greatly by the reverberation, even when preprocessing with the dereverberation technique. However, the Transformer-based model can reach a final WER of 16.50%, a 41.5% relative reduction, proving that the Transformer-based model is more robust than the RNN-based model.

Table 5. Performance in terms of average WER [%] on the **1st channel** of the spatialized **reverberant** wsj1-2mix corpus **after Nara-WPE**.

Model	dev	eval
RNN-based 1-channel Model	31.21	28.21
Transformer-based 1-channel Model	20.44	16.50

5. CONCLUSION

In this paper, we applied Transformer models for end-to-end multi-speaker ASR in both the single-channel and multi-channel scenarios, and observed consistent improvements. The RNN-based ASR module is replaced with the Transformers. To alleviate the fatal memory consumption issue when applying Transformers in the frontend with considerably long sequences, we modified the self-attention in the Transformers of the masking network by using a local context window. Furthermore, by incorporating an external dereverberation method, we largely reduced the performance gap between the reverberant condition and the anechoic condition, and hope to further reduce it in the future thanks to tighter integration of the dereverberation within our model.

6. REFERENCES

[1] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *The Journal of the Acoustical Society of America*, vol. 25, no. 5, pp. 975–979, 1953.

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

[3] 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. IEEE ICASSP*, Mar. 2017.

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

[5] X. Chang, Y. Qian, and D. Yu, "Monaural multi-talker speech recognition with attention mechanism and gated convolutional networks," in *Proc. ISCA Interspeech*, 2018.

[6] T. Menne, I. Sklyar, R. Schlüter, and H. Ney, "Analysis of deep clustering as preprocessing for automatic speech recognition of sparsely overlapping speech," in *Proc. ISCA Interspeech*, 2019, pp. 2638–2642.

[7] S. Settle, J. Le Roux, T. Hori, S. Watanabe, and J. R. Hershey, "End-to-end multi-speaker speech recognition," in *Proc. IEEE ICASSP*, 2018, pp. 4819–4823.

[8] 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*, Jul. 2018.

[9] X. Chang, Y. Qian, K. Yu, and S. Watanabe, "End-to-end monaural multi-speaker ASR system without pretraining," in *Proc. IEEE ICASSP*, 2019, pp. 6256–6260.

[10] L. Drude and R. Haeb-Umbach, "Tight integration of spatial and spectral features for bss with deep clustering embeddings." in *Proc. ISCA Interspeech*, 2017, pp. 2650–2654.

[11] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, "Multi-Channel Deep Clustering: Discriminative spectral and spatial embeddings for speaker-independent speech separation," in *Proc. IEEE ICASSP*, 2018, pp. 1–5.

[12] T. Yoshioka, H. Erdogan, Z. Chen, X. Xiao, and F. Alleva, "Recognizing overlapped speech in meetings: A multichannel separation approach using neural networks," in *Proc. ISCA Interspeech*, 2018.

[13] X. Chang, W. Zhang, Y. Qian, J. L. Roux, and S. Watanabe, "MIMO-Speech: End-to-end multi-channel multi-speaker speech recognition," *arXiv preprint arXiv:1910.06522*, 2019.

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

[15] 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.

[16] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *OpenAI preprint*, Jun. 2018.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2019.

[18] S. Karita, N. Enrique Yalta Soplin, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, "Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration," in *Proc. ISCA Interspeech*, 2019, pp. 1408–1412.

[19] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on Transformer vs RNN in speech applications," *arXiv preprint arXiv:1909.06317*, 2019.

[20] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," *arXiv preprint arXiv:1909.06247*, 2019.

[21] M.-T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," in *Proc. EMNLP*, 2015, pp. 1412–1421.

[22] D. Povey, H. Hadian, P. Ghahremani, K. Li, and S. Khudanpur, "A time-restricted self-attention layer for ASR," in *Proc. IEEE ICASSP*, 2018, pp. 5874–5878.

[23] T. Yoshioka and T. Nakatani, "Generalization of multi-channel linear prediction methods for blind mimo impulse response shortening," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 20, no. 10, pp. 2707–2720, 2012.

[24] M. Souden, J. Benesty, and S. Affes, "On optimal frequency-domain multichannel linear filtering for noise reduction," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 18, no. 2, pp. 260–276, 2009.

[25] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi *et al.*, "The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices," in *Proc. IEEE ASRU*, Dec. 2015, pp. 436–443.

[26] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *Proc. IEEE ICASSP*, Mar. 2016.

[27] H. Erdogan, J. R. Hershey, S. Watanabe, M. I. Mandel, and J. Le Roux, "Improved MVDR beamforming using single-channel mask prediction networks," in *Proc. ISCA Interspeech*, Sep. 2016, pp. 1981–1985.

[28] T. Ochiai, S. Watanabe, T. Hori, and J. R. Hershey, "Multichannel end-to-end speech recognition," in *Proc. ICML*, 2017, pp. 2632–2641.

[29] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "ESPnet: End-to-End Speech Processing Toolkit," in *Proc. ISCA Interspeech*, 2018, pp. 2207–2211.

[30] T. Hori, J. Cho, and S. Watanabe, "End-to-end speech recognition with word-based RNN language models," in *Proc. IEEE SLT*, Dec. 2018, pp. 389–396.

[31] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "NARA-WPE: A Python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing," in *ITG Fachtagung Sprachkommunikation (ITG)*, 2018.