



Guiding CTC Posterior Spike Timings for Improved Posterior Fusion and Knowledge Distillation

Gakuto Kurata, Kartik Audhkhasi

IBM Research AI

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

Abstract

Conventional automatic speech recognition (ASR) systems trained from frame-level alignments can easily leverage posterior fusion to improve ASR accuracy and build a better single model with knowledge distillation. End-to-end ASR systems trained using the Connectionist Temporal Classification (CTC) loss do not require frame-level alignment and hence simplify model training. However, sparse and arbitrary posterior spike timings from CTC models pose a new set of challenges in posterior fusion from multiple models and knowledge distillation between CTC models. We propose a method to train a CTC model so that its spike timings are guided to align with those of a pre-trained *guiding* CTC model. As a result, all models that share the same guiding model have aligned spike timings. We show the advantage of our method in various scenarios including posterior fusion of CTC models and knowledge distillation between CTC models with different architectures. With the 300-hour Switchboard training data, the single word CTC model distilled from multiple models improved the word error rates to 13.7%/23.1% from 14.9%/24.1% on the Hub5 2000 Switchboard/CallHome test sets without using any data augmentation, language model, or complex decoder.

Index Terms: End-to-end speech recognition, Connectionist Temporal Classification (CTC), Knowledge distillation

1. Introduction

End-to-end (E2E) automatic speech recognition (ASR) using the Connectionist Temporal Classification (CTC) loss function has been gathering interest since it significantly simplifies model training pipelines [1–7]. Prior to E2E ASR systems, the conventional training pipeline for GMM-HMM systems [8] and DNN-HMM hybrid systems [9] required output symbols for every input acoustic frame, that is, *frame-level alignment*, which made the training process complex and time-consuming. Instead, E2E ASR only requires pairs of input feature sequences and output symbol sequences [1, 10–14], such as phones [15], characters [16], words [7, 17], or their combinations [18, 19].

One advantage of training ASR models with frame-level alignments is that frame-level posterior fusion of multiple systems is easy. Models trained from the same frame-level alignment data have the same target output symbol for each frame. Thus, we can compute the average or a weighted average of the posteriors from multiple systems to yield better posteriors, which results in improved ASR accuracy [20] because such posterior averaging effectively creates a mixture model comprising of the different systems being combined. We can also use the (weighted) average of posteriors from multiple acoustic models as a teacher to train a single student model via knowledge distillation [21–25].

In contrast, as noted in prior literature [1, 3], CTC models emit very *spiky* posterior distributions¹ where most frames emit the garbage *blank* symbol with high probability and only a few frames emit the target symbols of interest. As a result, CTC models trained with the same training data can have different spike timings (typical examples are shown later in Figure 3(a)

¹We call the highest posterior except for blank and silence at each time index a “spike” hereafter.

and 3(e)), which poses a new set of challenges. Due to non-aligned spike timings, we cannot make better posteriors by computing their (weighted) average across multiple systems [26]. Hence, it’s difficult to improve ASR accuracy by a naive posterior fusion, and also to use the (weighted) average of posteriors as a teacher to train a single model via knowledge distillation.

Disagreement of CTC spike timings has been discussed in the context of knowledge distillation from bidirectional LSTM (BiLSTM) CTC to unidirectional LSTM (UniLSTM) CTC models [27] and from multiple accent-specific models to a single multi-accent model [28]. [27] proposed searching for a similar posterior from a BiLSTM CTC model at a different time index to train a UniLSTM model. This approach only focused on knowledge distillation from BiLSTM to UniLSTM and cannot be simply extended to a posterior fusion of multiple systems. [28] attempted to train an accent-specific model for each accent by feeding training data of the target accent to a common teacher model trained from the data of all accents via knowledge distillation. As a result, each accent-specific model has aligned spike timings with the common teacher model. This method did not address knowledge distillation between different neural network (NN) architectures, such as between BiLSTM and UniLSTM CTC models.

In this paper, we propose a method to explicitly guide the CTC spike timings to be aligned with those from a pre-trained CTC model with the NN architecture of interest. We call this pre-trained CTC model a *guiding model*. More specifically, when training a CTC model, in addition to a normal CTC loss, we add a new loss term that forces the spikes from the model being trained to occur at the same time as those from the guiding model. Hence, models guided by the same guiding model have aligned spike timings. The advantages of our proposed method are as follows:

Posterior fusion of multiple CTC models: We can make better posteriors by averaging aligned posteriors from multiple models that share the same guiding model and improve ASR accuracy as a result. In addition, we can use these averaged posteriors as a teacher to train a single model with improved ASR accuracy through knowledge distillation.

Knowledge distillation between CTC models with arbitrary neural network architectures: The spike timings of CTC models with different NN architectures can be significantly different, which makes knowledge distillation between such models difficult. A typical case is knowledge distillation from a BiLSTM CTC model to a UniLSTM CTC model. By training a teacher model using a guiding model that has the same architecture as the final desired student model, the trained teacher model can have the spike timings appropriate for the student model and can be used in knowledge distillation.

Improved speech recognition accuracy: The CTC model guided by the guiding model has improved ASR accuracy compared to the model without the guiding model. The guiding model promotes a good alignment path and hence does not allow the CTC training to assign sufficient probabilities to bad alignment paths, which results in an improved accuracy.

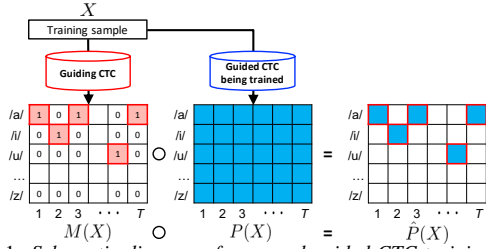


Figure 1: Schematic diagram of proposed guided CTC training. Vertical and horizontal axes of the matrices indicate output symbols and time indexes, respectively. The guiding CTC model at the left makes a mask that sets a 1 only at the output symbol of the highest posterior at each time index. This mask is applied to the posterior from the model being trained by Hadamard (element-wise) product. The summation of the masked posteriors at the right is multiplied by minus one and is minimized jointly with the CTC loss.

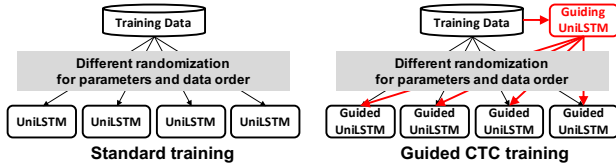


Figure 2: Standard and guided CTC training for UniLSTM CTC models.

2. Connectionist Temporal Classification

Let \mathbf{y} denote a length- L sequence of target output symbols. Let \mathbf{X} denote acoustic feature vectors over T time steps. The conventional alignment-based DNN/HMM hybrid system training requires L to be equal to T [9]. The alignment-free CTC introduces an extra blank symbol ϕ that expands the length- L sequence \mathbf{y} to a set of length- T sequences $\Phi(\mathbf{y})$. Each sequence $\hat{\mathbf{y}} \in \Phi(\mathbf{y})$ is one of the CTC alignments between \mathbf{X} and \mathbf{y} . The CTC loss is defined as the summation of symbol posterior probabilities over all possible CTC alignments: $\mathcal{L}_{CTC} = -\sum_{\hat{\mathbf{y}} \in \Phi(\mathbf{y})} P(\hat{\mathbf{y}}|\mathbf{X})$. We use phones and words as target output symbols, namely *phone CTC* and *word CTC*, for evaluation in Section 4.

3. Guided CTC training

In order to guide the spike timings of CTC models, we propose a two-step training process and name it *guided CTC training*. First, we train a guiding CTC model that has appropriate spike timings for the final target use case. Then, we train another CTC model (the *guided model*) so that similar spike timings as that of the guiding model can be obtained. Note that as long as the set of the output symbols is the same, different NN architectures can be used for the guiding and the guided models.

Figure 1 shows a schematic diagram on how to guide the spike timings of the guided model by using the pre-trained guiding CTC model, assuming that we have completed the training of the guiding model in advance. When training the guided CTC model, for a training sample \mathbf{X} of T time steps, we feed it to the guiding CTC model, do the forward pass, and obtain posteriors for each time index. We convert these posteriors to a mask $M(\mathbf{X})$ by setting a 1 at the output symbol with the highest posterior and 0 at other symbols at each time index. In cases where the blank symbol ϕ has the highest posterior, we set 0 for all symbols at this time index. This mask passes the output symbol that the guiding CTC model emits at each time index. Then we feed the same training sample to the guided CTC model being trained and obtain posteriors $P(\mathbf{X})$. By the Hadamard (element-wise) product of the mask and the posteriors, we obtain the masked posteriors $\hat{P}(\mathbf{X}) = M(\mathbf{X}) \circ P(\mathbf{X})$. The summation of masked posteriors $\hat{P}(\mathbf{X})$ becomes greater if the guided CTC model being trained has spikes for the same output symbol at the same times as the guiding CTC model. Thus, by maximiz-

Table 1: WERs for posterior fusion and ROVER of UniLSTM phone CTC models. [%]

	SWB	CH
1A UniLSTM	15.3	27.6
1B 4× posterior fusion of 1A	15.4	28.8
1C 4× ROVER of 1A	14.1	26.1
1D UniLSTM guided by UniLSTM	14.4	26.2
1E 4× posterior fusion of 1D	12.9	24.2
1F 4× ROVER of 1D	13.7	24.5

ing this summation, we can guide the spike timings of the guided CTC model to be the same as those of the guiding CTC model. We call the summation of the masked posteriors multiplied by -1 the *guide loss* defined as $\mathcal{L}_G = -1 \cdot \sum \hat{P}(\mathbf{X})^2$. The overall loss for the guided CTC training becomes the summation of the standard CTC loss and the guide loss as $\mathcal{L} = \mathcal{L}_{CTC} + \mathcal{L}_G$.

4. Experiments

We conducted ASR experiments on phone and word CTC models to determine the advantages of our proposed guided CTC training. We used the standard 300-hour Switchboard English conversational telephone speech as training data [20]. We trained multiple models for each experiment to investigate the effect of posterior fusion. By using different seeds for random parameter initialization, even models with exactly the same architecture trained with the same procedure can have sufficient diversity to benefit from posterior fusion [23]. We follow this procedure and also randomized the order of the training data after the first epoch (SortaGrad) [29].

4.1. Posterior fusion of UniLSTM phone CTC models

First, we conducted experiments on posterior fusion with UniLSTM phone CTC models. We used 40-dimensional logMel filterbank energies, their delta, and double-delta coefficients with frame stacking and skipping rate of 2 [30], resulting in 240-dimensional features. We used 44 phones from the Switchboard pronunciation lexicon [20] and the blank symbol. For decoding, we trained a 4-gram language model with 24M words from the Switchboard + Fisher transcripts with a vocabulary size of 30K. We constructed a CTC decoding graph similar to the one in [2]. For NN architecture, we stacked 6 UniLSTM layers with 640 units and a fully-connected linear layer of 640×45 , 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 evaluation, we used the Hub5-2000 Switchboard (SWB) and CallHome (CH) test sets.

Figure 2 shows the flow of experiments and Table 1 lists the results. We trained 4 UniLSTM models with the standard training. In 1A, we averaged the Word Error Rates (WERs) for the 4 decoding outputs from the 4 models. For posterior fusion in 1B, we averaged the posteriors from the 4 models and used it for decoding with the graph. For comparison, we also combined the 4 decoding outputs from the 4 models by using the Recognizer Output Voting Error Reduction (ROVER) [31–33], as shown in 1C. Comparing 1A and 1B, we can see that since the posteriors from the 4 models were not aligned, we did not benefit from posterior fusion. ROVER gave us a solid improvement in 1C.

For the proposed guided CTC training, as shown in the right of Figure 2, we first trained a UniLSTM phone CTC model with the same architecture and training data and then used it as a guiding model to train 4 guided models. 1D indicates the average WERs of the 4 decoding outputs by the 4 guided UniLSTM models trained by the proposed guided CTC training. Comparing 1A and 1D, we can see a WER reduction by the proposed guided

²The logarithmic guide loss is equivalent to a frame-level cross-entropy where the target is a sequence of the output symbols with the highest posterior from the guiding model. Note that the mask $M(\mathbf{X})$ is derived from this target sequence.

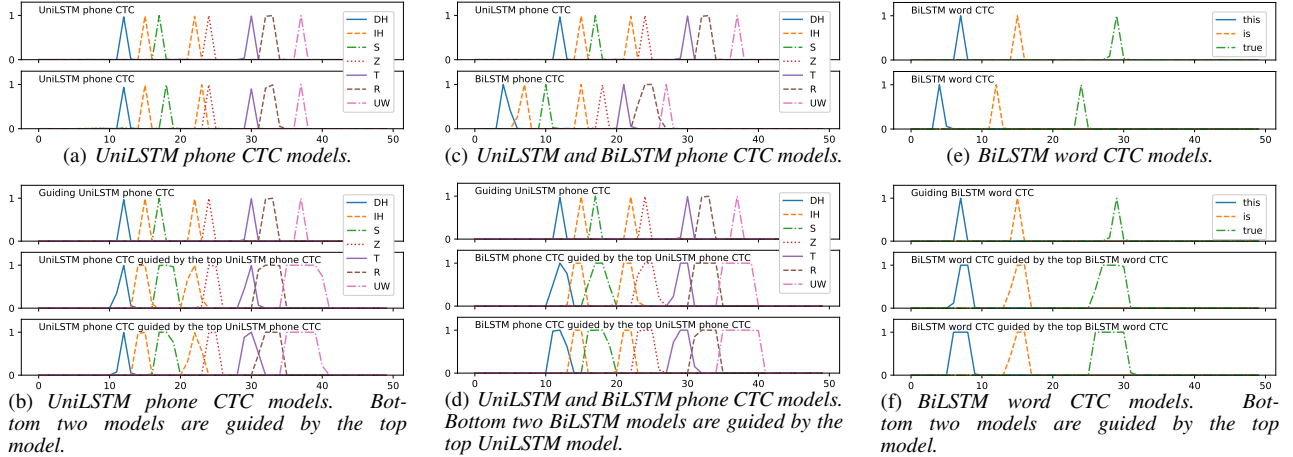


Figure 3: Posteriors for “this (DH IH S) is (IH S) true (T R UW)” in the SWB test set.

Table 2: WERs for knowledge distillation from BiLSTM to UniLSTM phone CTC models. [%]

	SWB	CH
2A UniLSTM	15.3	27.6
2B BiLSTM	11.8	21.8
2C UniLSTM distilled from		
1 × BiLSTM (2B)	17.1	29.9
4 × BiLSTMs (2B)	29.4	32.7
2D BiLSTM guided by UniLSTM	12.4	22.6
2E UniLSTM distilled from		
1 × BiLSTM guided by UniLSTM (2D)	13.4	25.4
4 × BiLSTMs guided by UniLSTM (2D)	12.9	24.8
8 × BiLSTMs guided by UniLSTM (2D)	12.9	24.7

CTC training. The guide loss promotes a good CTC alignment path and assigns lower probabilities to bad CTC alignment paths on the basis of the guiding model during the CTC training³. 1E and 1F show WERs by the posterior fusion and ROVER of the 4 models trained with the proposed guided CTC training. Due to the aligned posteriors, we benefit from the posterior fusion that outperformed the ROVER. Note that posterior fusion requires decoding with the graph just once, while ROVER needs 4 separate decodes. This is another advantage of posterior fusion realized by the proposed guided training.

Figure 3(a) shows posteriors of multiple UniLSTM phone CTC models for the same utterance, “this (DH IH S) is (IH S) true (T R UW)”, in the SWB test set, where symbols in parentheses represent phones. As expected, phone spike positions were not aligned between the UniLSTM CTC models. Figure 3(b) shows posteriors of the guiding UniLSTM CTC model at the top and the UniLSTM CTC models guided by the guiding model at the top with the proposed training. The spikes from the guided models were temporally smoothed and the spike positions of the guiding models were covered by the smoothed spikes of the guided models. Hence, the spike timings from multiple guided models overlapped, which underpins the improved WERs by posterior fusion.

4.2. Knowledge distillation from BiLSTM phone CTC to UniLSTM phone CTC models

Next, we explored knowledge distillation from BiLSTM to UniLSTM phone CTC models. UniLSTM CTC models are more suitable for deployment in actual streaming ASR services and products [26, 34] and closing the ASR accuracy gap between UniLSTM and BiLSTM CTC models is critical [27, 35, 36]. Here, we used the same speaker-independent (SI) 240-

³A naïve alternative is to conduct the standard UniLSTM CTC training starting from a pre-trained guiding CTC model and we obtained the WERs of 15.0% and 27.3%. The larger improvements obtained in 1D demonstrate the advantage of our proposed guided CTC training.

dimensional feature vectors as the previous experiments while keeping the actual use-case of streaming ASR in mind. For teacher BiLSTM phone CTC models, we stacked 6 BiLSTM layers with 320 units each in the forward and backward layers and a fully-connected linear layer of 640×45 , followed by a softmax activation function. For UniLSTM phone CTC models, we used the same architecture as in the previous experiments. All NN parameters were initialized in the same fashion as the previous experiments.

Table 2 shows the results. 2A and 2B indicate the WERs by UniLSTM and BiLSTM phone CTC models trained with the standard procedure. As a naïve approach, in 2C, we trained a UniLSTM model through knowledge distillation from BiLSTM models with minimizing the frame-wise KL divergence, but saw degradation in WERs. The spike timings from BiLSTM were not appropriate for UniLSTM, as also demonstrated in Figure 3(c), where the spike timings for the same utterance were completely different between the UniLSTM and BiLSTM models. In addition, posteriors from BiLSTM models trained with the standard training were not aligned and thus knowledge distillation from their posterior fusion was not successful.

2D is the key step where we first trained a guiding UniLSTM model and then trained a BiLSTM model guided by the guiding UniLSTM model. The trained BiLSTM model had spike timings appropriate for UniLSTM model as shown in Figure 3(d) while posteriors were estimated with bidirectional context. Comparing 2B and 2D, we can see that 2D had a worse WER because the spike positions were unnatural for a BiLSTM. However, this model can serve as an appropriate teacher to train a UniLSTM model. We trained multiple BiLSTM models guided by the same guiding UniLSTM model and used their posterior fusion as a teacher to train a single student UniLSTM model by knowledge distillation, where the frame-wise KL divergence from the student model to the fused teacher posterior was minimized. 2E shows the WERs by the UniLSTM models distilled from the posterior fusion of 1, 4, and 8 BiLSTM models guided by the same guiding UniLSTM model. Even in the case of just 1 guided BiLSTM model, the WER for SWB was reduced to 13.4%, which was equivalent to reducing the gap between BiLSTM and UniLSTM by 54.3%⁴. By increasing the number of BiLSTM models, the WER for SWB was reduced to 12.9%, which equals a 68.6% accuracy gap reduction. As in the bottom two posteriors in Figure 3(d), the temporally smoothed spikes of the guided BiLSTM models overlapped around the spikes of the guiding UniLSTM model, which underpins the improved WERs by increasing the number of guided BiLSTM models.

⁴We reported a smaller accuracy gap reduction (45.2%) with a similar setting in our former paper [27]. Besides, this method could not be extended to use a posterior fusion as a teacher.

Table 3: WERs for posterior fusion, ROVER, and knowledge distillation of BiLSTM word CTC models. [%]

	SWB	CH	RT02	RT03	RT04	Avg.
3A BiLSTM	14.9	24.1	23.7	24.1	22.6	21.9
3B 4×posterior fusion of 3A	48.2	57.7	57.7	58.9	59.3	56.4
3C 4×ROVER of 3A	16.0	23.2	24.8	26.1	26.7	23.3
3D BiLSTM guided by BiLSTM	14.3	23.3	23.1	23.8	22.0	21.3
3E 4×posterior fusion of 3D	11.7	20.2	19.2	19.7	18.5	17.9
3F 4×ROVER of 3D	13.0	20.6	20.9	21.2	19.9	19.1
3G BiLSTM distilled from 4×posterior fusion (3E)	13.7	23.1	22.4	22.9	21.7	20.8

Table 4: Comparison with published CTC-based direct acoustic-to-word models. [%]

	Alignments	SWB	CH
4A Word CTC [19]	N	14.6	23.6
4B Hierarchical subword CTC [37]	N	14.0	27.1
4C Joint CTC-CE [38]	Y	13.0	23.4
4D Our best single model	N	13.7	23.1
4E Our best posterior fusion	N	11.7	20.2

4.3. Posterior fusion and knowledge distillation of BiLSTM word CTC models

We applied our proposed guided CTC training to BiLSTM word CTC models. For input acoustic features, we added 100-dimensional i-vectors for each speaker extracted in advance and appended them to the same SI feature used in the previous experiments, resulting in 340-dimensional feature vectors [39]. We selected words with at least 5 occurrences in the training data [17, 19]. This resulted in an output layer with 10,175 words and the blank symbol. We stacked 6 BiLSTM layers with 512 units each in the forward and backward layers (BiLSTM encoder), added 1 fully-connected linear layer with 256 units to reduce computation [40], and put 1 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 BiLSTM phone CTC model⁵ [17, 19]. 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 evaluation, in addition to SWB and CH, we used RT02, RT03, and RT04 test sets [20].

Table 3 shows the results. We trained 4 BiLSTM models with the standard training and conducted posterior fusion and ROVER. 3A indicates the average WERs of 4 decoding outputs from the 4 models and 3B and 3C indicate the WERs by posterior fusion and ROVER, respectively. Comparing 3A and 3B, since the posteriors from the 4 models were not aligned and the spikes from word models were much more sparse than those from phone models, we saw a significant degradation in accuracy. Due to the sparse non-aligned word spikes, ROVER also did not improve the ASR accuracy, as in 3C. Figure 3(e) shows the word posteriors from BiLSTM word CTC models that were sparse and non-aligned, which caused degradation in 3B and 3C.

For the proposed guided CTC training, we first trained a BiLSTM word CTC model with the same architecture and training data and then used it as a guiding model to train the 4 guided models. 3D indicates the average WERs of 4 decoding outputs by the 4 BiLSTM word models trained by the proposed guided CTC training. Comparing 3A and 3D, we can see WER reduction by the proposed guided CTC training, the same as with the case of phone CTC models in Section 4.1. 3E and 3F show WERs by the posterior fusion and ROVER of the 4 guided models. Due to the aligned posteriors obtained by the guided CTC training, both posterior fusion and ROVER improved ASR accuracy compared with 3D while posterior fusion outperformed ROVER. Finally, we used the posterior fusion from the guided BiLSTM models as a teacher and trained a single student BiL-

⁵To initialize multiple word CTC models for posterior fusion and ROVER, we trained multiple phone CTC models with different parameter initialization and training data order.

Table 5: Coverage ratio of posterior spikes. [%]

	Training data	Test data
UniLSTM phone CTC		
2 non-guided models	68.4	66.9
Guiding and guided models	91.7	89.4
2 guided models	88.1	86.6
BiLSTM word CTC		
2 non-guided models	38.8	34.4
Guiding and guided models	92.3	85.7
2 guided models	88.2	82.9

STM word CTC model by knowledge distillation with minimizing the frame-wise KL divergence. 3G shows the results where the WERs were further decreased from 3D consistently over all test sets.

Table 4 shows a comparison on WERs with the published CTC-based direct acoustic-to-word models trained from the standard 300-hour Switchboard data without using any data augmentation or language model. To the best of our knowledge, our best single model trained by knowledge distillation outperformed the published CTC models trained in a purely E2E fashion in 4A and 4B [19, 37]. By combining CTC and cross-entropy (CE) training using the alignment between the acoustic frame and the labels, as in a conventional non-E2E system, 13.0% and 23.4% were achieved in 4C [38]. Our best single model achieved the comparable WERs of 13.7% and 23.1% with purely E2E training without any frame-level alignments.

4.4. Analysis on spike timings

Looking at the posteriors in Figure 3, we can see that spikes from models that were trained with the standard training (we call them *non-guided* models) were not aligned, while the spikes from the guiding and the guided models and those of the guided models that share the same guiding model were aligned. We quantitatively analyzed the coverage ratio by investigating if the posterior spikes (except for the blank and silence) from one model were covered by the spikes from the other model at the same time index. We used the training data and the test data (SWB) for this analysis; the results are shown in Table 5. Note that in the case of “guiding and guided models”, we investigated if the spikes from the guiding model were covered by the corresponding guided model. For other cases, we randomly picked 2 models trained using the same training procedure with changing the parameter initialization and the training data order. For both UniLSTM phone CTC and BiLSTM word CTC models, the coverage ratio of 2 non-guided models was low. Especially, due to sparse spikes, the coverage ratio of BiLSTM word CTC was much lower, which underpins the poor WERs by the posterior fusion in 3B of Table 3. The coverage ratio between the guiding and guided model was improved and the ratio between 2 guided models was slightly worse, but much higher than the ratio between the 2 non-guided models. Comparing the training and test data, the training data had a slightly higher coverage ratio, but the same trends were seen for all combinations of models, which supports the experimental results.

5. Conclusion

We proposed a method to guide spike timings of CTC models by using a pre-trained CTC model with the NN architecture of interest. We demonstrated its advantages in various scenarios including posterior fusion of multiple CTC models and knowledge distillation between CTC models. Through the experiments, we achieved state-of-the-art WERs in the CTC-based direct acoustic-to-word setting without using any data augmentation or language model. By qualitatively and quantitatively investigating the posterior spike timings from the CTC models trained with the proposed guided CTC training, we confirmed that the spike timings were aligned between the guiding and the guided models and between the multiple guided models.

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, “EESN: 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. Isroy, 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] G. Pundak and T. Sainath, “Lower frame rate neural network acoustic models,” in *Proc. INTERSPEECH*, 2016, pp. 22–26.
- [5] Y. Miao, M. Gowayyed, X. Na, T. Ko, F. Metze, and A. Waibel, “An empirical exploration of CTC acoustic models,” in *Proc. ICASSP*, 2016, pp. 2623–2627.
- [6] G. Zweig, C. Yu, J. Droppo, and A. Stolcke, “Advances in all-neural speech recognition,” in *Proc. ICASSP*, 2017, pp. 4805–4809.
- [7] 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.
- [8] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [9] 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.
- [10] A. Graves, A. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Proc. ICASSP*, 2013, pp. 6645–6649.
- [11] A. Graves, “Generating sequences with recurrent neural networks,” *arXiv preprint arXiv:1308.0850*, 2013.
- [12] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in *Proc. ICML*, 2014, pp. 1764–1772.
- [13] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Sathesh, S. Sengupta, A. Coates *et al.*, “Deep speech: Scaling up end-to-end speech recognition,” *arXiv preprint arXiv:1412.5567*, 2014.
- [14] A. Hannun, “Sequence modeling with CTC,” *Distill*, 2017, <https://distill.pub/2017/ctc>.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] H. Liu, Z. Zhu, X. Li, and S. Sathesh, “Gram-CTC: Automatic unit selection and target decomposition for sequence labelling,” in *Proc. ICML*, 2017, pp. 2188–2197.
- [19] 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.
- [20] 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.
- [21] J. Ba and R. Caruana, “Do deep nets really need to be deep?” in *Proc. NIPS*, 2014, pp. 2654–2662.
- [22] 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.
- [23] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.
- [24] Y. Chebotar and A. Waters, “Distilling knowledge from ensembles of neural networks for speech recognition,” in *Proc. INTERSPEECH*, 2016, pp. 3439–3443.
- [25] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran, “Efficient knowledge distillation from an ensemble of teachers,” in *Proc. INTERSPEECH*, 2017, pp. 3697–3701.
- [26] H. Sak, F. de Chaumont Quiry, T. Sainath, K. Rao *et al.*, “Acoustic modelling with CD-CTC-sMBR LSTM RNNs,” in *Proc. ASRU*, 2015, pp. 604–609.
- [27] 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.
- [28] S. Ghorbani, A. E. Bulut, and J. H. Hansen, “Advancing multi-accented LSTM-CTC speech recognition using a domain specific student-teacher learning paradigm,” in *Proc. SLT*, 2018, pp. 29–35.
- [29] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, “Deep speech 2: End-to-end speech recognition in English and Mandarin,” in *Proc. ICML*, 2016, pp. 173–182.
- [30] 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.
- [31] J. G. Fiscus, “A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER),” in *Proc. ASRU*, 1997, pp. 347–354.
- [32] Y. Wang, X. Deng, S. Pu, and Z. Huang, “Residual convolutional CTC networks for automatic speech recognition,” *arXiv preprint arXiv:1702.07793*, 2017.
- [33] L. Fritz and D. Burshtein, “Simplified end-to-end MMI training and voting for ASR,” *arXiv preprint arXiv:1703.10356*, 2017.
- [34] H. Soltau, H. Liao, and H. Sak, “Reducing the computational complexity for whole word models,” in *Proc. ASRU*, 2017, pp. 63–68.
- [35] 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.
- [36] 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.
- [37] R. Sanabria and F. Metze, “Hierarchical multi task learning with CTC,” in *Proc. SLT*, 2018, pp. 485–490.
- [38] C. Yu, C. Zhang, C. Weng, J. Cui, and D. Yu, “A multistage training framework for acoustic-to-word model,” in *Proc. INTERSPEECH*, 2018, pp. 786–790.
- [39] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.
- [40] 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.