Knowledge Distillation from Offline to Streaming RNN Transducer for End-to-end Speech Recognition

Gakuto Kurata[1]*, George Saon*[2]

IBM Research - AI
[1]IBM Research - Tokyo, [2]IBM T. J. Watson Research Center
gakuto@jp.ibm.com, gsaon@us.ibm.com

Abstract

End-to-end training of recurrent neural network transducers (RNN-Ts) does not require frame-level alignments between audio and output symbols. Because of that, the posterior lattices defined by the predictive distributions from different RNN-Ts trained on the same data can differ a lot, which poses a new set of challenges in knowledge distillation between such models. These discrepancies are especially prominent in the posterior lattices between an offline model and a streaming model, which can be expected from the fact that the streaming RNN-T emits symbols later than the offline RNN-T. We propose a method to train an RNN-T so that the posterior peaks at each node in the posterior lattice are aligned with the ones from a pretrained model for the same utterance. By utilizing this method, we can train an offline RNN-T that can serve as a good teacher to train a student streaming RNN-T. Experimental results on the standard Switchboard conversational telephone speech corpus demonstrate accuracy improvements for a streaming unidirectional RNN-T by knowledge distillation from an offline bidirectional counterpart.

Index Terms: End-to-end speech recognition, recurrent neural network transducer (RNN-T), knowledge distillation, streaming speech recognition

1. Introduction

End-to-end (E2E) automatic speech recognition (ASR) has been gaining attention due to its ease of training and decoding efficiency. Various modeling approaches including Connectionist Temporal Classification (CTC) [1–4], attention-based encoder-decoder [5–7], and recurrent neural network transducer (RNN-T) [8, 9] have been proposed. As accuracy of E2E ASR has been steadily improving [10–13], the importance of addressing problems related to actual deployment of E2E ASR is increasing [14–19]. Streaming ASR is one of the important requirements in many speech applications, such as closed captioning for television [20] and natural human-machine interactions [21, 22]. Among the proposed modeling techniques listed above, RNN-T and CTC enable monotonic decoding in time which is preferable for streaming ASR. An RNN-T can be seen as an extension of CTC and is composed of an encoder network for acoustic modeling, a prediction network for language modeling, and a joint network for decoding. Because of this architecture, an RNN-T does not assume conditional independence between predictions at different time steps unlike CTC. These advantages motivated us to focus in this paper on accuracy improvements for RNN-Ts applied to streaming ASR.

Compared to offline ASR, streaming ASR usually suffers an accuracy degradation. This is especially true for a streaming RNN-T with a unidirectional encoder network which sees a significant accuracy degradation from an offline RNN-T with a bidirectional encoder network [23]. This is because bidirectional models can capture whole utterance information while unidirectional models can only use the information from the past. Introducing some *look-ahead* to consider future information was shown to be effective, not only for RNN-Ts [24], but also for other models [25]. However, there is an apparent trade-off between the accuracy improvement and the latency increase introduced by look-ahead. In contrast, there were other approaches to improve the accuracy of streaming ASR without any explicit increase in latency. For CTC, knowledge distillation from bidirectional to unidirectional models has been proposed [26–29]. For attention-based encoder-decoder and hybrid RNN/HMM, twin regularization to make forward hidden states as close as backward hidden states has been applied [30, 31]. In this paper, we follow the same principle of avoiding an explicit increase in latency to improve accuracy of streaming RNN-Ts.

More specifically, we leverage knowledge distillation from an offline RNN-T with a bidirectional encoder network (*bidirectional RNN-T*) to a streaming RNN-T with a unidirectional encoder network (*unidirectional RNN-T*)[1]. Conventional DNN/HMM hybrid models [32] are trained from frame-level forced alignments between acoustic features and output symbols. Thus, naïve knowledge distillation worked well by minimizing Kullback-Leibler (KL) divergence between posterior distributions from *teacher* and *student* models at corresponding frames [33, 34]. Contrary to hybrid models, E2E models are typically trained from pairs of acoustic features and output symbols without frame-level alignments, which poses a new set of challenges in knowledge distillation between such models. Considering that bidirectional encoders typically react to acoustic features for each output symbol at earlier time steps than unidirectional encoders[2], it is not advisable to minimize the KL divergence between posterior distributions from a teacher bidirectional RNN-T and a student unidirectional RNN-T at the same time step after predicting the same output symbols. As shown later in Figure 5a and Figure 5e, RNN-T alignments for the same utterance with the bidirectional and the unidirectional model are completely different, which also indicates the difficulty of the naïve knowledge distillation approach.

In the previous paper, to realize knowledge distillation between CTC models, we aligned *posterior peaks*[3] for acoustic features at each time step from different CTC models [29]. In RNN-Ts, posterior distributions are conditioned not only on the acoustic features, but also on the output symbols predicted in the past and consequently calculating posterior distributions for acoustic features at each time step without considering the past symbols is not trivial.

[1]We use a unidirectional model for prediction networks for both bidirectional and unidirectional RNN-Ts.

[2]Similar with CTC models, a unidirectional encoder needs to consume sufficient acoustic information for each output symbol while a bidirectional encoder can leverage backward information from the acoustic features in the future [25, 35, 36].

[3]We call the symbol with the highest posterior probability a *posterior peak* hereafter.

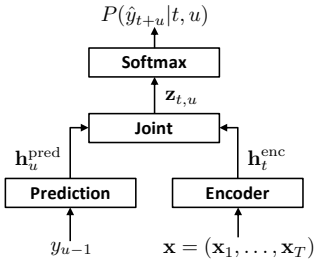


Figure 1: *Schematic diagram of RNN-T.*

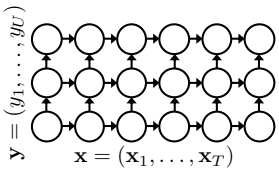


Figure 2: *Posterior lattice for RNN-T.*

In this paper, we propose a method to train an RNN-T so that the posterior peaks in the posterior lattice are aligned with the ones from a pretrained model for the same utterance. In other words, an RNN-T is trained with being guided by a pretrained model. By training a bidirectional RNN-T with being guided by a pretrained unidirectional model, we can obtain a bidirectional RNN-T that has the same posterior peaks with the pretrained unidirectional RNN-T at the same position in the posterior lattices. This bidirectional RNN-T can serve as a good teacher to train a student unidirectional model. Through experiments on the standard English Switchboard conversational telephone speech corpus, we will show that effective knowledge distillation from a bidirectional RNN-T to a unidirectional RNN-T can be realized with the proposed method.

2. RNN Transducer

Let $\mathbf{y} = (y_1, \cdots, y_U)$ denote a length-U sequence of target output symbols where we denote the target symbol set by $\mathcal{Y}$. This target symbol set can be letters, phonemes, graphemes, wordpieces, and so on [37, 38]. Let $\mathbf{x} = (\mathbf{x}_1, \cdots, \mathbf{x}_T)$ denote an acoustic feature vector over T time steps. In RNN-T modeling, an extra blank symbol ϕ is introduced to expand the length-U sequence $\mathbf{y}$ to a set of length-$(T+U)$ sequences $\Phi(\mathbf{y})$. Each sequence $\hat{\mathbf{y}} \in \Phi(\mathbf{y})$ is one of the *RNN-T alignments* between $\mathbf{x}$ and $\mathbf{y}$, where the elements of $\hat{\mathbf{y}} = (\hat{y}_1, \cdots, \hat{y}_{T+U})$ belong to the symbol set of $\mathcal{Y} \cup \{\phi\}$. The RNN-T loss is defined as the summation of symbol posterior probabilities over all possible RNN-T alignments: $\mathcal{L}_{\text{RNN-T}} = -\sum_{\hat{\mathbf{y}} \in \Phi(\mathbf{y})} P(\hat{\mathbf{y}}|\mathbf{x})$.

As shown in Figure 1, the encoder network serves as an acoustic model to convert the input features $\mathbf{x}$ to an embedding vector sequence $\mathbf{h}^{\text{enc}}$ with the same length T. The prediction network works as a language model (LM) to produce an embedding $\mathbf{h}_u^{\text{pred}}$ with conditioning on the previous predictions except the blank symbol, $(y_1, \cdots, y_{u-1})$. The joint network outputs an embedding $\mathbf{z}_{t,u}$ by combining the output from the encoder network $\mathbf{h}_t^{\text{enc}}$ and the output from the prediction network $\mathbf{h}_u^{\text{pred}}$. A common implementation of the joint network is a sum of linear transformations of both embeddings as $\mathbf{z}_{t,u} = \psi(\mathbf{W}^{\text{enc}}\mathbf{h}_t^{\text{enc}} + \mathbf{W}^{\text{pred}}\mathbf{h}_u^{\text{pred}} + \mathbf{b})$ where ψ is hyperbolic tangent, $\mathbf{W}^{\text{enc}}$ and $\mathbf{W}^{\text{pred}}$ are weight matrices, and $\mathbf{b}$ is a bias. Another linear transformation followed by a softmax operation is applied to $\mathbf{z}_{t,u}$ to calculate a posterior distribution $P(\hat{y}_{t+u}|t,u)$ over the set $\mathcal{Y} \cup \{\phi\}$. As a result, $P(\hat{y}_{t+u}|t,u)$ defines a posterior lattice as shown in Figure 2 where each node represents the posterior distribution. With these definitions, RNN-T training is achieved by minimizing the RNN-T loss $\mathcal{L}_{\text{RNN-T}}$ that can be efficiently computed by a forward-backward algorithm [8, 39].

3. Proposed Knowledge Distillation for RNN Transducer

Knowledge distillation, also known as teacher-student modeling, is a mechanism to train a student model not from the true labels for the training data, but from the posterior distributions from the pretrained teacher model that has stronger modeling capability [33, 34]. Typically, the KL divergence between the posterior distributions of the student model being trained and the teacher model for the same training sample is minimized.

In order to improve the accuracy of a unidirectional RNN-T, we try to make the best use of knowledge distillation from a bidirectional RNN-T. To realize knowledge distillation between RNN-T models, it is natural to minimize KL divergence of $T \times U$ posterior distributions in the posterior lattices of the teacher and student models. However, because bidirectional encoder and unidirectional encoder behave differently with regard to when they react to acoustic features, minimizing the KL divergence between posterior distributions at the same position in the posterior lattices is not viable. As shown later in Figure 5a and Figure 5e, the RNN-T alignments for the same utterance with the bidirectional RNN-T model and the unidirectional RNN-T model are completely different, which indicates the difficulty of the naïve knowledge distillation approach.

We propose a method to train a bidirectional RNN-T so that it has the same posterior peaks with the pretrained unidirectional RNN-T at the same position in the posterior lattice. Consequently, the trained bidirectional RNN-T can serve as a good teacher for a student unidirectional model.

Specifically, as shown in Figure 3, there are three steps in the proposed method. First in step 1, we train a unidirectional RNN-T with the standard method of minimizing the RNN-T loss. In step 2, we feed the training data to the unidirectional model trained in step 1 and obtain posterior peaks $\tilde{y}_{t,u}^{\text{uni}}$ for each position in the posterior lattice. Then we feed the same training data to the bidirectional RNN-T model being trained and obtain the posterior lattice with the same size. In addition to the normal RNN-T loss $\mathcal{L}_{\text{RNN-T}}$, we jointly minimize the cross entropy between the posterior distributions from the bidirectional RNN-T and the posterior peaks from the unidirectional RNN-T at the same position in the posterior lattice as $\mathcal{L}_{\text{XE}} = -\sum_{t=1}^{T}\sum_{u=1}^{U} \log(P^{\text{bi}}(\tilde{y}_{t,u}^{\text{uni}}|t,u))$. By minimizing $\mathcal{L}_{\text{XE}}$, the bidirectional RNN-T being trained is guided to have the same posterior peaks with the unidirectional model at the same position in the posterior lattice. Figure 4 describes this step in more detail. Note that we don't minimize the KL divergence between two posteriors from bidirectional and unidirectional RNN-T models because we need to let the bidirectional model (1) just learn the preferred symbol at each position from the posterior peaks predicted by the unidirectional model and (2) still capture whole utterance information to yield posterior distributions. If we simply minimize the KL divergence, the bidirectional model results in simply yielding similar posterior distributions with the unidirectional model and can not be a good teacher to train a unidirectional model in the next step. Finally, in step 3, we train a unidirectional model by jointly minimizing the RNN-T loss and the KL divergence between the posterior distributions at the same position in the posterior lattices from the unidirectional model being trained and the teacher bidirectional RNN-T trained in step 2.

4. Experiments

To confirm the advantage of the proposed knowledge distillation method, we conducted ASR experiments to improve accuracy of letter-based unidirectional RNN-T models. In a first set of experiments, we used 262 hours of segmented speech from the

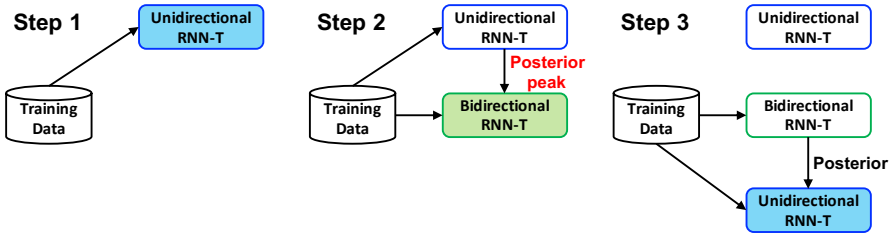


Figure 3: *Proposed knowledge distillation for RNN-T. In each step, the model filled in color is trained and others are fixed. In step 1, a unidirectional RNN-T is trained with the standard method. In step 2, a bidirectional RNN-T is trained with being guided by the unidirectional RNN-T trained in step 1. In step 3. a unidirectional RNN-T is trained by knowledge distillation from the bidirectional RNN-T trained in step 2.*

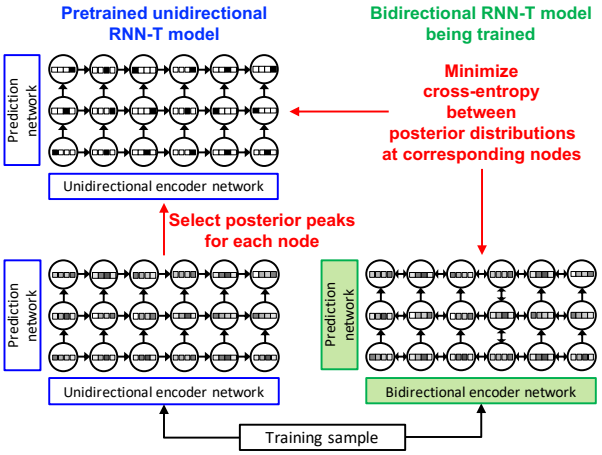


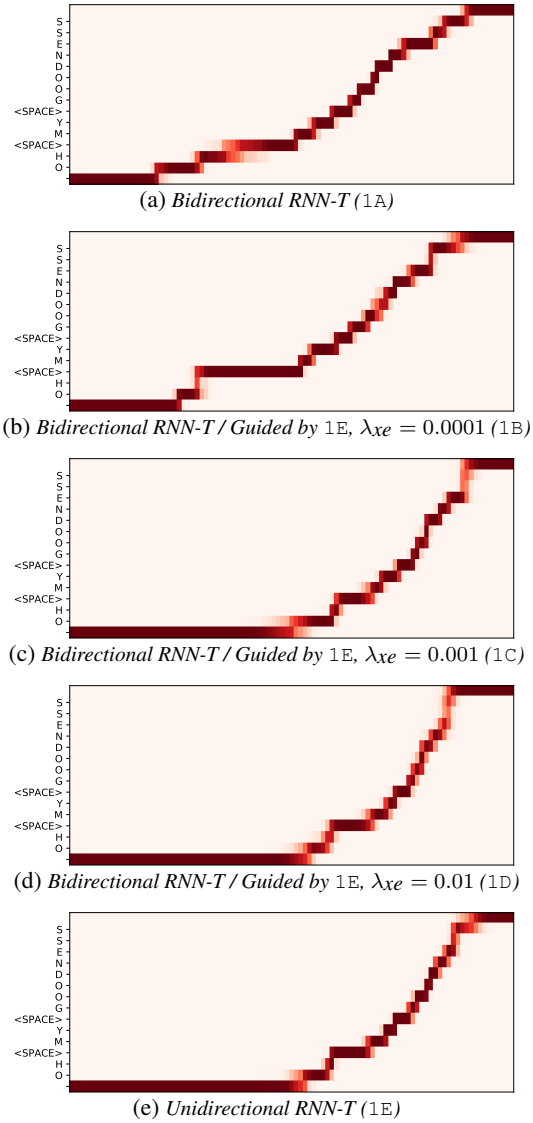
Figure 4: *Detail of step 2 in the proposed method.*

Table 1: *Word Error Rates for knowledge distillation from bidirectional (Bidir.) RNN-T to unidirectional (Unidir.) RNN-T with 262 hours of training data. [%]*

		Training method	SWB	CH
1A	Bidir.	Standard	16.7	26.6
Step 2				
1B	Bidir.	Guided by 1E, $\lambda_{\text{XE}} = 0.0001$	16.1	26.5
1C	Bidir.	Guided by 1E, $\lambda_{\text{XE}} = 0.001$	17.1	27.8
1D	Bidir.	Guided by 1E, $\lambda_{\text{XE}} = 0.01$	20.7	32.6
Step 1				
1E	Unidir.	Standard	23.3	38.9
Step 3				
1F	Unidir.	Distilled from 1A	33.0	45.4
1G	Unidir.	Distilled from 1B	30.9	43.9
1H	Unidir.	Distilled from 1C	**21.4**	**34.6**
1I	Unidir.	Distilled from 1D	21.9	35.3

Table 2: *Word Error Rates after Density Ratio Fusion with LSTM language model on key results in Table 1. [%]*

		Training method	SWB	CH
1A′	Bidir.	Standard	12.8	22.9
1E′	Unidir.	Standard	18.6	34.3
1H′	Unidir.	Distilled from 1C	**17.0**	**31.4**



(a) *Bidirectional RNN-T* (1A)

(b) *Bidirectional RNN-T / Guided by* 1E, $\lambda_{xe} = 0.0001$ (1B)

(c) *Bidirectional RNN-T / Guided by* 1E, $\lambda_{xe} = 0.001$ (1C)

(d) *Bidirectional RNN-T / Guided by* 1E, $\lambda_{xe} = 0.01$ (1D)

(e) *Unidirectional RNN-T* (1E)

Figure 5: *RNN-T alignments for an example utterance "OH MY GOODNESS" in training data. Vertical axis corresponds to target symbol sequence and horizontal axis corresponds to time.*

standard 300-hour Switchboard-1 English conversational telephone speech as training data [40]. Then we moved to a larger scale experiment by creating 1300 hours of augmented training data by applying speed and tempo perturbation (90% and 110% for both perturbation) to the original 262 hours [41]. For both cases, we report results on the Hub5 2000 Switchboard (SWB) and CallHome (CH) evaluation test sets.

For unidirectional RNN-Ts, we stacked 6 unidirectional Long-Short Term Memory (LSTM) layers with 1024 units for the encoder network. For the prediction network, we stacked an embedding layer of size 10 and 1 unidirectional LSTM layer with 1024 units. The outputs from the encoder and the predic-

Table 3: *Word Error Rates for knowledge distillation from bidirectional (Bidir.) RNN-T to unidirectional (Unidir.) RNN-T with 1300 hours of training data. (Characters in line identifiers are consistent with Table 1.) [%]*

		Training method	SWB	CH
3A	Bidir.	Standard	11.7	20.0
3C	Bidir.	Guided by 3E, $\lambda_{\text{XE}} = 0.001$	12.1	21.8
3E	Unidir.	Standard	16.1	28.6
3H	Unidir.	Distilled from 3C	**14.8**	**27.5**

Table 4: *Word Error Rates after Density Ratio Fusion with LSTM language model on key results in Table 3. [%]*

		Training method	SWB	CH
3A′	Bidir.	Standard	9.1	17.5
3E′	Unidir.	Standard	11.9	24.5
3H′	Unidir.	Distilled from 3C	**11.3**	**24.2**

tion networks were linearly transformed to 256 dimensions in the joint network. The output layer has 41 letters, 1 <SPACE> symbol that represents a word boundary, and an extra blank symbol ϕ. For bidirectional RNN-Ts, we used 6 bidirectional LSTM layers with 640 units per layer per direction. Other network topologies are the same as the unidirectional models. All parameters were initialized to samples of a uniform distribution over $(-\epsilon, \epsilon)$, where ϵ is the inverse square root of the input vector size. All models were trained for 20 epochs using stochastic gradient descent with the Nesterov momentum of 0.9 and a learning rate starting from 0.2 and annealing at $\sqrt{0.5}$ per-epoch after the 8th epoch.

For acoustic features, we used 40-dimensional logMel filterbank energies, their delta, and double-delta coefficients with frame stacking and skipping rate of 2 [35], resulting in 240-dimensional features. We did not use any speaker-dependent feature transformations since we focus on improving accuracy of streaming unidirectional RNN-T models.

For decoding, we used an efficient beam search algorithm called alignment-length synchronous decoding [42]. In addition to decoding only with a RNN-T model, we also integrate an external LM using the recently proposed density ratio fusion technique[4] [43]. To this end, we trained a letter-based unidirectional LSTM (1 layer with 1024 units) LM with the audio transcripts of the 300-hour Switchboard-1 corpus as a source LM and another letter-based unidirectional LSTM (2 layers with 1024 units) LM with the audio transcripts of the 2000-hour Switchboard-Fisher corpus as an external LM.

4.1. 262-hour Training Data

We show the Word Error Rates (WERs) of the models trained on the 262-hour data in Table 1. In addition, we show the RNN-T alignments for an example utterance from the training data in Figure 5 where the RNN-T alignments are given by the product of forward and backward variables during the forward-backward algorithm [8].

As a reference, we trained a bidirectional RNN-T in the standard method of minimizing the RNN-T loss (1A in Table 1). Based on the proposed method shown in Section 3 and Figure 3, we first trained a unidirectional model in the standard way (1E). Then, we trained bidirectional models by minimizing $\mathcal{L} = \mathcal{L}_{\text{RNN-T}} + \lambda_{\text{XE}}\mathcal{L}_{\text{XE}}$ where the latter cross entropy term is calculated based on the posterior peaks from the unidirectional RNN-T trained in the previous step. By changing the weight λ_{XE}, we trained three models shown in 1B to 1D of Table 1. By reducing the weight to $\mathcal{L}_{\text{XE}}$, we obtained similar WERs as the bidirectional RNN-T trained with the standard method. However, when looking at the RNN-T alignments in Figure 5a and 5b, the bidirectional model trained with a smaller $\mathcal{L}_{\text{XE}}$ (1B) has a similar alignment as the standard bidirectional model (1A). Conversely, when looking at Figure 5c, 5d, and 5e, the RNN-T alignments from the bidirectional models with larger $\mathcal{L}_{\text{XE}}$ (1C, 1D) are similar to the one from the unidirectional model (1E), which is preferable for the subsequent knowledge distillation. Finally, we conducted knowledge distillation from the bidirectional RNN-Ts trained in the previous step. As a reference, we also conducted naïve knowledge distillation from the bidirectional RNN-T trained with the standard method (1A). As shown in 1F and 1G of Table 1, knowledge distillation from the bidirectional models in 1A and 1B did not work. As shown in Figure 5, the RNN-T alignments are not similar with that of the unidirectional RNN-T model and thus minimizing KL divergence at the same position in the posterior lattice was not effective. However, when looking at 1H and 1I, we confirmed accuracy improvement from the unidirectional RNN-T trained with the standard method (1E). The best result was obtained in 1H with 1.9% improvement from 23.3% to 21.4% on Switchboard and 4.3% improvement from 38.9% to 34.6% on CallHome test sets from the standard unidirectional RNN-T model (1E). These results indicate that as long as the RNN-T alignments are sufficiently guided to be similar to those from the unidirectional RNN-T when training a teacher bidirectional model, a smaller λ_{XE}, which is equivalent to putting more emphasis on bidirectional information, resulted in a better teacher model (1C) and a better student model (1H).

In Table 2, we also report the WERs by applying LM density ratio fusion to the key results (1A, 1E, and 1H) in Table 1. By comparing 1E′ and 1H′ in Table 2, the improvement from the proposed method was kept after density ratio fusion.

4.2. 1300-hour Augmented Training Data

To confirm the advantage of the proposed method in a more competitive setup, we follow the same procedure as in the previous section while using the augmented 1300 hours of training data. As shown in Table 3, we set λ_{XE} to 0.001, which achieved the best result in the previous experiment. We demonstrated the same result that the bidirectional model (3C) guided by the unidirectional model (3E) served as a good teacher to train an improved unidirectional model (3H). In Table 4, we also confirmed that the accuracy improvement by the unidirectional model trained by the proposed method was reduced, but remained after density ratio fusion.

5. Conclusion

In this paper, we proposed a method to train an RNN-T so that the posterior peaks in the posterior lattice are aligned with the ones from a pretrained model for the same utterance. We confirmed that the RNN-T alignment from the model trained by the proposed method is similar to that of the pretrained model, which enables knowledge distillation between RNN-T models with different architectures. By leveraging this method, we realized knowledge distillation from an offline bidirectional RNN-T model to a streaming unidirectional RNN-T model. Experimental results on the standard 262-hour Switchboard corpus and the augmented 1300-hour corpus confirmed that the unidirectional RNN-T trained with the proposed knowledge distillation has better accuracy than the unidirectional model trained with the standard way. As future work, we plan to include a comparison with sequence-level knowledge distillation for RNN-Ts.

[4]The density ratio fusion was originally confirmed effective for the cross-domain scenario. We will show that it works well with the in-domain scenario with a small source LM and a large external LM.

6. 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] 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.

[3] H. Sak, A. Senior, K. Rao, O. Irsoy, A. Graves, F. Beaufays, and J. Schalkwyk, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," in *Proc. ICASSP*, 2015, pp. 4280–4284.

[4] 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.

[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] 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.

[7] 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.

[8] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

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

[10] 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.

[11] Z. Tüske, K. Audhkhasi, and G. Saon, "Advancing sequence-to-sequence based speech recognitio," in *Proc. INTERSPEECH*, 2019, pp. 3780–3784.

[12] 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.

[13] C. Lüscher, E. Beck, K. Irie *et al.*, "RWTH ASR systems for librispeech: Hybrid vs attention," in *Proc. INTERSPEECH*, 2019, pp. 231–235.

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

[15] N. Jaitly, Q. V. Le, O. Vinyals, I. Sutskever, D. Sussillo, and S. Bengio, "An online sequence-to-sequence model using partial conditioning," in *Proc. NIPS*, 2016, pp. 5067–5075.

[16] S. Xue and Z. Yan, "Improving latency-controlled BLSTM acoustic models for online speech recognition," in *Proc. ICASSP*, 2017, pp. 5340–5344.

[17] R. Prabhavalkar, T. N. Sainath, B. Li, K. Rao, and N. Jaitly, "An analysis of "Attention" in sequence-to-sequence models." in *Proc. INTERSPEECH*, 2017, pp. 3702–3706.

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

[19] Y. He, T. N. Sainath, R. Prabhavalkar *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *Proc. ICASSP*, 2019, pp. 6381–6385.

[20] S. Thomas, M. Suzuki, Y. Huang *et al.*, "English broadcast news speech recognition by humans and machines," in *Proc. ICASSP*, 2019, pp. 6455–6459.

[21] T. N. Sainath, R. Pang, D. Rybach *et al.*, "Two-pass end-to-end speech recognition," in *Proc. INTERSPEECH*, 2019, pp. 2773–2777.

[22] B. Li, S. Chang, T. Sainath, R. Pang, Y. R. He, T. Strohman, and Y. Wu, "Towards fast and accurate streaming end-to-end ASR," in *Proc. ICASSP*, 2020, pp. 6069–6073.

[23] E. Battenberg, J. Chen, R. Child *et al.*, "Exploring neural transducers for end-to-end speech recognition," in *Proc. ASRU*, 2016, pp. 206–213.

[24] J. Li, R. Zhao, H. Hu, and Y. Gong, "Improving RNN transducer modeling for end-to-end speech recognition," in *Proc. ASRU*, 2019, pp. 114–121.

[25] D. Amodei, S. Ananthanarayanan, R. Anubhai *et al.*, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. ICML*, 2016, pp. 173–182.

[26] S. Kim, M. L. Seltzer, J. Li, and R. Zhao, "Improved training for online end-to-end speech recognition systems," in *Proc. INTERSPEECH*, 2018, pp. 2913–2917.

[27] R. Takashima, S. Li, and H. Kaswai, "An investigation of a knowledge distillation method for CTC acoustic models," in *Proc. ICASSP*, 2018, pp. 5809–5813.

[28] G. Kurata and K. Audhkhasi, "Improved knowledge distillation from bi-directional to uni-directional LSTM CTC for end-to-end speech recognition," in *Proc. SLT*, 2018, pp. 411–417.

[29] ——, "Guiding CTC posterior spike timings for improved posterior fusion and knowledge distillation," in *Proc. INTERSPEECH*, 2019, pp. 1616–1620.

[30] D. Serdyuk, N. R. Ke, A. Sordoni, A. Trischler, C. Pal, and Y. Bengio, "Twin networks: Matching the future for sequence generation," in *Proc. ICLR*, 2018.

[31] M. Ravanelli, D. Serdyuk, and Y. Bengio, "Twin regularization for online speech recognition," *arXiv preprint arXiv:1804.05374*, 2018.

[32] G. Hinton, L. Deng, D. Yu *et al.*, "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.

[33] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Proc. NIPS*, 2014, pp. 2654–2662.

[34] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[35] 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.

[36] H. Sak, F. de Chaumont Quitry, T. Sainath, K. Rao *et al.*, "Acoustic modelling with CD-CTC-sMBR LSTM RNNs," in *Proc. ASRU*, 2015, pp. 604–609.

[37] Y. Wu, M. Schuster, Z. Chen *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[38] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer," in *Proc. ICASSP*, 2017, pp. 193–199.

[39] T. Bagby, K. Rao, and K. C. Sim, "Efficient implementation of recurrent neural network transducer in Tensorflow," in *Proc. SLT*, 2018, pp. 506–512.

[40] G. Saon, G. Kurata, T. Sercu *et al.*, "English conversational telephone speech recognition by humans and machines," in *Proc. INTERSPEECH*, 2017, pp. 132–136.

[41] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. INTERSPEECH*, 2015, pp. 3586–3589.

[42] G. Saon, Z. Tüske, and K. Audhkhasi, "Alignment-length synchronous decoding for RNN transducer," in *Proc. ICASSP*, 2020, pp. 7804–7808.

[43] E. McDermott, H. Sak, and E. Variani, "A density ratio approach to language model fusion in end-to-end automatic speech recognition," in *Proc. ASRU*, 2019, pp. 434–441.