HIGH-ACCURACY AND LOW-LATENCY SPEECH RECOGNITION WITH TWO-HEAD CONTEXTUAL LAYER TRAJECTORY LSTM MODEL

Jinyu Li, Rui Zhao, Eric Sun, Jeremy H. M. Wong, Amit Das, Zhong Meng, and Yifan Gong

Microsoft Speech and Language Group

ABSTRACT

While the community keeps promoting end-to-end models over conventional hybrid models, which usually are long short-term memory (LSTM) models trained with a cross entropy criterion followed by a sequence discriminative training criterion, we argue that such conventional hybrid models can still be significantly improved. In this paper, we detail our recent efforts to improve conventional hybrid LSTM acoustic models for high-accuracy and low-latency automatic speech recognition. To achieve high accuracy, we use a contextual layer trajectory LSTM (cltLSTM), which decouples the temporal modeling and target classification tasks, and incorporates future context frames to get more information for accurate acoustic modeling. We further improve the training strategy with sequence-level teacher-student learning. To obtain low latency, we design a two-head cltLSTM, in which one head has zero latency and the other head has a small latency, compared to an LSTM. When trained with Microsoft's 65 thousand hours of anonymized training data and evaluated with test sets with 1.8 million words, the proposed two-head cltLSTM model with the proposed training strategy yields a 28.2% relative WER reduction over the conventional LSTM acoustic model, with a similar perceived latency.

***Index Terms*—** LSTM, teacher-student learning, automatic speech recognition, latency

1. INTRODUCTION

There is a clear trend of recent development of end-to-end (E2E) modeling [1, 2, 3, 4, 5, 6, 7, 8]. However, it is still difficult for E2E models to replace the popular hybrid systems in industry, where flexibility of modeling is important. Furthermore, latency or even streaming is always a concern for E2E models [9, 10, 11], while hybrid systems usually have low latency. Hence, hybrid systems continue to dominate in industry, and advancing the hybrid systems is still an important research topic. Conventional hybrid systems are usually trained, first with a cross entropy (CE) criterion, followed by a sequence discriminative criterion, such as maximum mutual information (MMI) [12] and state-level minimum Bayes' risk (sMBR) [13]. Such hybrid systems [14] were used as strong baselines to justify the accuracy advantage of E2E models [4]. However, we argue that such baseline hybrid models can still be improved significantly, retaining their competitiveness with the improvements in E2E modeling. In this study, we detail our efforts to develop high-accuracy and low-latency hybrid models.

To achieve high accuracy, we work in two directions. The first direction is to have an advanced model structure. There have been many works to improve the dominant long short-term memory (LSTM) [15, 16] model structures for acoustic modeling, such as highway LSTM [17], residual LSTM [18, 19], time-frequency LSTM [20, 21, 22], and grid LSTM [23, 24]. In this paper, we use a new model called context layer trajectory LSTM [25] to significantly improve the model accuracy. The advantage of cltLSTM comes from 1) decoupling the task of temporal modeling and target classification using time-LSTM and depth-LSTM respectively; and 2) exploring context frames to incorporate future information. The second direction is to improve upon the hybrid model training strategy. Specifically, on top of sequence discriminative training, we further improve the model accuracy by performing sequence-level teacher-student (T/S) learning [26, 27] toward a strong ensemble teacher.

To obtain low latency, we propose a two-head cltLSTM structure that has two softmax output layers with shared time-LSTM units but different depth-LSTM branches. One depth-LSTM branch does not use any future context frames, and hence has zero additional latency. This is used for first-pass decoding. The other depth-LSTM branch incorporates future context frames for high-accuracy modeling, and is used for second-pass decoding. Such a design allows for an ASR model with high accuracy and low perceived latency.

Using 65 thousand hours of Microsoft anonymized production training data with personally identifiable information removed, our proposed model together with the proposed better training strategy achieved a 28.2% relative word error rate (WER) reduction from the conventional MMI-trained LSTM model, while having almost the same low perceived latency.

2. IMPROVEMENTS TO THE LSTM ACOUSTIC MODEL

For a multi-layer LSTM, we define the hidden output of the lth layer at time t as

$$h_t^l = \text{LSTM}\left(h_{t-1}^l, x_t^l\right), \tag{1}$$

where the LSTM() function is the standard LSTM unit with a projection layer [16]. Here, h_t^l is the hidden output of the lth layer at time t and x_t^l is the input vector for the lth layer with

$$x_t^l = \begin{cases} h_t^{l-1}, & \text{if } l > 1 \\ s_t, & \text{if } l = 1 \end{cases}, \tag{2}$$

where s_t is the speech spectrum input at time step t. Next, several models used in this study will be introduced

2.1. Layer trajectory LSTM

The standard LSTM units used in recurrent neural networks serve two very different purposes at the same time, namely temporal modeling and target classification. In [28], the layer trajectory LSTM (ltLSTM) was introduced to decouple the tasks of temporal modeling and target classification, using time-LSTM and depth-LSTM units respectively. As is reported in [28], the ltLSTM significantly

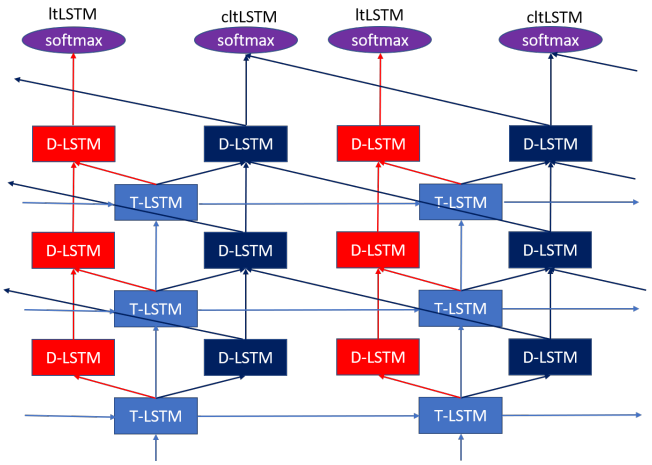

Fig. 1. Diagram of the two-head cltLSTM. The time-LSTM (T-LSTM) units are shared, while the ltLSTM head and cltLSTM head have separate depth-LSTM (D-LSTM) units.

outperformed the LSTM or residual LSTM. The time-LSTM formulation is the same as Eq. (1) while the depth-LSTM formulation is

$$g_t^l = \text{LSTM}\left(h_t^l, g_t^{l-1}\right), \tag{3}$$

where g_t^l is the output of the depth-LSTM at layer l and time t.

2.2. Contextual layer trajectory LSTM

In [25], the contextual layer trajectory LSTM (cltLSTM) was proposed to further improve the performance of ltLSTM models by using context frames to capture future information. In the cltLSTM, g_t^{l-1} in Eq. (3) is replaced by the look-ahead embedding vector ζ_t^{l-1} in order to incorporate future context information,

$$g_t^l = \text{LSTM}(h_t^l, \zeta_t^{l-1}), \tag{4}$$

while h_t^l is still computed with Eq. (1). The embedding vector is computed by transforming the outputs of the depth-LSTM from current and future frames as

$$\zeta_t^{l-1} = \sum_{\delta=0}^{\tau} G_\delta^{l-1} g_{t+\delta}^{l-1}, \tag{5}$$

where G_δ^{l-1} denotes the weight matrix applied to the depth-LSTM output $g_{t+\delta}^{l-1}$. An L layer cltLSTM with τ future context frames at each layer has a total of $N = L \times \tau$ look-ahead frames.

2.3. Two-head contextual layer trajectory LSTM

Ideally, industrial speech services should have both high accuracy and low latency. The latter is usually overlooked in many studies, but is very important to the user's experience. A high latency gives the user the impression that the system is not responding, even though it may have a high accuracy. However, these two requirements sometimes conflict with each other, especially when the system explores future information (e.g., cltLSTM) to boost its modeling accuracy. In this study, we propose a two-head cltLSTM shown in Figure 1 to build ASR systems with both high accuracy and low latency.

The two-head cltLSTM has an ltLSTM head and a cltLSTM head, which share the same time-LSTM units, but have their own respective depth-LSTM units. The ltLSTM head, without any access to future context frames, provides low-latency decoding, while the cltLSTM head provides high-accuracy recognition results, thanks to the future look-ahead. The training steps are

1. Train a cltLSTM model with the best training recipe, which will be described in Section 3.
2. Take the time-LSTM layers out from the well-trained cltLSTM, and then build an ltLSTM with Eq. (3), by adding the depth-LSTM and softmax output. With a new softmax layer, we form an ltLSTM head for the two-head cltLSTM.
3. Train the new ltLSTM model without updating the time-LSTM layers.

The runtime steps of the two-head cltLSTM are as follows.

1. Start the first-pass decoding using the ltLSTM head. For every frame, store the time-LSTM hidden vectors h_t^l at all layers, calculated with Eq. (1).
2. Start the second-pass decoding using the cltLSTM head after the first pass-decoding processes N acoustic frames, where N is the number of contextual frames that the cltLSTM model needs to look ahead.
3. Replace the output results of the first-pass decoding with the output results of the second-pass decoding when they differ.

The first-pass decoder gives users almost 0 latency, while the second-pass refines the results later on. Since the absolute WER difference between cltLSTM and ltLSTM is small, the replacement of results at runtime step 3 only happens occasionally. Furthermore, a small N ensures that the replacement latency is not too long. Therefore, the whole system has a high accuracy and a low perceived latency.

3. BOOSTING ACCURACY WITH SEQUENCE-LEVEL TEACHER-STUDENT LEARNING

A conventional hybrid model training recipe is to first train toward the frame-level CE criterion, followed by a sequence discriminative criterion such as MMI [12], sMBR [13] or the recently proposed word-level edit-based minimum Bayes' risk (EMBR) [29]. In this study, we further boost the accuracy by using teacher-student (T/S) learning to train the model to emulate a strong ensemble.

A popular T/S learning strategy was first proposed by Li et al. in 2014 [30] and later by Hinton et al. in 2015 [31] as knowledge distillation. The training criterion is to minimize the Kullback-Leibler (KL) divergence between the frame-level output posterior distributions of the teacher and student networks. Although it has achieved much success in deep learning, it does not take into account the sequential nature of speech. Instead, [26] proposed to do T/S learning at the sequence level by minimizing the KL divergence between hypothesis sequence posteriors,

$$\mathcal{L}_{\text{seq-TS}}\left(\theta_{\text{S}}\right) = -\sum_{H} P\left(H|X;\theta_{\text{T}}\right) \log P\left(H|X;\theta_{\text{S}}\right), \tag{6}$$

where X is an input sequence, θ_{T} is the teacher, θ_{S} is the student, and H is a hypothesis, which may be expressed as a sequence of words, sub-word units, or states [27]. This study considers state sequences.

Strong teacher targets may be obtained by combining an ensemble of multiple teachers together, such that

$$P\left(H|X;\theta_{\text{T}}\right) = \sum_{k} \alpha_k P\left(H|X;\theta_{\text{T}_k}\right), \tag{7}$$

where α_k is the combination weight for the kth teacher, θ_{T_k}, satisfying $\sum_k \alpha_k = 1$ and $\alpha_k \geq 0$. The contribution to the T/S gradient from the teachers can be obtained by performing a separate forward-backward operation over each of the teachers' lattices, which represent each teachers' hypotheses [27]. This is computationally expensive, especially when using a large amount of training data. To reduce this cost, we instead combine the teachers at the frame level,

$$P\left(q_t|x_t;\theta_{\mathrm{T}}\right) = \sum_k \alpha_k P\left(q_t|x_t;\theta_{\mathrm{T}_k}\right), \tag{8}$$

where q_t is the senone state at time t. The combined frame posteriors can then be used with a single lattice for all combined teachers, over which a single forward-backward instance can be performed during the gradient computation. A similar frame-level teacher combination has previously been used in [32] within a lattice-free framework. In this paper, we investigate its use within a lattice-based framework.

In [26], a unigram word-level language model (LM) was used to generate lattices for sequence T/S learning. This followed the justification used in MMI [12], that a weak LM should be used for lattice generation to allow for a diverse representation of hypotheses. However, in the context of sequence T/S learning, we argue that it is better to generate the training lattices using a strong LM, so that the targets are as representative as possible of hypotheses generated at runtime, to allow the student to better emulate the teacher's runtime behavior. In this paper, we will compare the use of different ngram LMs for lattice generation for sequence T/S learning within a lattice-based framework.

The proposed hybrid model training recipe is as follows.

1. Train the model of interest, A, and another strong model, B, with the CE, MMI, and then EMBR criteria in order.
2. Use a frame-level combination of the EMBR versions of A and B as the teacher ensemble with Eq. (8).
3. Initialise the student as the MMI version of A and perform sequence T/S learning toward the ensemble with Eq. (6).

4. EXPERIMENTS

In this section, we evaluate the effectiveness of the proposed models. All models were trained with 65 thousand (K) hours of transcribed data from a variety of Microsoft products. The test set covers 13 application scenarios such as Cortana and far-field speech, using a total of 1.8 million (M) words. All the training and test data are anonymized data with personally identifiable information removed. We report the WER averaged over all test scenarios.

All of the models have 6 uni-directional LSTM layers with 1024 hidden units, and the output dimension is reduced to 512 using a linear projection layer [16]. The input feature is 80-dimension log Mel filter bank for every 10 milliseconds (ms) of speech. The softmax layer has 9404 nodes to model the senone labels. The target senone label is delayed by 50 ms, similarly to [16]. We applied frame skipping by a factor of 2 [33] to reduce the runtime cost, which corresponds to 20 ms per frame. Runtime decoding is performed using a 5-gram language model with around 100 M ngrams.

4.1. Training strategy

We first explore how to build a high-accuracy hybrid model. We use the cltLSTM with a 24-frame look-ahead ($\tau = 4$ and $L = 6$) described in Section 2.2, denoted here as cltLSTM-24. Since we applied frame skipping by a factor of 2 [33], this 24-frame look-ahead introduces 480 ms of latency compared to a standard LSTM model. There are 63 M parameters in this model. As shown in Table 1, the CE-trained cltLSTM-24 model has a WER of 11.15% averaged over the whole test sets. MMI training reduces the WER to 10.36%. EMBR training from the CE seed model yields a similar WER to the MMI model. Starting from the MMI model, additional EMBR training further reduces the WER to 10.18%.

Table 1. Average WERs of cltLSTM-24 and lt-lc-BLSTM on 13 test sets with 1.8 M words.

	cltLSTM-24	lt-lc-BLSTM	combine
CE	11.15	10.11	-
CE $\rightarrow$ MMI	10.36	9.45	-
CE $\rightarrow$ EMBR	10.38	-	-
MMI $\rightarrow$ EMBR	10.18	9.24	8.62
MMI $\rightarrow$ T/S (bigram)	9.63	-	-
MMI $\rightarrow$ T/S (5-gram)	9.34	8.92	-

Next, we further improve the model with sequence T/S learning toward an ensemble, as is described in Section 3. The ensemble combines the cltLSTM-24 with a layer-trajectory latency-control bi-directional LSTM (lt-lc-BLSTM) [34]. Similarly to the layer-trajectory LSTM models used in this study, the lt-lc-BLSTM also decouples the tasks of temporal modeling and target classification with time-BLSTM and depth-LSTM units, respectively. It also has 6 hidden layers. At each layer, the forward and backward LSTMs in the time-BLSTMs use 800 hidden units and then are projected to 400 by a linear projection layer. The depth-LSTMs uses 800 hidden units that are also then projected to 400. It totally has 102 M parameters, and has up to 800 ms of latency with the latency-control implementation [17]. The size and latency of this lt-lc-BLSTM do not satisfy the requirements for most Microsoft application scenarios. However, it is a strong model that can be combined with the cltLSTM-24 as a teacher ensemble for sequence T/S learning. From Table 1, this lt-lc-BLSTM has WERs of 10.11%, 9.45%, and 9.24% for CE, MMI, and EMBR training, respectively. Using Eq. (8), an equally weighted frame-level combination of the EMBR cltLSTM-24 and lt-lc-BLSTM models yields a teacher ensemble WER of 8.62%.

The cltLSTM-24 MMI model was used as the initial student. We experimented on generating the lattices that represent the T/S hypotheses using either a bigram or 5-gram LM, with the acoustic scores from the initial cltLSTM-24 MMI model. The lattices were acoustically re-scored with the frame-level combined teachers' acoustic scores to compute the contribution to the T/S gradient from the teachers. The 5-gram LM was the same as that used at runtime. The cltLSTM-24 student trained using the bigram and 5-gram lattices achieved WERs of 9.63% and 9.34%, respectively. This shows that sequence T/S learning performs better when the hypotheses are represented with the stronger LM that is used during runtime. The final cltLSTM-24 student yields an 9.8% relative WER improvement over its MMI initialization. As a comparison, an lt-lc-BLSTM student, was also trained toward the same ensemble, yielding a WER of 8.92%. This is better than the cltLSTM-24 student, but has a model size and latency that is too expensive for runtime application.

4.2. Models with different look-ahead frames

The next experiment used the same CE$\rightarrow$MMI$\rightarrow$T/S training strategy, using the same teacher ensemble as Table 1. The students used were LSTM, ltLSTM (zero frame look-ahead), and cltLSTM-6 and cltSLTM-12, which have 6 ($\tau = 1$) and 12 ($\tau = 2$) frames look-

Table 2. Average WERs of all models on 13 test sets with 1.8 M words.

models	CE	MMI	sequence T/S
LSTM	14.75	13.01	11.49
ltLSTM	12.41	11.06	10.10
cltLSTM-6	11.97	10.67	9.66
cltLSTM-12	11.38	10.32	9.34
cltLSTM-24	11.15	10.36	9.34
two-head cltLSTM-12			
first head	12.24	11.33	10.03
second head	11.38	10.32	9.34

Table 3. Latency and number of parameters of all models.

	latency compared to LSTM (ms)	number of parameters (M)
LSTM	0	31
ltLSTM	0	57
cltLSTM-6	120	58
cltLSTM-12	240	60
cltLSTM-24	480	63
two-head cltLSTM-12		
first head	0	57
second head	240	34

aheads. The WERs of these models are reported in Table 2. The latency and the number of parameters are shown in Table 3. For all models, consistent gains are observed from CE to MMI, and then to sequence T/S. The relative WER reductions from sequence T/S over MMI for LSTM, ltLSTM, cltLSTM-6, cltLSTM-12, and cltLSTM-24 are 11.7%, 8.7%, 9.5%, 9.5%, and 9.8%, respectively.

Comparing the final T/S models, the 12.1% relative WER reduction of ltLSTM over LSTM shows the benefit of decoupling the temporal modeling and target classification tasks. The 4.4% relative WER reduction of cltLSTM-6 over ltLSTM indicates the effectiveness of incorporating future information. cltLSTM-12 further reduces the WER by 3.3% relative over cltLSTM-6, while cltLSTM-24 does not yield further gains. The frame skipping [33] used during runtime means that every frame spans 20 ms. Therefore, ltLSTM, cltLSTM-6, cltLSTM-12, and cltLSTM-24 respectively have 0, 120, 240, and 480 ms greater latencies than an LSTM. It is therefore better to use cltLSTM-12 rather than cltLSTM-24, since both yield similar WERs but cltLSTM-12 has half the latency. However, the 240 ms latency of cltLSTM-12 may still result in poor user experience. The next section considers the two-head cltLSTM model to alleviate this.

4.3. Two-head cltLSTM

We extracted the time-LSTM out from cltLSTM-12, and built ltLSTM on it, as is described in Section 2.3. We then do CE, MMI, and sequence T/S training for this new ltLSTM without updating the time-LSTM parameters. The WERs and costs of this two-head cltLSTM are shown at the bottom of Tables 2 and 3 respectively. The first head, ltLSTM with zero additional latency compared to LSTM, obtained a WER of 10.03% WER after sequence T/S learning, which is slightly better than the 10.10% obtained by the ltLSTM model trained from scratch. The second head, cltLSTM-12, is the same model as the cltLSTM-12 trained from scratch. In the first-pass decoding, we run a decoder with the first head. After 240 ms, we kick off the second-pass decoding. Because the WER gap between ltLSTM and cltLSTM-12 is only 0.69% absolute, just a small fraction of the words are replaced from the first-pass results by the second-pass. Therefore, the perceived latency is small.

Since the time-LSTM units are shared in the two-head cltLSTM-12, the total number of parameters is $57 + 34 = 91$ M. We can also use an LSTM in the first-pass decoding and a separate cltLSTM-12 in the second-pass decoding. This setup also has $31 + 60 = 91$ M parameters. However, the LSTM only has a 11.49% WER, much worse than the 10.10% WER obtained from the first head ltLSTM in the two-head cltLSTM-12. When compared to a conventional baseline hybrid setup [14] that trains an LSTM with the CE and then the MMI criteria, the two-head cltLSTM-12 reduces the WER from 13.01 to 9.34%, which is a 28.2% relative reduction. While we are also working on replacing hybrid models with E2E models [35], the work conducted in this paper indeed presents us a super challenging hybrid model baseline to beat.

5. RELATION TO PRIOR WORK

Like the cltLSTM, the grid LSTM [23, 24] also operates along both the time and depth axes. However, the grid LSTM works in a layer-by-layer and step-by-step fashion, while the cltLSTM totally decouples the temporal modeling and target classification with dedicated time-LSTM and depth-LSTM units. Furthermore, the cltLSTM has the context modeling which leverages more information from future context frames, while the grid LSTM does not. Unlike the grid LSTM which cannot be configured to handle heads with different latency requirement, the decoupled temporal modeling and target classification allows for the two-head cltLSTM model design, which is shown to have low perceived latency and high accuracy.

We improve upon sequence T/S learning [26] by using a frame-level teacher combination and a strong LM for lattice generation. Using frame-level combination of the teachers and a strong LM have already been investigated for lattice-free sequence T/S [32, 36]. In this paper, these are studied in a lattice-based implementation, and also with a large training set. We also innovated the training process as CE→MMI→T/S to get the best model accuracy.

Previously investigated two-pass decoding usually starts the second-pass after the first-pass decoding finishes [9]. In contrast, the proposed two-head cltLSTM model starts the second-pass decoding just 240 ms after the first-pass decoding starts in the cltLSTM-12 setup. This yields a low perceived latency and high final accuracy.

6. CONCLUSIONS

This study aims to achieve high-accuracy and low-latency ASR by designing a two-head cltLSTM model, with one ltLSTM head for first-pass decoding and another cltLSTM head for a second-pass. The ltLSTM has the same latency as an LSTM, but has improved accuracy, due to decoupling of temporal modeling and senone classification tasks. The cltLSTM further improves upon the ltLSTM by using context frames to incorporate future information. This design enables high-accuracy and low perceived latency performance. Improvements to lattice-based sequence T/S learning were also investigated, by simplifying the teacher combination and using a strong LM, to allow the student to better emulate the teachers' runtime behaviour. When trained with Microsoft's 65 K hours anonymized training data, the proposed two-head cltLSTM model and new training strategy yield a 28.2% relative WER reduction from an LSTM trained with the conventional CE then MMI strategy, while retaining a perceived latency that is similar to the LSTM.

7. REFERENCES

[1] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "Eesen: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *Proc. ASRU*. IEEE, 2015, pp. 167–174.

[2] Hagen Soltau, Hank Liao, and Hasim Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," *arXiv preprint arXiv:1610.09975*, 2016.

[3] Eric Battenberg, Jitong Chen, Rewon Child, Adam Coates, Yashesh Gaur Yi Li, Hairong Liu, Sanjeev Satheesh, Anuroop Sriram, and Zhenyao Zhu, "Exploring neural transducers for end-to-end speech recognition," in *Proc. ASRU*. IEEE, 2017, pp. 206–213.

[4] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, K. Gonina, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. ICASSP*, 2018.

[5] Yanzhang He, Tara N Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, et al., "Streaming end-to-end speech recognition for mobile devices," in *Proc. ICASSP*, 2019, pp. 6381–6385.

[6] Shigeki Karita, Nelson Enrique Yalta Soplin, Shinji Watanabe, Marc Delcroix, Atsunori Ogawa, and Tomohiro Nakatani, "Improving transformer based end-to-end speech recognition with connectionist temporal classification and language model integration," in *Proc. Interspeech*, 2019.

[7] Chengyi Wang, Yu Wu, Yujiao Du, Jinyu Li, Shujie Liu, Liang Lu, Shuo Ren, Guoli Ye, Sheng Zhao, and Ming Zhou, "Semantic mask for transformer based end-to-end speech recognition," *arXiv preprint arXiv:1912.03010*, 2019.

[8] D. Yu and J. Li, "Recent Progresses in Deep Learning Based Acoustic Models," *IEEE/CAA J. of Autom. Sinica.*, vol. 4, no. 3, pp. 399–412, July 2017.

[9] T. Sainath, R. Pang, and et. al., "Two-pass end-to-end speech recognition," in *Proc. Interspeech*, 2019.

[10] C.-C. Chiu and C. Raffel, "Monotonic chunkwise attention," *arXiv preprint arXiv:1712.05382*, 2017.

[11] N. Moritz, T. Hori, and J. Le Roux, "Triggered attention for end-to-end speech recognition," in *Proc. ICASSP*, 2019, pp. 5666–5670.

[12] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models for speech recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 25–47, 2002.

[13] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition," in *Proc. Interspeech*, 2006.

[14] G. Pundak and T. N. Sainath, "Lower frame rate neural network acoustic models," in *Proc. Interspeech*, 2016, pp. 22–26.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] F. Beaufays H. Sak, A. Senior, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Interspeech*, 2014.

[17] Y. Zhang, G. Chen, D. Yu, K. Yao, S. Khudanpur, and J. Glass, "Highway long short-term memory RNNs for distant speech recognition," in *Proc. ICASSP*, 2016.

[18] Y. Zhao, S. Xu, and B. Xu, "Multidimensional residual learning based on recurrent neural networks for acoustic modeling," in *Proc. Interspeech*, 2016, pp. 3419–3423.

[19] J. Kim, M. El-Khamy, and J. Lee, "Residual LSTM: Design of a deep recurrent architecture for distant speech recognition," *arXiv preprint arXiv:1701.03360*, 2017.

[20] J. Li, A. Mohamed, G. Zweig, and Y. Gong, "LSTM time and frequency recurrence for automatic speech recognition," in *Proc. ASRU*, 2015.

[21] J. Li, A. Mohamed, G. Zweig, and Y. Gong, "Exploring multidimensional LSTMs for large vocabulary ASR," in *Proc. ICASSP*, 2016.

[22] T. N. Sainath and B. Li, "Modeling time-frequency patterns with LSTM vs. convolutional architectures for LVCSR tasks," in *Proc. Interspeech*, 2016.

[23] N. Kalchbrenner, I. Danihelka, and A. Graves, "Grid long short-term memory," *arXiv preprint arXiv:1507.01526*, 2015.

[24] W.-N. Hsu, Y. Zhang, and J. Glass, "A prioritized grid long short-term memory RNN for speech recognition," in *Proc. SLT*. IEEE, 2016, pp. 467–473.

[25] J. Li, L. Lu, C. Liu, and Y. Gong, "Improving layer trajectory LSTM with future context frames," in *Proc. ICASSP*, 2019.

[26] J. H. M. Wong and M. J. F. Gales, "Sequence student-teacher training of deep neural networks," in *Proc. Interspeech*, 2016.

[27] J. H. M. Wong, M. J. F. Gales, and Y. Wang, "General sequence teacher-student learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1725–1736, 2019.

[28] J. Li, C. Liu, and Y. Gong, "Layer trajectory LSTM," in *Proc. Interspeech*, 2018.

[29] M. Shannon, "Optimizing expected word error rate via sampling for speech recognition," in *Proc. Interspeech*, 2017.

[30] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning small-size DNN with output-distribution-based criteria.," in *Proc. Interspeech*, 2014, pp. 1910–1914.

[31] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[32] N. Kanda, Y. Fujita, and K. Nagamatsu, "Investigation of lattice-free maximum mutual information-based acoustic models with sequence-level Kullback-Leibler divergence," in *Proc. ASRU*, 2017, pp. 69–76.

[33] Y. Miao, J. Li, Y. Wang, S. Zhang, and Y. Gong, "Simplifying long short-term memory acoustic models for fast training and decoding," in *Proc. ICASSP*, 2016.

[34] E. Sun, J. Li, and Y. Gong, "Layer trajectory BLSTM," in *Proc. Interspeech*, 2019.

[35] Jinyu Li, Rui Zhao, Hu Hu, and Yifan Gong, "Improving RNN transducer modeling for end-to-end speech recognition," in *Proc. ASRU*, 2019.

[36] V. Manohar, P. Ghahremani, D. Povey, and S. Khudanpur, "A teacher-student learning approach for unsupervised domain adaptation of sequence-trained ASR models," in *Proc. SLT*, 2018, pp. 250–257.