INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Multi-task CTC Training with Auxiliary Feature Reconstruction for End-to-end Speech Recognition

Gakuto Kurata, Kartik Audhkhasi

IBM Research AI

gakuto@jp.ibm.com, kaudhkha@us.ibm.com

Abstract

We present a multi-task Connectionist Temporal Classification (CTC) training for end-to-end (E2E) automatic speech recognition with input feature reconstruction as an auxiliary task. Whereas the main task of E2E CTC training and the auxiliary reconstruction task share the encoder network, the auxiliary task tries to reconstruct the input feature from the encoded information. In addition to standard feature reconstruction, we distort the input feature only in the auxiliary reconstruction task, such as (1) swapping the former and latter parts of an utterance, or (2) using a part of an utterance by stripping the beginning or end parts. These distortions intentionally suppress long-span dependencies in the time domain, which avoids overfitting to the training data. We trained phone-based CTC and word-based CTC models with the proposed multi-task learning and demonstrated that it improves ASR accuracy on various test sets that are matched and unmatched with the training data.

Index Terms: End-to-end automatic speech recognition, Connectionist Temporal Classification, Long short-term memory, Multi-task learning

1. Introduction

End-to-end (E2E) Automatic Speech Recognition (ASR) using the Connectionist Temporal Classification (CTC) loss function [1] and/or the attention-based encoder-decoder architecture [2] has been gathering interest since it significantly simplifies the model training pipelines. E2E ASR only requires pairs of input feature sequences and output symbol sequences [3, 4], such as phones [5], characters [6], words [7, 8], or their combinations [9, 10].

To improve E2E ASR accuracy, some recent studies [10–15] have leveraged Multi-task Learning (MTL) [16, 17]. MTL typically shares lower hidden layers for all tasks, prepares task-specific output layers, and jointly optimizes all tasks, which reduces the risk of overfitting in each specific task [18].

Another trend triggered by E2E modeling, including fields other than ASR, is the optimization of separate models that interact with each other [19–21]. For example, speech recognition and speech synthesis were jointly optimized by feeding the output of one component as the input of the other component iteratively [19]. In neural machine translation (NMT), NMT models from language A to B (forward-translation) and from B to A (back-translation) were jointly optimized in *dual learning* [20]. An input sentence was first forward-translated from A to B, then back-translated from B to A, and lastly compared with this back-translated output sentence[1]. These works suggest that the output from one model should include sufficient information for the other related model to reconstruct the original input, and optimizing the models to perform this reconstruction well in-turn improves each model.

In this paper, we propose MTL that leverages input feature reconstruction as an auxiliary task. In E2E ASR systems, the input feature sequence is first encoded by an encoder network. The auxiliary task introduces a decoder network that tries to reconstruct the input feature from the encoded information. The idea of using feature reconstruction as an auxiliary task comes from the success of joint training of speech recognition and speech synthesis [19]. Since we are interested in improving speech recognition accuracy only, we simplify the speech synthesis part to generate features from an encoded vector instead of generating speech from text. We combine this input feature reconstruction with the E2E ASR model training. In addition, we distort the input feature in the auxiliary task through various methods, such as (1) swapping the former and latter parts of an utterance or (2) using a part of an utterance by removing the beginning or end parts of the utterance. Please note that these distortions cannot be applied in the main task of E2E ASR[2].

To confirm the advantage of the proposed MTL, we focus on phone-based CTC training and word-based CTC training as the main tasks. We used standard Switchboard English conversational telephone speech data for training. For evaluation, we used conversational telephone speech data as matched data and down-sampled broadcast news data as mismatched data.

This paper has three main contributions:

- It proposes MTL with an auxiliary feature reconstruction task to improve E2E ASR accuracy.
- It experiments with various ways of distorting the input features for the auxiliary task.
- It demonstrates ASR accuracy improvement on matched and especially unmatched test data.

2. Related Works

Here, we briefly explain CTC [1,25–27] and MTL in the context of ASR.

2.1. Connectionist Temporal Classification

Let $\boldsymbol{y}$ denote a length-L sequence of target output symbols. Let $\boldsymbol{X}$ denote acoustic feature vectors over T time steps. The conventional alignment-based Deep Neural Network/Hidden Markov Model (DNN/HMM) hybrid system training requires L to be equal to T [28]. The alignment-free CTC introduces an extra *blank* symbol ϕ that expands the length-L sequence $\boldsymbol{y}$ to

[1]The idea of back-translation has been used for data augmentation for NMT [22, 23] and ASR [24].

[2]For example, if we remove a part of the input feature sequence in the E2E ASR training, we need to remove the corresponding part of an output symbol sequence. However, since there is no alignment between the input feature sequence and the output symbol sequence, we cannot identify the part of the output sequence that corresponds to the removed part in the input sequence.

http://dx.doi.org/10.21437/Interspeech.2019-1710

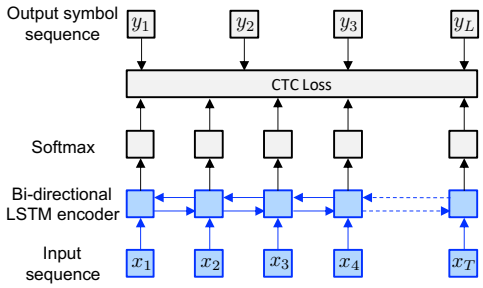


Figure 1: *Bi-directional LSTM CTC training between input sequence of* $x_1, x_2, \cdots, x_T$ *and output symbol sequence* $y_1, y_2, \cdots, y_L$.

a set of length-T sequences $\Phi(\boldsymbol{y})$. Each sequence $\hat{\boldsymbol{y}} \in \Phi(\boldsymbol{y})$ is one of the *CTC alignments* between $\boldsymbol{X}$ and $\boldsymbol{y}$. The CTC loss is defined as the summation of symbol posterior probabilities over all possible CTC alignments:

$$\mathcal{L}_{\text{CTC}} = -\sum_{\hat{\boldsymbol{y}} \in \Phi(\boldsymbol{y})} P(\hat{\boldsymbol{y}}|\boldsymbol{X}) = -\sum_{\hat{\boldsymbol{y}} \in \Phi(\boldsymbol{y})} \prod_{t=1}^{T} P(\hat{y}_t|\boldsymbol{x_t}) \,.$$

Typically, Bi-directional LSTM (BiLSTM) is used for encoding the input sequence and the CTC loss is computed on the basis of the encoded results, as shown in Figure 1. More complex networks, such as pyramidal BiLSTM [6] and a combination of VGG [29] and BiLSTM, can be used for the encoder [30].

We use phones and words as target output symbols, namely *phone CTC* and *word CTC* respectively, for evaluation in Section 4.

2.2. Multi-task Learning

In MTL for E2E ASR, all or some of the encoding layers are typically shared across the main and auxiliary tasks, and task-specific output layers are prepared. One possible combination of tasks is to use the same type of architecture with different granularity of output symbols. For example, combinations of word CTC with phone CTC [10], subword CTC with phone CTC [11], and a character-based decoder with a phone-based decoder [12] have been used. The other type of combination is to use different types of architectures, like CTC with an attention-based encoder-decoder [13, 14, 30], and CTC with a segmental conditional random field (CRF) [15]. Typical optimization strategies are (1) to minimize weighted summation of losses from multiple tasks, (2) to switch multiple tasks with minimizing respective losses, or (3) to start the training of the main task from the shared layers initialized by the training of the auxiliary task.

3. Multi-task CTC Training with Feature Reconstruction

To improve ASR accuracy, we propose a new MTL that uses feature reconstruction as an auxiliary task as shown in Figure 2(a). We have a common BiLSTM encoder and prepare two task-specific output layers. The main task is CTC training where target symbols can be phones, characters, subwords, or words, as shown in the left branch. In the auxiliary task in the right branch, we prepare an additional BiLSTM decoder operating on the output from the BiLSTM encoder. Then we minimize Means Square Error (MSE) loss between the output of the BiLSTM decoder and the reconstruction target. The reconstruction target can be identical to the input feature, which we call *full* reconstruction target. Considering that the input feature for ASR usually contains delta and double delta feature that can be derived from the static feature [31], another option can be using only the static feature as the reconstruction target, which we call *static-only* reconstruction target.

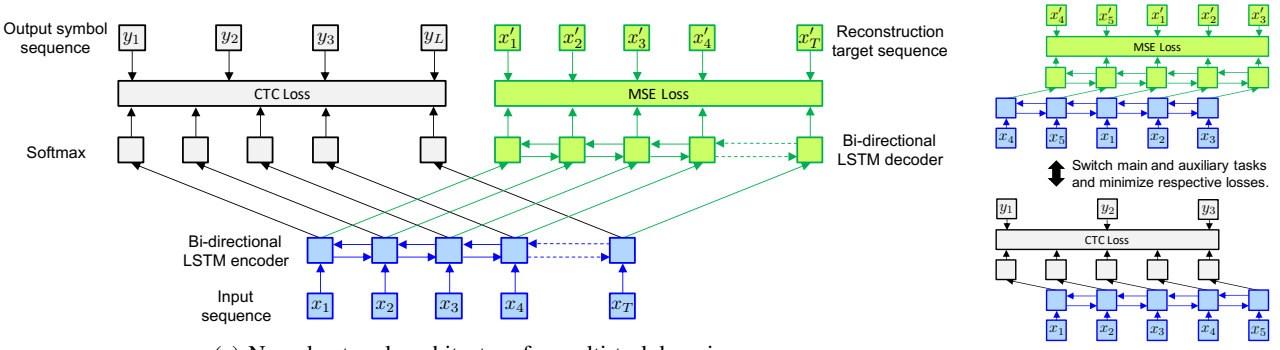
The idea of using feature reconstruction as an auxiliary task comes from the success of joint training of speech recognition and speech synthesis [19]. Since we are interested in improving speech recognition accuracy only, we simplify the speech synthesis part to generate features from an encoded vector instead of generating speech from text.

The advantage of using feature reconstruction as an auxiliary task is that it provides us with the flexibility to distort features in any one of a multitude of ways. In the main task of CTC training, a symbol sequence is paired with a whole input feature sequence and thus the whole input sequence needs to be used in training. In the feature reconstruction task, since the input feature and the reconstruction targets are basically the same, we can make the pair of the input feature and the reconstruction target from any arbitrary part of the input feature. As shown in Figure 3 using an example input feature $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_5}\}$ and corresponding reconstruction target $\boldsymbol{X'} = \{\boldsymbol{x'_1}, \boldsymbol{x'_2}, \cdots, \boldsymbol{x'_5}\}$[3], we use three types of feature reconstruction, namely, *standard*, *swap*, and *strip* feature reconstruction. In the standard feature reconstruction, a whole input sequence is used as the input and reconstruction target. In the swap feature reconstruction, the input sequence is split into two parts at a randomly selected position, and then the former and latter parts are swapped. The reconstruction targets are also swapped at the same position accordingly. In the strip feature reconstruction, the beginning part before or end part after a randomly selected position is stripped from the input sequence. The same parts are also stripped from the reconstruction target. By using swap and strip feature reconstruction, we can expect that the local dependencies in time domain can still be learned, while long-spanning dependencies in the time domain can be discarded, which results in avoiding overfitting to the training data. Especially, in the swap feature reconstruction, the hidden state of the BiLSTM encoder is completely "corrupted" at the position of swapping, and thus stronger regularization can be expected.

Please note that even when the input feature is distorted in the auxiliary task, the original input sequence without distortion and the corresponding output symbol sequence are used in the main CTC task. To easily realize this in MTL, we switched two tasks and minimized respective losses as shown in Figure 2(b), which is an example of using the swap feature reconstruction in the auxiliary task. More specifically, we randomly picked a mini-batch with a certain ratio. For the picked mini-batch, we first minimized the MSE loss for feature reconstruction by updating all parameters in the BiLSTM decoder and the shared BiLSTM encoder. Then we minimized the CTC loss by updating the parameters of the linear layer before the softmax activation and the shared BiLSTM encoder. For the other mini-batches, we only minimized the CTC loss in the same way.

A previous work [32] used feature reconstruction for acoustic modeling in the DNN/HMM hybrid system on the TIMIT

[3] If we use the full reconstruction target, $\boldsymbol{X} = \boldsymbol{X'}$.



(a) Neural network architecture for multi-task learning.

(b) Training procedure.

Figure 2: *Proposed multi-task CTC training. (a) Auxiliary task tries to reconstruct the input sequence. Note that* $x'_1, x'_2, \cdots, x'_T$ *indicates the full or static-only reconstruction target derived from the input sequence* $x_1, x_2, \cdots, x_T$*. (b) During training, we randomly picked a mini-batch with a certain ratio. For the picked mini-batch, we first minimized the MSE loss for feature reconstruction and then minimized the CTC loss. For other mini-batches, we only minimized the CTC loss. Note that this is an example of using the swap feature reconstruction in the auxiliary task.*

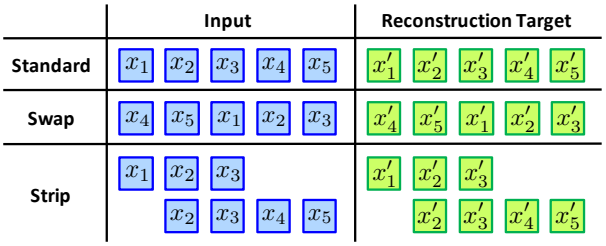


Figure 3: *Summary of the standard, swap, and strip feature reconstruction. The original input sequence is* $x_1, x_2, \cdots, x_5$ *as used in the standard reconstruction. The former and latter parts are swapped in the swap reconstruction and beginning and end parts are stripped in the strip reconstruction. Please note that the original input sequence is used in the main task of CTC training regardless of feature reconstruction types.*

corpus [33]. To the best of our knowledge, this is the first attempt to use feature reconstruction for E2E ASR for large vocabulary continuous speech recognition tasks and to enhance feature reconstruction by distorting the input features.

4. Experiments

We conducted experiments for phone CTC models and word CTC models as the main tasks to confirm the advantage of the auxiliary feature reconstruction task in MTL. We used 262 hours of segmented speech from the standard 300-hour Switchboard-1 English conversational telephone speech as the training data set. In all the tasks of phone CTC training, word CTC training, and feature reconstruction tasks, we used 40-dimensional logMel filterbank energies, their delta, and double-delta coefficients with frame stacking having a decimation rate of 2 [34]. We did not use any speaker-dependent feature transformations.

For phone CTC models, we used 44 phones from the Switchboard pronunciation lexicon [35] and the blank symbol. For decoding, we trained a 4-gram language model with 24M words from the Switchboard and Fisher transcripts with a vocabulary size of 30K. We constructed a CTC decoding graph similar to the one in [36]. For neural network (NN) architecture, we stacked 6 BiLSTM layers (BiLSTM encoder) with 512 units each in the forward and backward layers and a fully-connected linear layer of 1024×45, followed by a softmax activation function. All NN parameters were initialized to samples of a uniform distribution over $(-\epsilon, \epsilon)$, where ϵ is the inverse square root of the input vector size.

For word CTC models, we selected words with at least 5 occurrences in the training data [7, 10]. This resulted in an output layer with 10,175 words and the blank symbol. We used the same 6 BiLSTM layers, added 1 fully-connected linear layer with 256 units to reduce computation [37], and put a fully-connected linear layer of $256 \times 10{,}176$, followed by a softmax activation function. For better convergence, we initialized the BiLSTM encoder part with the trained phone CTC model[4] [7, 10]. Other parameters were initialized in similar fashion as the phone CTC models. For decoding, we performed a simple peak-picking over the output word posterior distribution, and removed repetitions and blank symbols.

For the auxiliary feature reconstruction task, we stacked 2 BiLSTM layers (BiLSTM decoder) with 512 units each in the forward and backward layers and a fully-connected linear layer. All parameters were initialized similarly to in the phone CTC models.

In the baseline phone CTC and word CTC model training, we minimized the phone CTC loss and word CTC loss, respectively. In MTL with feature reconstruction, we randomly picked mini-batches with a ratio in $\{0.1, 0.2, 0.3\}$, and minimized both the CTC loss and the MSE loss of feature reconstruction for the picked mini-batches shown in Figure 2(b).

We trained all models for 20 epochs and used stochastic gradient descent with Nesterov momentum of 0.9 and a learning rate starting from 0.02 and annealing at $\sqrt{0.5}$ per-epoch after the 10th epoch. We set the batch size to 128.

For evaluation, we used the Switchboard (SWB) and Call-Home (CH) subsets of the NIST Hub5 2000 evaluation data set. Considering that the training data consists of SWB-like data, testing on the CH test set is a mismatched scenario for our system [35]. For a further mismatched test set, we used the down-

[4]Since we had confirmed the effect of initialization by other CTC models, we did not conduct pretraining by the auxiliary task in MTL and instead switched the main and auxiliary tasks during training.

Table 1: *Word Error Rates (WERs) by phone CTC models for Switchboard (SWB), CallHome (CH), and Broadcast News (BN) test sets with using full and static-only reconstruction targets in the auxiliary reconstruction task. Standard feature reconstruction was used in all cases.*

	SWB	CH	BN	Avg.
Baseline (w/o MTL)	11.8	22.0	31.4	21.7
w/ MTL				
Full (240 dimensions)	11.8	21.3	31.2	21.4
Static-only (80 dimensions)	11.5	21.6	30.8	**21.3**

Table 2: *Word Error Rates (WERs) by phone CTC models for Switchboard (SWB), CallHome (CH), and Broadcast News (BN) test sets with using standard, swap, and strip feature reconstruction in the auxiliary reconstruction task.*

	SWB	CH	BN	Avg.
Baseline (w/o MTL)	11.8	22.0	31.4	21.7
w/ MTL				
Standard	11.5	21.6	30.8	21.3
Swap	11.5	21.2	30.7	**21.1**
Strip	11.6	21.2	30.8	21.2

Table 3: *Word Error Rates (WERs) by word CTC models for Switchboard (SWB), CallHome (CH), and Broadcast News (BN) test sets with using standard, swap, and strip feature reconstruction in the auxiliary reconstruction task.*

	SWB	CH	BN	Avg.
Baseline (w/o MTL)	15.6	26.0	42.1	27.9
w/ MTL				
Standard	15.9	25.5	41.9	27.8
Swap	15.7	25.4	41.2	**27.4**
Strip	15.5	25.8	41.9	27.7

sampled DARPA EARS dev04f broadcast news data (BN). We report the Word Error Rates (WERs) for these evaluation data.

4.1. Reconstruction Target

First, we compared full and static-only reconstruction targets. The input feature has 240 dimensions consisting of 2 stacking frames of 40-dimensional logMel filterbank energies, their delta, and double delta coefficients. Thus, the full reconstruction target has 240 dimensions, whereas the static-only reconstruction target has 80 dimensions. We trained phone CTC models with MTL using different reconstruction targets. Table 1 shows WERs on SWB, CH, and BN test sets. For both reconstruction targets, we found WER improvement in most cases[5]. Since the static-only target obtained improvements for all test sets and has smaller dimensions, we use the static-only target in the following experiments.

[5]As for the switching ratio to the auxiliary task in MTL, 0.1 was the best in all cases. This was same with the following experiments in Section 4.2 and Section 4.3.

4.2. Phone CTC

Then we trained phone CTC models with MTL. Here, we tried three types of feature reconstruction with using the static-only targets. Table 2 shows WERs on SWB, CH, and BN test sets. For all cases, WER was improved compared with the baseline without MTL. Looking at the average WER, the swap and strip feature reconstruction resulted in better WER than the standard feature reconstruction, which indicates the advantage of distorting input features. As an overall trend, the improvement was bigger on CH and BN than SWB. The auxiliary task mitigated overfitting to the training data and thus we obtained larger improvement on the CH and BN test sets that are mismatched with the training data.

4.3. Word CTC

Finally, we trained word CTC models with MTL. Please note that BiLSTM encoder parts were initialized with the baseline phone CTC model trained in Section 4.2. Table 3 shows WERs on SWB, CH, and BN test sets. Looking at the average WER, the swap and strip feature reconstruction resulted in better WER than the standard feature reconstruction, which indicates the advantage of distorting input features.

Comparing with the baseline, we confirmed WER improvement on CH and BN test sets by MTL. We saw a minor degradation on a few SWB test cases where training and test data are completely matched in terms of acoustics and linguistics. The main task of word CTC model inherently learns a Language Model (LM) and an external LM was not used in decoding in our experiments. The auxiliary task of feature reconstruction does not consider linguistics and thus could hurt the implicit LM. Applying rescoring with an external LM might mitigate this problem.

5. Conclusion

In this paper, we proposed a multi-task CTC training by using feature reconstruction as an auxiliary task, inspired by the success of joint training of speech recognition with synthesis and forward and backward NMT. We confirmed that our proposed MTL improves ASR accuracy for phone CTC models and word CTC models, especially on test sets that are unmatched with training data. In addition, we distorted input feature sequences only in the auxiliary feature reconstruction tasks to intentionally suppress dependencies spanning long in the time domain and demonstrated its effect. We used E2E CTC models as a main task, but feature reconstruction can be combined with other E2E modeling such as attention-based encoder-decoder, which is our future work.

6. Acknowledgments

We thank Dr. George Saon of IBM T. J. Watson Research Center for his valuable suggestions.

7. References

[1] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006, pp. 369–376.

[2] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. ICASSP*, 2016, pp. 4945–4949.

[3] 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.

[4] A. Hannun, "Sequence modeling with CTC," *Distill*, 2017, https://distill.pub/2017/ctc.

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

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

[7] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct acoustics-to-word models for English conversational speech recognition," in *Proc. INTERSPEECH*, 2017, pp. 959–963.

[8] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," in *Proc. INTERSPEECH*, 2017, pp. 3707–3711.

[9] H. Liu, Z. Zhu, X. Li, and S. Satheesh, "Gram-CTC: Automatic unit selection and target decomposition for sequence labelling," in *Proc. ICML*, 2017, pp. 2188–2197.

[10] K. Audhkhasi, B. Kingsbury, B. Ramabhadran, G. Saon, and M. Picheny, "Building competitive direct acoustics-to-word models for English conversational speech recognition," in *Proc. ICASSP*, 2018, pp. 4759–4763.

[11] K. Krishna, S. Toshniwal, and K. Livescu, "Hierarchical multitask learning for CTC-based speech recognition," *arXiv preprint arXiv:1807.06234*, 2018.

[12] S. Toshniwal, H. Tang, L. Lu, and K. Livescu, "Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition," in *Proc. INTERSPEECH*, 2017, pp. 3532–3536.

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

[14] T. Hori, S. Watanabe, and J. Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *Proc. ACL*, vol. 1, 2017, pp. 518–529.

[15] L. Lu, L. Kong, C. Dyer, and N. A. Smith, "Multitask learning with CTC and segmental CRF for speech recognition," *arXiv preprint arXiv:1702.06378*, 2017.

[16] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[17] Y. Zhang and Q. Yang, "A survey on multi-task learning," *arXiv preprint arXiv:1707.08114*, 2017.

[18] J. Baxter, "A Bayesian/information theoretic model of learning to learn via multiple task sampling," *Machine learning*, vol. 28, no. 1, pp. 7–39, 1997.

[19] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *Proc. ASRU*, 2017, pp. 301–308.

[20] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T. Liu, and W.-Y. Ma, "Dual learning for machine translation," in *Proc. NIPS*, 2016, pp. 820–828.

[21] H. Hassan Awadalla, A. Aue, C. Chen, V. Chowdhary, J. Clark, C. Federmann, X. Huang, M. Junczys-Dowmunt, W. Lewis, M. Li, S. Liu, T.-Y. Liu, R. Luo, A. Menezes, T. Qin, F. Seide, X. Tan, F. Tian, L. Wu, S. Wu, Y. Xia, D. Zhang, Z. Zhang, and M. Zhou, "Achieving human parity on automatic Chinese to English news translation," *arXiv preprint arXiv:1803.05567*, 2018.

[22] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," *arXiv preprint arXiv:1511.06709*, 2015.

[23] M. Artetxe, G. Labaka, E. Agirre, and K. Cho, "Unsupervised neural machine translation," *arXiv preprint arXiv:1710.11041*, 2017.

[24] T. Hayashi, S. Watanabe, Y. Zhang, T. Toda, T. Hori, R. Astudillo, and K. Takeda, "Back-translation-style data augmentation for end-to-end ASR," *arXiv preprint arXiv:1807.10893*, 2018.

[25] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, 2013, pp. 6645–6649.

[26] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[27] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. ICML*, 2014, pp. 1764–1772.

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

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[30] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint CTC-Attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in *Proc. INTERSPEECH*, 2017, pp. 949–953.

[31] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 29, no. 2, pp. 254–272, 1981.

[32] M.-H. Yang, H.-S. Lee, Y.-D. Lu, K.-Y. Chen, Y. Tsao, B. Chen, and H.-M. Wang, "Discriminative autoencoders for acoustic modeling," in *Proc. INTERSPEECH*, 2017, pp. 3557–3561.

[33] J. W. Lyons, "DARPA TIMIT acoustic-phonetic continuous speech corpus," *National Institute of Standards and Technology*, 1993.

[34] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," *arXiv preprint arXiv:1507.06947*, 2015.

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

[36] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *Proc. ASRU*, 2015, pp. 167–174.

[37] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *Proc. ICASSP*, 2013, pp. 6655–6659.