

# Sequence-level self-learning with multiple hypotheses

Kenichi Kumatani, Dimitrios Dimitriadis, Yashesh Gaur, Robert Gmyr,  
Sefik Emre Eskimez, Jinyu Li, Michael Zeng

Microsoft, WA, USA

## Abstract

In this work, we develop new self-learning techniques with an attention-based sequence-to-sequence (seq2seq) model for automatic speech recognition (ASR). For untranscribed speech data, the hypothesis from an ASR system must be used as a label. However, the imperfect ASR result makes unsupervised learning difficult to consistently improve recognition performance especially in the case that multiple powerful teacher models are unavailable. In contrast to conventional unsupervised learning approaches, we adopt the *multi-task learning* (MTL) framework where the  $n$ -th best ASR hypothesis is used as the label of each task. The seq2seq network is updated through the MTL framework so as to find the common representation that can cover multiple hypotheses. By doing so, the effect of the *hard-decision* errors can be alleviated. We first demonstrate the effectiveness of our self-learning methods through ASR experiments in an accent adaptation task between the US and British English speech. Our experiment results show that our method can reduce the WER on the British speech data from 14.55% to 10.36% compared to the baseline model trained with the US English data only. Moreover, we investigate the effect of our proposed methods in a federated learning scenario.

**Index Terms:** unsupervised learning, self-learning, encoder-decoder, multi-task learning, end-to-end acoustic modeling

## 1. Introduction

Unsupervised learning of an acoustic model (AM) has a long history from the Bayesian model to the deep neural network (DNN) system in the field of automatic speech recognition (ASR) [1, 2, 3, 4, 5]. It is typically done by transferring knowledge from stronger teacher model(s) to the student model [6, 7, 8, 9, 10, 11, 12] or adapting the seed model pre-trained with a sufficient amount of labeled data [1, 2, 3, 13, 14, 15]. Because of no need of heavy decoding processes with teacher models, the latter self-training approach is perhaps preferable in many situations such as on-device personalization and federated learning scenarios [16, 17, 18, 19]. In this work, we thus address the self-training task for AM without an additional bigger teacher model and parallel adaptation data<sup>1</sup>.

The optimization criteria of unsupervised learning for DNN-based ASR systems can fall into two categories: frame-level [8, 10, 11, 12], and sequence-level criteria [7, 15, 9]. It is well known that direct sequence-level optimization [15] and sequence training from the model initialized with the frame-level cross-entropy objective [10, 11] provide better recognition

<sup>1</sup>The definitions of unsupervised learning and self-learning vary in different applications and fields. For presentation purposes, we will refer any model optimization methods without manual transcripts as unsupervised learning even if the initial seed model is pre-trained with labeled data. We will say self-learning is a special class of unsupervised learning techniques which optimizes the seed model without another teacher model.

accuracy. Many successful results have been reported by adapting the hybrid model of the DNN and hidden Markov model (HMM) in an unsupervised way.

In the hybrid DNN-HMM systems, acoustic, pronunciation, and language models have been trained separately, each with a different objective. While such an approach can use each type of training data efficiently, this disjoint modeling method leads to the sub-optimal solution, especially in terms of model size. This issue has been addressed by designing the *end-to-end* network, which directly generates a word or character sequence from a sequence of speech features. Two popular approaches in this area are perhaps recurrent NN transducer (RNN-T) [20] and attention-based encoder-decoder (AED) network [21, 22, 23]. Unlike connectionist temporal classification (CTC) [24], both methods do not make an unreasonable assumption for ASR that the label outputs are conditionally independent of each other. It is shown in [25] that the AED has the potential of outperforming the RNN-T arguably because of a more flexible assumption on the alignment between input and output, given a large number of training samples [26, 27].

Based on the reasons above, we focus on the development of unsupervised learning methods for the AED network. For that, knowledge distillation (KD) [28, 29, 9] is a promising approach. In particular, sequence-level KD [29, 9] is more suitable for the AED network to improve modeling capability for sequential input and output. Kim and Rush indeed showed in [29] that the sequence-level KD technique can achieve better machine translation performance with lower complexity. Sequence-level KD is also applied to the ASR tasks: model compression [9] and domain adaptation with a slight modification of information injected to the student [30]. Notice that the unsupervised learning method proposed in [30] requires parallel data for the teacher and student models, a pair of clean and noisy data. Again, the use of the bigger teachers or additional parallel adaptation data is not possible in some application scenarios, such as a federated learning scenario [18].

In contrast to prior work on unsupervised learning for the AED [9, 30], we develop the new self-learning techniques which neither require gigantic teacher models nor the parallel data. In such a scenario, we must use the ASR output of the seed model as the target labels. The use of the inaccurate ASR result as the ground truth may degrade recognition accuracy after adaptation. In order to alleviate such a noisy transcript issue, we use the  $N$ -best hypotheses from the ASR output for adaptation. More specifically, we optimize the AED with multiple objective functions associated with the  $N$ -best hypotheses. We implement the optimization algorithm with the multi-label learning (MLL) or multi-task learning (MTL) framework [31, 32]. Our techniques proposed here are evaluated through two kinds of ASR tasks: an accent adaptation task and a federated learning situation.

The rest of the paper is organized as follows. Section 2

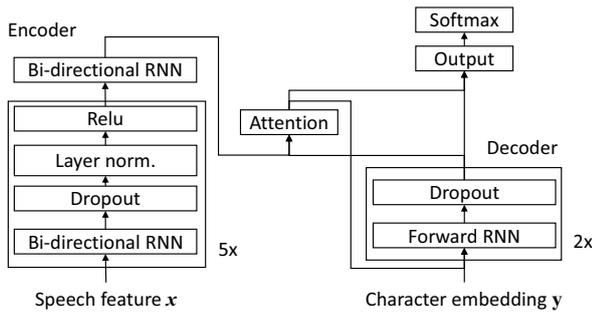


Figure 1: Attention-based encoder-decoder (AED) network

briefly reviews the sequence-to-sequence model with the AED and its optimization criterion in supervised and unsupervised manners. Section 3 describes our unsupervised learning algorithms for the attention-based encoder-decoder network. Section 4 describes the ASR experiments in two kinds of tasks: accent adaptation and federated transfer learning tasks. In section 5, we conclude this work.

## 2. Background

### 2.1. Attention-based encoder-decoder (AED)

Let  $\mathbf{x} = [x_1, \dots, x_K]$  and  $\mathbf{y} = [y_1, \dots, y_M]$  be the feature input and target word embeddings with  $K$  and  $M$  being the source and target lengths, respectively. ASR involves finding the most probable target sequence given the feature input:

$$\hat{\mathbf{y}} = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} \log p(\mathbf{y}|\mathbf{x}) \quad (1)$$

where  $\mathcal{Y}$  is the set of all possible sequences. As shown in Figure 1,  $p(\mathbf{y}|\mathbf{x})$  is modeled with an encoder, a decoder and an attention component. The encoder network reads the feature sequence and the decoder network produces a distribution over the target words, one word embedding at a time given the source [23, 26]. We employ the attention architecture from [21] by replacing the scoring function with the ReLU unit.

### 2.2. Supervised learning

We briefly review supervised learning for the sequence-to-sequence model before formulating the unsupervised learning problem. First, consider the sequence-level distribution specified with the model parameters  $\theta$  over all possible sequences  $\mathbf{y} \in \mathcal{Y}$  in the log domain,

$$\log p(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^M \log p(y_i|\mathbf{x}, \mathbf{y}_{<i}; \theta) \quad (2)$$

where  $\mathbf{y}_{<i}$  is either the ground truth of the previous characters or sampled from the model with a certain sampling rate in order to make the model robust against prediction errors during inference [23]. The sampling rate can be constant [23] or scheduled [33]. For the experiment, we use the scheduled sampling method. Sequence-level training then involves estimating  $\theta$  that maximizes the log-likelihood (2) over all the sequences.

### 2.3. Unsupervised learning

The self-learning framework of the attention-based sequence-to-sequence model is very similar with sequence-level knowledge distillation (KD) [29]. The only difference is that there is no better teacher model which transfers knowledge to the seed

model in the self-learning scenario [34]. In such a scenario, the seed model must adapt itself to new data without knowledge distillation from the more powerful teacher model; the sequence distribution of target words can be very noisy. The erroneous labels can degrade accuracy of the seed model severely [5, 34]. In this section, we describe how we approximate the objective function in the case of the unsupervised learning scenario.

First, let us consider the case of the semi-supervised setting with sequence-level KD. As shown in [29], the distribution from the data is replaced with a probability distribution derived from another teacher model  $q(\mathbf{y}|\mathbf{x})$ . The loss function of sequence-level KD can be written as

$$\mathcal{L}_{\text{SEQ-KD}} = - \sum_{\mathbf{y} \in \mathcal{Y}} q(\mathbf{y}|\mathbf{x}) \log p(\mathbf{y}|\mathbf{x}) \quad (3)$$

Calculating (3) is computationally intractable. Kim and Rush showed in [29] that the loss (3) could be well approximated with the best hypothesis from the beam search output with the teacher model in a machine translation task. With the first best hypothesis  $\hat{\mathbf{y}}_{T,1}$ , they approximated (3) as

$$\mathcal{L}_{\text{SEQ-KD}} \approx - \log p(\mathbf{y} = \hat{\mathbf{y}}_{T,1}|\mathbf{x}) \quad (4)$$

It is worth noting that the same algorithm was also applied to an ASR model compression task in [9] although their improvement was not as significant as that reported in [29].

Now consider the case of self-learning. Again, we only have the seed model in this case. Thus, we use the best hypothesis of the seed model  $\hat{\mathbf{y}}_{S,1}$  instead of that of the teacher model  $\hat{\mathbf{y}}_{T,1}$ . It is straightforward to express the loss approximation function:

$$\mathcal{L}_{\text{SEQ-KD}} \approx - \log p(\mathbf{y} = \hat{\mathbf{y}}_{S,1}|\mathbf{x}) \quad (5)$$

The model can still be improved with self-learning since the seed model learns features from unseen data. However, the seed model needs to provide reasonably good accuracy on the unseen data. Otherwise, the model may degrade due to the erroneous label estimates.

## 3. Proposed self-learning method

### 3.1. Loss function approximation with multiple hypotheses

In many cases, the best ASR output contains errors. Thus, approximating (3) with the only best hypothesis may degrade recognition performance after unsupervised training. Instead, we use the  $N$ -best hypotheses weighted based on the ASR confidence score for approximation. First, let us denote the weight normalized with the Softmax function for the  $n$ -th best candidate  $\mathbf{y}_{S,n}$  as

$$q_n(\mathbf{y}_{S,n}) = \frac{\exp(s_n/T)}{\sum_{i=1}^N \exp(s_i/T)} \quad (6)$$

where  $s_n$  is the  $n$ -th best confidence score and  $T$  is a temperature which controls a probability distribution over the classes [28]. We set  $T = 1$  for the experiments. We then rewrite the loss function (3) as

$$\mathcal{L}_{\text{SEQ-KD}} \approx - \sum_{n=1}^N q_n(\hat{\mathbf{y}}_{S,n}) \log p(\mathbf{y} = \hat{\mathbf{y}}_{S,n}|\mathbf{x}) \quad (7)$$

Minimizing this loss function can be done by creating multiple labels associated with weights for one utterance. We refer to this method as multi-label learning (MLL) in this paper. The loss function can be also optimized through multi-task learning (MTL) [31, 32] as it will be explained in the next section.

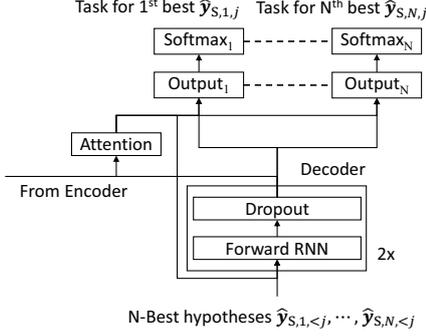


Figure 2: MT network with encoder-decoder shared.

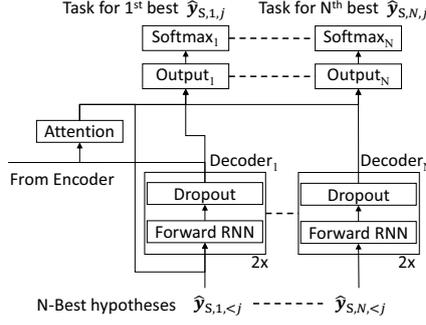


Figure 3: MT network with encoder shared.

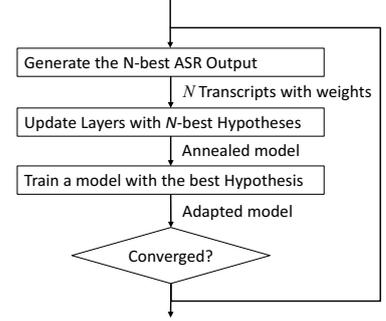


Figure 4: Flow chart of unsupervised training with the  $N$ -best hypotheses

### 3.2. Optimization with multi-task learning

Optimization of the loss function (7) can be done with the multi-task (MT) network where each separate Softmax layer is associated with the loss of the  $n$ -th hypothesis. We consider two network architectures in this work. Figure 2 shows the first architecture proposed here. In the architecture shown in Figure 2, the parameters of the encoder, attention and decoder network are shared during the optimization process. Only the output layer is separated for each task associated with the term in (7). The weights of the new branch for the  $n$ -th best hypothesis are initialized with those of the main task.

Figure 3 shows the second architecture investigated in this work. In contrast to the architecture in Figure 2, the second network architecture shares the encoder and attention module only; the set of the decoder and the output layer is separated for each task. After self-training, the encoder is expected to generate a better acoustic embedding that covers the multiple hypotheses. This can be viewed as acoustic feature-space adaptation with the AED network.

Figure 4 shows a flow chart of our unsupervised training scheme with the  $N$ -best hypotheses. As shown in Figure 4, we first decode the sequence of the speech features with the beam search in order to obtain the  $N$ -best hypotheses,  $\hat{y}_{S,1} \cdots \hat{y}_{S,N}$ . Then, the seed model is adapted with the  $N$  hypotheses. This can be done by either multi-label learning (MLL) or MTL framework, as described before. The normalized probability computed with (6) is used for weighting the loss of each utterance in MLL and the task in MTL. After adopting the model with the  $N$ -th hypotheses, we train the model with the best hypothesis only. The only single task branch is used in the case of MTL. These steps will be repeated until there is no improvement in a character error rate on the validation data. During decoding, we can remove the sub-task branches and keep the main task branch only.

## 4. ASR experiment

In this section, we describe the ASR experiments in order to compare the unsupervised learning techniques with the AED.

### 4.1. Accent Adaptation Task

First, we addressed the English accent adaptation task from American English to British English. The data used for the experiment were collected through the live traffic and stored with the sampling rate of 16 kHz. For the experiments, we train the seed model with approximately 75,000 hours of American En-

glish speech data. About 1,000 hours of the British English speech data were used for adaptation. The adaptation data are treated as untranscribed in the case of unsupervised learning experiments. The British Cortana corpus was used as a test set. The test data were also collected through the live traffic and consists of 14,000 utterances spoken by hundreds of users.

The feature extraction front-end computes 40 log mel-filterbank energy features at a frame rate of 10 milliseconds (msec), concatenates three adjacent feature frames, and then downsamples it into a 30 msec frame rate. Each feature was rescaled to have zero mean and unit variance over the training set. The global mean and variance calculated with the training data were applied to adaptation and evaluation. In order to augment adaptation data, we randomly scale the amplitude of a signal, change the speed in the same manner implemented in Sox software [35] and perform the spec-augment on the feature domain [36]. The network used here has 6 layers of 1024 bidirectional GRU nodes in the encoder and 2 layers of 1024 unidirectional GRU nodes in the decoder. All dropout types were applied with 0.1 dropout probability. We used label smoothing and Adam optimizer in the experiments.

Table 1 shows the word error rates (WERs) obtained with different models, the seed model trained with American English data, British AMs adapted in the supervised and unsupervised learning manners. In the self-learning scenario, we performed 4 types of sequence-level self-knowledge distillation methods, conventional training with the 1-best hypothesis (5), multi-label learning (MLL) with the 4-best hypotheses (7), multi-task learning (MTL) with the attention, encoder and decoder (AED) shared and that with the attention and encoder (AE) shared. For unsupervised MTL, we use the 4-best hypotheses. It is clear from Table 1 that the WER of the American English model can be reduced from 14.55% to 7.8% by supervised learning with the manual transcripts. It is also clear from Table 1 that the WER of the seed model can be reduced by the self-learning methods, although the improvement is not as significant as that obtained with supervised learning. Table 1 also shows that the use of the 4-best hypotheses provides better recognition accuracy than the case of the 1-best hypothesis only. The WERs of the 1-best and  $N$ -best results on the adaptation data obtained with the seed model were 22.6% and 13.7%, respectively. Thus, we consider using multiple hypotheses to provide better recognition accuracy. We did not observe a significant difference among different self-learning methods with the 4-best hypotheses after decoding the adaptation data twice, but it is worth mentioning that MTL with the shared AE network provided the lowest WER with the first decoding pass.

Table 1: Word error rates (WER) on the British Cortana data for each training method

Adaptation Type	Adaptation Method	WER(%)
Seed model	US baseline model	14.55
Supervised learning	Training with transcription	7.8
	Training with the 1-best hypo	12.09
Self-learning	MLL with 4-best hypos	10.66
	MTL with shared AED	10.83
	MTL with shared AE	10.36

## 4.2. Librispeech Task

We also conducted ASR experiments with the Librispeech (LS) data [37] for this study. For the LS task, we used the same front-end as described in Section 4.1 but slightly different model configuration. The AED network used here consists of 6 layers of 1024 bidirectional LSTM nodes and 2 layers of 1024 unidirectional LSTM nodes. For the LS task, we used byte-pair encoding (BPE) [38] to create 16,000 subword units. The optimizer settings are the same as described in [18].

The LS corpus contains approximately 1000 hours of read speech for training. We split the whole LS training set into two subsets. A seed model was first trained with the first half of the training set. For adaptation, we further divide the second half dataset into the per-speaker dataset. The adaptation dataset contains approximately 1100 speakers. Since the seed model does not see the LS vocabulary and word sequences of the adaptation dataset, the ASR result on the adaptation data will not be perfect; self-learning has to be done with noisy labels. The seed model was then adapted with the speaker-partitioned dataset through our federated learning simulator [18, 19]. Briefly, the federated learning algorithm iteratively performs the following processes on a server and clients: (1) broadcasting the global (seed) model from the server to the clients, (2) decoding speech with the global model at each client, (3) adapting the global model with each client’s data and the ASR hypotheses, (4) encrypting each client’s model so as to ensure privacy<sup>2</sup>, (5) aggregating the linearly-weighted models from a pool of clients, and (6) optimizing the global model through the aggregated gradients. Those processes are repeated until the loss on the validation data is converged. We only decode speech every 512 aggregation steps for unsupervised training. We found that this could avoid training divergence due to inaccurate ASR results with an intermediate model trained halfway. Notice that the adapted model is only uploaded to the server while privacy data such as speech and transcripts are secured in the client. The adapted models will be sent to the server after processing all the utterances per speaker on each client. This saves a tremendous amount of network bandwidths.

Table 2 shows the WERs on the LS test clean set obtained with different models in the same way as Table 1. We performed supervised and unsupervised learning algorithms on the seed model with the second half of the training data. The last row in Table 2 shows the upper bound WER in the case that the model is trained from scratch with the full amount of the LS training data. It is clear from Table 2 that the WER of the seed model can be significantly reduced by supervised-training with the rest 50% of the training data from 5.66% to 4.40%. It is also clear from Table 2 that recognition accuracy can still be improved by self-learning with MTL even when the seed model does not know the vocabulary and word sequences of the adaptation data. We observed that the validation loss diverged in

<sup>2</sup>We did not use encryption here for the sake of simplicity although it is straightforward to apply multi-party computation (MPC) secure algorithms or differential privacy techniques such as [17].

Table 2: WERs on the LS test clean dataset for each training method in the privacy preserving scenario (where the seed model has no language information on the adaptation data)

Adaptation Type	Adaptation Method	WER(%)
Seed model	Trained from scratch with 50% of data	5.66
Supervised learning	Training with the rest 50% of data	4.40
	Training with the 1-best hypo	5.58
Self-learning	MLL with 4-best hypos	5.72
	MTL with shared AED	5.55
	MTL with shared AE	5.22
Reference	The whole LS training data in centralized training	4.00

MLL, which very likely led to recognition accuracy degradation. The results also show that the convergence performance is improved by separating more layers relevant to the language model, the decoder and output layers, for each hypothesis. This suggests that the  $N$  best hypotheses contain little language information shareable over the decoders and output layers. This may be because language knowledge of the seed model on the adaptation dataset is poor. In order to check how much the  $N$ -best results can potentially help recognition accuracy, we decoded the second half of the LS training data with our LS seed model. The WERs of the 1-best and  $N$ -best results on the LS adaptation data were 10.4% and 7.62%. The difference between the 1-best and  $N$ -best results was not as significant as that in the case of the accent adaptation task. We assume that many word errors in this LS task cannot be recovered by simply increasing the  $N$ -best hypotheses because the seed model does not observe the language of the adaptation dataset. On the other hand, the seed model used in the British Cortana task should cover the vocabulary and grammar for the adaptation data very well because it is trained with a large amount of the American English data containing the Cortana task. Of course, there are some different word usages between the American and British English. We are, however, led to conclude that the short phrase ASR task such as the Cortana task was probably not affected significantly.

## 5. Conclusions

In this work, we have investigated the ASR accuracy of self-learning methods with the attention-based sequence-to-sequence model. We proposed the new self-learning methods with the  $N$  best hypotheses; the seed model is adapted with MLL or MTL in an end-to-end manner. We conducted two kinds of ASR experiments: the English accent adaptation task and the federated-learning scenario. In the accent adaptation task, our experiment results show that our best self-learning method can reduce the WER on the British speech data from 14.55% to 10.36% compared to the baseline model trained with the US English data only. In the federated-learning scenario, the improvement on the LS dataset was not as significant as that on the Cortana dataset. In contrast to the Cortana task, the seed model used in the LS task does not contain language information on the adaptation data well. Thus, misrecognition results may not be entirely recoverable by increasing the  $N$ -best candidates. We plan to combine our AM self-learning and the language data augmentation method a text-to-speech module [39]

## 6. Acknowledgements

The authors would like to thank Masaki Itagaki, Heiko Rahmel, Naoyuki Kanda, Lei He, Ziad Al Bawab, Jian Wu, and Xuedong Huang for their project support and technical discussions.

## 7. References

- [1] P. C. Woodland, D. Pye, and M. J. F. Gales, "Iterative unsupervised adaptation using maximum likelihood linear regression," in *The 4th International Conference on Spoken Language Processing, Philadelphia, PA, USA, October 3-6, 1996*. ISCA, 1996.
- [2] G. Zavaliagkos and T. Colthurst, "Utilizing untranscribed training data to improve performance," in *DARPA Broadcast News Transcription and Understanding Workshop, Landsdowne*, 1998, pp. 301–305.
- [3] L. Lamel, J. luc Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech and Language*, vol. 16, pp. 115–129, 2002.
- [4] F. Wessel and H. Ney, "Unsupervised training of acoustic models for large vocabulary continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 1, pp. 23–31, 2005.
- [5] Y. Huang, Y. Wang, and Y. Gong, "Semi-supervised training in deep learning acoustic model," in *Proc. Interspeech*, September 2016.
- [6] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning small-size dnn with output-distribution-based criteria," in *Proc. Interspeech*, September 2014.
- [7] J. H. M. Wong and M. J. F. Gales, "Sequence student-teacher training of deep neural networks," in *Proc. Interspeech*, September 2016.
- [8] J. Li, M. L. Seltzer, X. Wang, R. Zhao, and Y. Gong, "Large-scale domain adaptation via teacher-student learning," in *Proc. Interspeech*, 2017, pp. 2386–2390.
- [9] R. M. Mun'im, N. Inoue, and K. Shinoda, "Sequence-level knowledge distillation for model compression of attention-based sequence-to-sequence speech recognition," *CoRR*, vol. abs/1811.04531, 2018.
- [10] S. H. K. Parthasarathi, N. Sivakrishnan, P. Ladkat, and N. Strom, "Realizing petabyte scale acoustic modeling," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 2, pp. 422–432, 2019.
- [11] L. Mosner, M. Wu, A. Raju, S. H. K. Parthasarathi, K. Kumatani, S. Sundaram, R. Maas, and B. Hoffmeister, "Improving noise robustness of automatic speech recognition via parallel data and teacher-student learning," in *Proc. ICASSP*, 2019, pp. 6475–6479.
- [12] Z. Meng, J. Li, Y. Zhao, and Y. Gong, "Conditional teacher-student learning," *CoRR*, vol. abs/1904.12399, 2019.
- [13] J. Huang, E. Marcheret, K. Visweswariah, V. Libal, and G. Potamianos, *The IBM Rich Transcription 2007 Speech-to-Text Systems for Lecture Meetings*. Berlin, Heidelberg: Springer, 2007, vol. 4625.
- [14] K. Kumatani, T. Arakawa, K. Yamamoto, J. W. McDonough, B. Raj, R. Singh, and I. Tashev, "Microphone array processing for distant speech recognition: Towards real-world deployment," in *Proc. APSIPA ASC*, 2012.
- [15] V. Manohar, P. Ghahremani, D. Povey, and S. Khudanpur, "A teacher-student learning approach for unsupervised domain adaptation of sequence-trained ASR models," in *2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, December 18-21, 2018*. IEEE, 2018, pp. 250–257.
- [16] J. Konecny, B. McMahan, and D. Ramage, "Federated Optimization: Distributed Optimization Beyond the Datacenter," *arXiv preprint arXiv:1511.03575v1*, 2015.
- [17] B. Jayaraman, L. Wang, D. Evans, and Q. Gu, "Distributed learning without distrust: Privacy-preserving empirical risk minimization," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 6343–6354.
- [18] D. Dimitriadis, K. Kumatani, R. Gmyr, Y. Gaur, and S. E. Eskimez, "A federated approach in training acoustic models," in *Proc. Interspeech*, 2020.
- [19] —, "Federated transfer learning with dynamic gradient aggregation," vol. ArXiv: 2008.02452, 2020.
- [20] A. Graves, "Sequence transduction with recurrent neural networks," *CoRR*, vol. abs/1211.3711, 2012.
- [21] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 577–585.
- [22] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *ICASSP*, 2016, pp. 4945–4949.
- [23] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *CoRR*, vol. abs/1508.01211, 2015.
- [24] A. Graves, S. Fernández, and F. Gomez, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *In Proceedings of the International Conference on Machine Learning, ICML 2006*, 2006, pp. 369–376.
- [25] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *Proc. Interspeech*, 2017, pp. 939–943.
- [26] E. Battenberg, J. Chen, R. Child, A. Coates, Y. Gaur, Y. Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, "Exploring neural transducers for end-to-end speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017, Okinawa, Japan, December 16-20, 2017*. IEEE, 2017, pp. 206–213.
- [27] C.-C. Chiu, W. Han, Y. Zhang, R. Pang, S. Kishchenko, P. Nguyen, A. Narayanan, H. Liao, S. Zhang, A. Kannan, R. Prabhavalkar, Z. Chen, T. N. Sainath, and Y. Wu, "A comparison of end-to-end models for long-form speech recognition," 2019, pp. 889–896.
- [28] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *ArXiv*, vol. abs/1503.02531, 2015.
- [29] Y. Kim and A. M. Rush, "Sequence-level knowledge distillation," *CoRR*, vol. abs/1606.07947, 2016.
- [30] Z. Meng, J. Li, Y. Gaur, and Y. Gong, "Domain adaptation via teacher-student learning for end-to-end speech recognition," in *ASRU*. IEEE, December 2019.
- [31] R. Caruana, "Multitask learning," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA 15213, 9 1997.
- [32] S. Ruder, "An overview of multi-task learning in deep neural networks," *ArXiv*, vol. abs/1706.05098, 2017.
- [33] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," *CoRR*, vol. abs/1506.03099, 2015.
- [34] Q. Xie, E. H. Hovy, M.-T. Luong, and Q. V. Le, "Self-training with noisy student improves imagenet classification," *ArXiv*, vol. abs/1911.04252, 2019.
- [35] "Sox - sound exchange, audio manipulation tool," Available at <http://sox.sourceforge.net/> (2020/04/28).
- [36] 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.
- [37] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," *Proc. ICASSP*, pp. 5206–5210, 2015.
- [38] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725.
- [39] N. Rossenbach, A. Zeyer, R. Schlüter, and H. Ney, "Generating synthetic audio data for attention-based speech recognition systems," in *Proc. ICASSP*, 2020, pp. 7069–7073.