



End-to-End Multi-Speaker Speech Recognition using Speaker Embeddings and Transfer Learning

Pavel Denisov, Ngoc Thang Vu

Institute for Natural Language Processing (IMS), University of Stuttgart

{pavel.denisov, thang.vu}@ims.uni-stuttgart.de

Abstract

This paper presents our latest investigation on end-to-end automatic speech recognition (ASR) for overlapped speech. We propose to train an end-to-end system conditioned on speaker embeddings and further improved by transfer learning from clean speech. This proposed framework does not require any parallel non-overlapped speech materials and is independent of the number of speakers. Our experimental results on overlapped speech datasets show that joint conditioning on speaker embeddings and transfer learning significantly improves the ASR performance.

Index Terms: end-to-end asr, overlapped speech

1. Introduction

Recently, deep learning technology has boosted automatic speech recognition (ASR) performance significantly [1, 2, 3, 4]. Overlapped speech – well known in a more general context as the cocktail party problem – remains, however, to be a largely unsolved problem. Its difficulty can be mainly explained by high similarity of acoustic characteristics of signals that need to be separated and absence of any other obvious clues that might guide a potential solution.

Speech recognition of overlapped speech is usually approached in two stages: first overlapped speech is split to separate recordings of each speaker, then speech recognition is performed on separated recordings. Previous works on speech separation problem include computational auditory scene analysis [5], non-negative matrix factorization [6], graphical modeling [7] and spectral clustering [8]. Similarly to speech recognition, speech separation methods have also made major progress with the help of deep learning. The deep clustering method has been introduced in [9] and consequently improved in [10, 11] and has become one of the most remarkable speech separation methods in the recent years. Deep clustering operates on a spectrogram of overlapped speech and learns to map time-frequency (T-F) units to high dimensional embedding space. More recently, a different approach, named VoiceFilter, has been proposed in [12]. The work simplifies the problem of multiclass classification of T-F units over multiple speakers to binary classification between target speaker's speech and everything else. To condition the neural network on specific speaker, the input is extended with a speaker embedding vector, which is extracted by a separately trained network from the reference clean speech.

A new type of ASR systems, called end-to-end ASR, has emerged in the past years [13, 14, 15, 16, 17]. End-to-end ASR maps the acoustic signal to written language with a single encoder-decoder recurrent neural network and does not require any domain specific knowledge for solving of intermediate sub-tasks, such as grapheme-to-phoneme conversion. Recently, two works have proposed to integrate speaker separation stage to end-to-end ASR. The first one [18] connects pretrained deep

clustering model and end-to-end ASR for the subsequent joint fine-tuning for the better ASR results. The second one [19] removes explicit speech separation part and trains end-to-end ASR for simultaneous speech separation and recognition by permutation invariant procedure, in which ASR system is optimized for multiple outputs corresponding to multiple speakers in the input mixture. Joint speech separation and recognition is also described in [20], although this work is based on the conventional ASR. Both [19] and [20] suggest that transfer learning from clean speech ASR improves the results of overlapped speech ASR.

Our work blends the ideas from [21, 22, 12, 18, 19, 20] and proposes to train an end-to-end overlapped speech recognition system conditioned by speaker embeddings and improved by transfer learning from clean speech. It has advantages over [12] [18] and [20] of not requiring parallel clean speech material and over [19] of not depending on the number of speakers. We evaluate our proposed framework on overlapped speech datasets with two and three overlapped speakers, within and across settings. Overall, we observe significant improvements over the baseline end-to-end ASR system.

2. Method

The outline of the proposed method is presented on Figure 1. It shows two separate neural network models, the speaker encoder and the end-to-end ASR, together with their inputs and outputs. The speaker encoder takes reference speech utterances on the input and produces speaker embedding vectors, as described in sections 2.2 and 3.3. The end-to-end ASR takes acoustic features of overlapped speech with speaker embedding vector of the target speaker on the input and generates transcription of the target speaker's speech on the output, which is used to update the parameters of end-to-end ASR model during the training or provided as the final output during the decoding. A more detailed description of end-to-end ASR is given in sections 2.1 and 3.2. Optionally, clean speech can be used during the training in order to perform a basic form of transfer learning, which is described in section 2.3.

2.1. End-to-end ASR

End-to-end ASR has a hybrid CTC/attention architecture, described in detail in [17]. The input of the model is defined as a T -length sequence of d dimensional feature vectors $X = \{x_t \in \mathbb{R}^d | t = 1, \dots, T\}$, and the output of the model is defined as a U -length sequence of output labels $Y = \{y_u \in \mathcal{U} | u = 1, \dots, U\}$, where \mathcal{U} is a set of distinct output labels and usually $U < T$. During the training, a weighted sum of CTC loss and attention-based cross-entropy loss is minimized:

$$\mathcal{L} = \lambda \mathcal{L}_{\text{ctc}} + (1 - \lambda) \mathcal{L}_{\text{att}}, \quad (1)$$

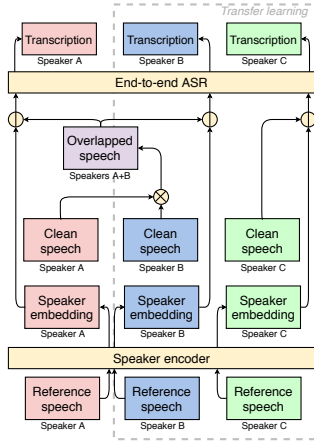


Figure 1: Overview of end-to-end ASR using speaker embeddings and transfer learning.

where $0 \leq \lambda \leq 1$. Attention-based cross-entropy loss is calculated according to predictions of the attention-based encoder-decoder network:

$$\mathcal{L}_{\text{att}} = -\log p_{\text{att}}(Y|X) \quad (2)$$

$$p_{\text{att}}(Y|X) = \prod_u p(y_u|X, y_{1:u-1}) \quad (3)$$

$$p(y_u|X, y_{1:u-1}) = \text{Decoder}(\mathbf{r}_u, \mathbf{q}_{u-1}, y_{u-1}) \quad (4)$$

$$\mathbf{h}_t = \text{Encoder}(X) \quad (5)$$

$$a_{ut} = \text{Attention}(\{a_{u-1}\}_t, \mathbf{q}_{u-1}, \mathbf{h}_t) \quad (6)$$

$$\mathbf{r}_u = \sum_t a_{ut} \mathbf{h}_t. \quad (7)$$

Here, $\text{Encoder}(\cdot)$ and $\text{Decoder}(\cdot)$ are recurrent neural networks, $\text{Attention}(\cdot)$ is an attention mechanism and \mathbf{h}_t , \mathbf{q}_{u-1} and \mathbf{r}_u are the hidden vectors. CTC loss is calculated from a linear transformation of encoder output and all possible T -length sequences of an extended output labels set $Z = \{z_t \in \mathcal{U} \cup \langle \text{blank} \rangle | t = 1, \dots, T\}$, corresponding to sequence Y of original output labels:

$$\mathcal{L}_{\text{ctc}} = -\log p_{\text{ctc}}(Y|X) \quad (8)$$

$$p(Y|X) \approx \underbrace{\sum_Z \prod_t p(z_t|z_{t-1}, Y) p(z_t|X) p(Y)}_{\triangleq p_{\text{ctc}}(Y|X)} \quad (9)$$

$$p(z_t|X) = \text{Softmax}(\text{Lin}(\mathbf{h}_t)). \quad (10)$$

During the decoding, the same CTC and attention-based probabilities are summed with possibly another weight and are used to find the most probable output labels sequence:

$$\hat{Y} = \arg \max_Y \{\lambda \log p_{\text{ctc}}(Y|X) + (1 - \lambda) \log p_{\text{att}}(Y|X)\}. \quad (11)$$

2.2. Speaker embeddings

Speaker embedding is a vector of fixed dimensionality that represents speaker's characteristics and can be extracted from a reference recording of speaker's speech. Speaker embeddings have been shown to be a useful source of information about speaker in many tasks, including speaker verification, speaker diarization [23], speech synthesis [24] and speech separation [12]. We

condition ASR system for the recognition of speech of a certain speaker in the recording of overlapped speech. This arrangement removes a major roadblock of the permutation problem, appearing when multiple correct outputs are possible for a single input, and simplifies application of a wide range of well studied machine learning methods for further system optimization.

2.3. Transfer learning

Speech recognition of overlapped speech can be viewed as speech recognition of clean speech in mismatched conditions and be addressed by numerous transfer learning methods. It has been proposed in [20] to utilize teacher-student training for transfer learning from clean to overlapped speech recognition. This approach has a limitation of applicability on training sets with parallel clean and overlapped speech only, which is hardly achievable in real life scenarios. Another example of transfer learning from clean to overlapped speech recognition is given in [19], which applies parameters transfer. Multi-condition training is an alternative to the parameters transfer method, in which training samples from different conditions are mixed together and used for the training simultaneously. Multi-condition training has been demonstrated to provide better results for transfer learning between languages [25] as well as between channels [26]. In this work, we experiment with both parameters transfer and multi-condition training methods in order to improve overlapped speech recognition by utilization of clean speech training data.

3. Experimental setup

3.1. Datasets

We evaluate our models on the widely used mixed speech datasets wsj0-2mix and wsj0-3mix [9, 10]. The datasets contain two-speaker and three-speaker mixtures of randomly selected utterances from WSJ0 corpus. Training, development and evaluation sets, named tr , cv and tt , are generated from WSJ0 training, development and evaluation sets si_tr_s , si_dt_05 and si_et_05 and contain speech of speakers of both genders in different combinations. We use max version of the datasets, meaning that the length of every mixed speech utterance is chosen to be maximum of the lengths of original utterances being used for a mixture. The sampling rate of the used datasets is 16 kHz. Training, development and evaluation sets of both datasets contain 20000, 5000 and 3000 utterances. Training and development sets contain speech of same 101 speakers, therefore development sets are used for the evaluation in closed speaker set condition. Evaluation sets contain speech of another 19 speakers, therefore evaluation sets are used for the evaluation in open speaker set condition. Total durations of training, development and evaluation sets of wsj0-2mix dataset are 46, 11 and 7 hours. Total durations of training, development and evaluation sets of wsj0-3mix dataset are 51, 13 and 8 hours. LibriSpeech [27] train-clean-100 dataset is used for the transfer learning experiments. It contains 28539 utterances of read speech by 251 speakers and has total duration of 100 hours. The sampling rate of the dataset is 16 kHz.

3.2. Baseline

Our end-to-end ASR system is based on ESPnet toolkit [28] and its WSJ recipe. 80-dimensional log Mel filterbank coefficients with pitch with a frame length of 25 ms and shift of 10 ms are

used as input features. The input features are extracted and normalized to zero mean and unit variance with Kaldi toolkit [29]. The encoder network consists of four BLSTM layers with 320 units in each layer and direction and linear projection layer with 320 units. No subsampling is applied to the input. The decoder network consists of one LSTM layer with 300 units. Additive attention mechanism with 320 dimensions is utilized. We use 49 characters as output units. PyTorch backend of ESPnet is used to implement the networks. Training is performed with AdaDelta optimizer [30] and gradient clipping on two GPUs in parallel with a batch size of 30 for 30 epochs. The optimizer is initialized with $\rho = 0.95$ and $\epsilon = 10^{-8}$. ϵ is halved after an epoch if performance of the model did not improve on development set. The model with the highest accuracy on development set is used for the decoding. The CTC weight λ is set to 0.2 during the training and 0.3 during the decoding. The decoding is performed with a beam search with a beam size of 30.

The decoding makes use of word-based RNN-LM [31] with a weight of 1.0. Word-based RNN-LM is trained on the LM training set of WSJ0 corpus containing 37M words and 1.6M sentences, dictionary size is 65K words. Word-based RNN-LM contains 1 LSTM layer with 1000 units. The stochastic gradient descent optimizer is used to train word-based RNN-LM with a batch size of 300 for 20 epochs.

3.3. Speaker embeddings extractor

We extract 512-dimensional speaker embeddings from the reference utterances with x-vector system from Kaldi toolkit. We use the pretrained model downloaded from <http://kaldi-asr.org/models/m8>. The model is trained on augmented VoxCeleb 1 [32] and VoxCeleb 2 [33] datasets according to the procedure closely following the description in [34] and evaluated on Speakers in the Wild dataset [35] with 3.5% equal error rate. The input features of x-vector extractor are 30-dimensional MFCCs without cepstral truncation with a frame length of 25 ms and shift of 10 ms. Mean normalization with a sliding window of up to three seconds is applied to the input features. Speaker embeddings are extracted from voiced frames only, which are selected by the same energy-based VAD. We obtain one vector from each utterance and average them per speaker to get speaker specific vector. L_2 normalized speaker embedding is used as an additional input for the ASR model. We denote insertion of the speaker embedding vector to the beginning of the sequence of acoustic features vectors as *horizontal stacking*. We denote concatenation of the speaker embedding vector with every acoustic features vector as *vertical stacking*. Horizontal stacking requires speaker embedding and acoustic features to have same size, which can be achieved either by downscaling of the speaker embedding vector, which we perform with the trainable linear transformation, or by padding of the acoustic feature vectors, which we perform by appending appropriate number of zeros to each acoustic features vector. Vertical stacking allows the size of the speaker embedding to be independent of the size of the acoustic features.

4. Results

4.1. Speaker embeddings inclusion strategies

The first set of experiments aims to determine the best strategy for inclusion of speaker embeddings in the model’s input. While vertical stacking does not enforce same size of the speaker embedding and acoustic features, we perform two experiments with vertical stacking: the first one with unchanged

sizes of the input vectors and the second one with the down-scaled speaker embedding. The second experiment isolates the effect of different stacking types from the effect of the speaker embedding downscaling. 50 utterances of reference speech are used to produce each speaker embedding in this experiment. It can be seen from the data in Table 1 that vertical stacking clearly outperforms horizontal stacking, while the downscaling of the speaker embedding results in minor degradation of the model’s performance.

Table 1: *Results (WER, %) with different speaker embedding inclusion strategies on the two-speaker overlapped speech dataset.*

Strategy	dev	eval
Baseline (no speaker embeddings)	79.6	85.7
Horizontal stacking with downscaled embedding	77.0	84.1
Horizontal stacking with padded acoustic features	83.0	88.6
Vertical stacking with downscaled embedding	11.7	24.9
Vertical stacking with unchanged vectors’ sizes	11.4	22.1

4.2. Amount of reference speech data

The next set of experiments is concerned with amount of reference speech data required for the generation of speaker embeddings. Table 2 gives an overview of ASR performance with different numbers of reference utterances in case of two-speaker overlapped speech. It is apparent from this table that larger amount of reference speech data allows to generate more general speaker embeddings preventing the ASR model from the overfitting towards known speakers and seen utterances. It is worth noting, however, that even one reference utterance of the duration of approximately ten seconds appears to be sufficient material for the major improvement.

Table 2: *Results (WER, %) with different amount of reference speech data on the two-speaker overlapped speech dataset.*

Utterances	Reference speech amount per speaker		dev	eval
	Seconds	Voiced frames		
1	8.3 \pm 2.9	641 \pm 271	15.8	32.6
5	42.3 \pm 6.4	3156 \pm 623	17.3	29.1
10	84.6 \pm 10.1	6341 \pm 1090	11.3	22.6
20	170.3 \pm 16.9	12545 \pm 1783	10.8	22.5
50	426.3 \pm 34.2	31490 \pm 4260	11.4	22.1

4.3. Transfer learning

We evaluate two transfer learning approaches, namely parameters transfer and multi-condition training, for improving ASR performance on overlapped speech by utilization of non-parallel clean speech training data. Table 3 presents the results of the systems trained on training sets of wsj0-2mix and wsj-3mix datasets and tested on development and evaluation sets of the corresponding datasets. The most striking observation to emerge from the results is that the training process has not converged on the dataset with three overlapping speakers, but the system has been able to decode the same recordings when trained on the combination of overlapped and clean speech datasets. This finding can be attributed to higher complexity of the modeled function in case of increased number of overlapping speakers, which the neural network could not learn just from the overlapped speech data, and demonstrates how crucial the role of the transfer learning approach for the solution

of certain problems can be. The transfer learning results are expectedly better on the dataset with two overlapping speakers as well, especially for the open speaker condition, what is due to the number of additional speaker embeddings in the training data and subsequent better generalization of the relationship between speaker embedding and relevant acoustic features. Finally, multi-condition training has demonstrated slightly better results than parameters transfer. This finding is in agreement with the previous reports on transfer learning applications in ASR.

Table 3: Results (WER, %) of the baseline ASR, speaker embeddings conditioning and transfer learning on the two- and three-speaker overlapped speech datasets.

	wsj0-2mix		wsj0-3mix	
	dev	eval	dev	eval
Baseline	79.6	85.7	95.9	96.0
+ speaker embeddings	11.4	22.1	95.6	95.7
+ parameters transfer	8.8	16.9	22.7	45.3
+ multi-condition training	8.5	14.6	21.7	42.9

As our method does not utilize any explicit knowledge about the number of overlapping speakers, the models should theoretically also work for testing data with larger or smaller number of overlapping speakers than in the training data. We test whether this is true in practice by decoding wsj0-3mix testing data with the model trained on wsj0-2mix training data (combined with LibriSpeech 100) and vice versa. Encouraged by the success of the previous transfer learning experiments, we also train a system on a combination of clean and overlapped two- and three-speaker datasets. The results of testing with mismatching number of overlapping speakers are given in Table 4. In general it seems that the proposed method does not depend on the number of overlapping speakers but can benefit from training on larger amount of different conditions of speech overlap. A possible explanation of the slightly worse result of the best system on the open speaker set condition with two overlapping speakers might be a bias towards WSJ0 speakers in the combined training dataset introduced by the addition of wsj0-3mix dataset, which is slightly larger than wsj0-2mix dataset and therefore has more impact on two-speaker testing data than wsj0-2mix on three-speaker testing data in this experiment.

Table 4: Results (WER, %) of the proposed method depending on the training data.

Training data	wsj0-2mix		wsj0-3mix	
	dev	eval	dev	eval
LibriSpeech 100 + wsj0-2mix	8.5	14.6	45.2	55.3
LibriSpeech 100 + wsj0-3mix	7.8	28.5	21.7	42.9
LibriSpeech 100 + wsj0-{2,3}mix	4.8	15.2	15.5	32.3

4.4. Comparison with earlier work

Table 5 compares our best result on the evaluation set of wsj0-2mix dataset with the results reported on the same dataset in the previous works. From the table we can see that the proposed system outperforms the best known result by 42% relatively. However, it should be noted that the listed systems differ from each other in a number of ways, including types of ASR system, types of LM and types and amount of training data.

Following [19], we also visualize the encoder networks outputs of an example utterance (see Figure 2). We apply principal

Table 5: Results (WER, %) of the proposed method and previous works on the two-speaker overlapped speech dataset.

Method	eval
Deep Clustering, hybrid ASR [10]	30.8
Permutation Invariant Training, hybrid ASR [36]	28.2
Permutation Invariant Training, end-to-end ASR [19]	28.2
Speaker Parallel Attention, end-to-end ASR [37]	25.4
Proposed, end-to-end ASR	14.6

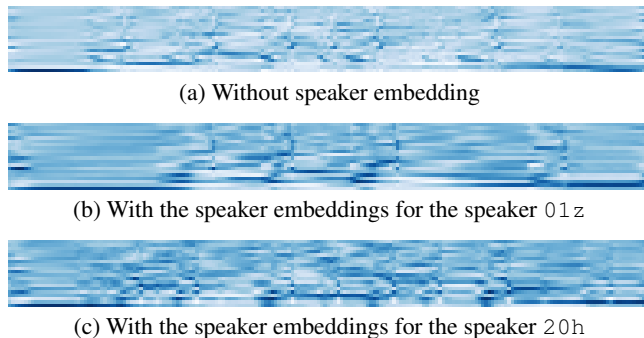


Figure 2: Visualization of the hidden vector sequences for the utterance 01zc020o.2.3474_20hc010j_-2.3474 of wsj-2mix dataset.

component analysis to the hidden vectors on the vertical axis as well. Figure 2(a) shows the output of the baseline’s encoder, while figures 2(b) and 2(c) show the encoder network conditioned on speaker embeddings of two different speakers. Some patterns from the encoder’s output of the baseline model appear on the conditioned encoder’s output for the first speaker, and another ones appear on the encoder’s output for the second speaker. This observation suggests that the conditioning on the speaker embeddings indeed allows the encoder network to perform the separation of overlapped speech.

5. Conclusions

In this paper, we proposed an effective end-to-end speech recognition framework for overlapped speech using speaker embeddings and transfer learning techniques. Experimental results on simulated overlapped speech datasets revealed that using speaker embeddings our framework was able to automatically identify relevant information of the target speaker for recognition. The application of transfer learning technique played a crucial role with the increasing number of speakers. Finally, we observed significant improvements over the baseline end-to-end system even while using just ten seconds of reference speech per speaker.

6. References

- [1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, “Deep Neural Networks for Acoustic Modeling in Speech Recognition,” *Signal Processing Magazine*, 2012.
- [2] G. Dahl, D. Yu, L. Deng, and A. Acero, “Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition,” in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, 2012, pp. 30–42.
- [3] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke,

- D. Yu, and G. Zweig, "The Microsoft 2016 Conversational Speech Recognition System," *CoRR*, 2016.
- [4] 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 *Interspeech*, 2016, pp. 2751–2755.
- [5] D. Wang and G. J. Brown, *Computational auditory scene analysis: Principles, algorithms, and applications*. Wiley-IEEE press, 2006.
- [6] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *Ninth International Conference on Spoken Language Processing*, 2006.
- [7] J. R. Hershey, S. J. Rennie, P. A. Olsen, and T. T. Kristjansson, "Super-human multi-talker speech recognition: A graphical modeling approach," *Computer Speech & Language*, vol. 24, no. 1, pp. 45–66, 2010.
- [8] F. R. Bach and M. I. Jordan, "Learning spectral clustering, with application to speech separation," *Journal of Machine Learning Research*, vol. 7, no. Oct, pp. 1963–2001, 2006.
- [9] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 31–35.
- [10] Y. Isik, J. Le Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-Channel Multi-Speaker Separation Using Deep Clustering," *Interspeech 2016*, pp. 545–549, 2016.
- [11] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, "Alternative Objective Functions for Deep Clustering," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018*.
- [12] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno, "Voice-Filter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking," *arXiv preprint arXiv:1810.04826*, 2018.
- [13] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.
- [14] Y. Miao, M. Gowayyed, and F. Metze, "EESSEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 167–174.
- [15] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4945–4949.
- [16] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4960–4964.
- [17] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.
- [18] S. Settle, J. Le Roux, T. Hori, S. Watanabe, and J. R. Hershey, "End-to-end multi-speaker speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4819–4823.
- [19] 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 *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2018, pp. 2620–2630.
- [20] Z. Chen, J. Droppo, J. Li, and W. Xiong, "Progressive joint modeling in unsupervised single-channel overlapped speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 1, pp. 184–196, 2018.
- [21] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Higuchi, A. Ogawa, and T. Nakatani, "Speaker-Aware Neural Network Based Beamformer for Speaker Extraction in Speech Mixtures," 2017, pp. 2655–2659.
- [22] J. Wang, J. Chen, D. Su, L. Chen, M. Yu, Y. Qian, and D. Yu, "Deep Extractor Network for Target Speaker Recovery from Single Channel Speech Mixtures," *Proc. Interspeech 2018*, pp. 307–311, 2018.
- [23] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with LSTM," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5239–5243.
- [24] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, Y. Wu *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in Neural Information Processing Systems*, 2018, pp. 4485–4495.
- [25] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *ICASSP*. IEEE, 2013, pp. 8619–8623.
- [26] P. Ghahremani, V. Manohar, H. Hadian, D. Povey, and S. Khudanpur, "Investigation of transfer learning for ASR using LF-MMI trained neural networks," in *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*. IEEE, 2017, pp. 279–286.
- [27] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5206–5210.
- [28] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N.-E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "ESP-net: End-to-End Speech Processing Toolkit," *Proc. Interspeech 2018*, pp. 2207–2211, 2018.
- [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*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.
- [30] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.
- [31] T. Hori, J. Cho, and S. Watanabe, "End-to-end Speech Recognition with Word-based RNN Language Models," *arXiv preprint arXiv:1808.02608*, 2018.
- [32] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," *Proc. Interspeech 2017*, pp. 2616–2620, 2017.
- [33] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," *Proc. Interspeech 2018*, pp. 1086–1090, 2018.
- [34] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," *Submitted to ICASSP*, 2018.
- [35] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The Speakers in the Wild (SITW) Speaker Recognition Database," in *Interspeech*, 2016, pp. 818–822.
- [36] Y. Qian, X. Chang, and D. Yu, "Single-channel multi-talker speech recognition with permutation invariant training," *Speech Communication*, vol. 104, pp. 1–11, 2018.
- [37] X. Chang, Y. Qian, K. Yu, and S. Watanabe, "End-to-End Monaural Multi-speaker ASR System without Pretraining," *arXiv preprint arXiv:1811.02062*, 2018.