INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Maximum A Posteriori based Decoding for CTC Acoustic Models

Naoyuki Kanda, Xugang Lu, Hisashi Kawai

National Institute of Information and Communications Technology, Japan
{naoyuki.kanda,xugang.lu,hisashi.kawai}@nict.go.jp

Abstract

This paper presents a novel decoding framework for connectionist temporal classification (CTC)-based acoustic models (AM). Although CTC-based AM inherently has the property of a language model (LM) in itself, an external LM trained with a large text corpus is still essential to obtain the best results. In the previous literatures, a naive interpolation of the CTC-based AM score and the external LM score was used, although there is no theoretical justification for it. In this paper, we propose a theoretically more sound decoding framework derived from a maximization of the posterior probability of a word sequence given an observation. In our decoding framework, a subword LM (SLM) is newly introduced to coordinate the CTC-based AM score and the word-level LM score. In experiments with the Wall Street Journal (WSJ) corpus and Corpus of Spontaneous Japanese (CSJ), our proposed framework consistently achieved improvements of 7.4 – 15.3 % over the conventional interpolation-based framework. In the CSJ experiment, given 586 hours of training data, the CTC-based AM finally achieved a 6.7 % better word error rate than the baseline method with deep neural networks and hidden Markov models.

Index Terms: Connectionist temporal classification, long-short term memory cell, acoustic model

1. Introduction

The hybrid framework of hidden Markov models (HMMs) and deep neural networks (DNNs) has achieved great progress in speech recognition [1, 2], showing high recognition accuracies in numerous scenarios [3, 4, 5]. Encouraged by the strong performance of DNNs, various extensions of the network architecture have been proposed, such as convolutional neural networks (CNNs) [6], sigmoid-unit-type recurrent neural networks (RNNs) [7, 8, 9, 10], and long short-term memory neural networks [11, 12, 13, 14, 15]. These HMM-based hybrid approaches have led to great improvements in speech recognition accuracy. Notwithstanding, further advances are still needed.

Recently, the connectionist temporal classification (CTC) [16], which enables a neural network to learn the direct mapping from audio input $\mathbf{X}$ to symbol sequence $\mathbf{s}$, has been successfully applied to large vocabulary continuous speech recognition (LVCSR) tasks [17, 18, 19, 20, 21, 22, 23]. Because CTC-based neural networks can have the property of both acoustic models (AMs) and language models (LMs), some initial attempts tried to use only one neural network (i.e., without an external LM) in decoding. However, it became apparent that the external LM was still essential for obtaining the best results [17, 18, 23, 24]. One difficulty of using the external LM with CTC-AM is that both models contain the properties of an LM, which is different from the case of the HMM-based decoding framework. In addition, CTC-AM is normally trained on the subword unit (e.g., characters or phonemes), [1] which makes it more difficult to combine the subword-level CTC-AM score with the word-level LM score. As a result, previous works has used a naive interpolation of the CTC-AM score $Pr(\mathbf{s}|\mathbf{X})$ and external LM score $Pr(\mathbf{W})$ although there is no theoretical justification for using such an interpolation [17, 18, 19, 20, 21, 22, 23]. Heuristic priors were sometimes introduced on the interpolated score [19, 20, 23]; however, such priors also lacked any theory behind them. This lack of theory has led the systems to produce inconsistent results, often with worse performance than the state-of-the-art DNN-HMM baseline [23, 24].

In this paper, we propose a novel theoretically based decoding framework for CTC-AM. In our framework, the speech recognition problem is defined as a problem of finding word sequence $\mathbf{W}$ that maximizes posterior probability $Pr(\mathbf{W}|\mathbf{X})$ given audio input $\mathbf{X}$. This framework itself is the same as the traditional HMM-based one, but we expand the formula to fit the CTC-AM. A subword-level LM is newly introduced to appropriately connect the CTC-AM score and word-level LM score. Our framework can be represented by the weighted finite state transducer (WFST), similarly to the previous work [23], so it is easy to extend the previous framework into our framework. The proposed method is an extension of the "direct decoding" approach for cross entropy-based RNN-AM [10], which was recently proposed by the authors of this paper.

In the next section, we first explain the conventional framework using CTC-based AMs. We then introduce our proposed decoding framework in Section 3. Finally, Section 4 presents the various experimental results on English and Japanese LVCSR tasks.

2. CTC-based AM and its conventional decoding framework

2.1. CTC

CTC is a neural network model that can directly learn the mapping function from observation sequence $\mathbf{X}$ to symbol sequence $\mathbf{s}$. CTC was first proposed for phoneme recognition [16, 11], and was recently extended to LVCSR tasks [17, 18, 19, 20, 21, 22, 23]. In CTC-modeling, subwords (e.g., characters or phonemes) are normally used as the recognition unit.

In CTC-based acoustic modeling, the input for the neural network is a frame-wise feature sequence $\mathbf{X}$, whose length is much longer than that of the target subword sequence $\mathbf{s}$. Therefore, to compensate for difference of length between $\mathbf{s}$ and $\mathbf{X}$, an additional *blank* label $\emptyset$ is introduced into the set of subword units (CTC-label). Then, the posterior probability of CTC-label sequence $\mathbf{c} = \{c_1, ..., c_T\}$ given observation $\mathbf{X}$ is modeled by

[1] Some researchers have tried word-based CTC modeling, but they could not achieve good results [19].

http://dx.doi.org/10.21437/Interspeech.2016-71

the frame-wise product of the neural network's output as follows.

$$Pr(\mathbf{c}|\mathbf{X}) = \prod_{t=1}^{T} y_t^{c_t}. \quad (1)$$

Here, $y_t^{c_t}$ is the output score of the neural network for CTC-label c_t at time frame t.

Next, *collapsing function* $\Phi()$ is introduced to map frame-wise CTC-label sequence $\mathbf{c}$ into target subword sequence $\mathbf{s}$. This function converts the repetition of the CTC-label into one symbol, removing blank label $\emptyset$. For example, the CTC-label sequence "$AA\emptyset B\emptyset CC\emptyset$" and "$\emptyset A\emptyset BB\emptyset C\emptyset$" are both mapped to the subword sequence "ABC" by applying Φ. Based on the collapsing function, the posterior probability of subword sequence $\mathbf{s}$ given observation $\mathbf{X}$ is finally modeled as,

$$Pr(\mathbf{s}|\mathbf{X}) = \sum_{\mathbf{c}\in\Phi^{-1}(\mathbf{s})} Pr(\mathbf{c}|\mathbf{X}). \quad (2)$$

Here, Φ^{-1} is an inverse of the collapsing function, i.e., $\Phi^{-1}(\mathbf{s})$ indicates a set of possible CTC-label sequences that are mapped to $\mathbf{s}$ through Φ.

While arbitrary network architectures can be used for CTC-based models, we used a deep bidirectional long-short term memory (BLSTM) [25, 16] because of its high potential to represent the sequence properties. The neural network parameters can be trained using the CTC-loss function, which is derived from the principle of maximum likelihood [16].

2.2. Interpolation-based decoding for CTC-based AM

Previous papers used a naive interpolation with an LM score when using CTC-based AMs [17, 18, 19, 20, 21, 22]. In this framework, word sequence $\mathbf{W}$ given observation $\mathbf{X}$ is estimated as follows.

$$\tilde{\mathbf{W}} = \arg\max_{\mathbf{W}} \{\alpha \cdot \log Pr(\mathbf{s}|\mathbf{X}) + \log Pr(\mathbf{W})\}, \quad (3)$$

under the constraint of

$$\mathbf{s} \in \Psi(\mathbf{W}). \quad (4)$$

Here, Ψ is a function that converts word sequence $\mathbf{W}$ into a set of possible subword sequences $\mathbf{s}$. Further, $Pr(\mathbf{s}|\mathbf{X})$ is the CTC-AM probability introduced above, and $Pr(\mathbf{W})$ is a word-level LM (WLM) probability. The term α is a scaling factor for CTC-AM. Practically, a word insertion penalty (denoted as $|\mathbf{W}|$ in [17]) is often used in combination. In some papers [18, 22], a subword LM $Pr(\mathbf{s})$ is used instead of $Pr(\mathbf{W})$.

In contrast to the case of the DNN-HMM [1, 2], the posterior $Pr(\mathbf{s}|\mathbf{X})$ is often not normalized by the prior [17, 18, 21]. In the CTC-based speech recognition system EESEN [23], the authors propose normalizing the posterior by dividing by a prior $Pr(c_t)$ for each frame. In the experimental section, we also tested this normalization framework (in this paper, we call it the "EESEN prior" method).

2.3. WFST-based implementation

When decoding, $\mathbf{W}$ and $\mathbf{s}$ must satisfy the relation between the words and subwords (Eq. 4). The WFST-based decoding framework can be used to represent the search graph with such restrictions. In [23], the authors created a search graph by composing a token finite state transducer (FST) T, lexicon FST L and grammar WFST G, as follows.

$$T \circ min(det(L \circ G)) \quad (5)$$

Here, T is an FST that converts frame-wise CTC-label sequence $\mathbf{c}$ into corresponding subword sequence $\mathbf{s}$. Further, L is a FST that converts subword sequence $\mathbf{s}$ into word sequence $\mathbf{W}$. Finally, G is a grammar WFST that converts word sequence $\mathbf{W}$ into the same word sequence $\mathbf{W}$ with weight $Pr(\mathbf{W})$. The beam search algorithm is used to search for the best hypothesis given observation $\mathbf{X}$. See [23] for more detailed information.

3. Maximum a posteriori based decoding for CTC AM

3.1. Overview of the proposed framework

In the proposed decoding framework, the speech recognition problem is defined as the problem of finding word sequence $\mathbf{W}$ that maximizes posterior probability $Pr(\mathbf{W}|\mathbf{X})$ given observation $\mathbf{X}$, as follows.

$$\tilde{\mathbf{W}} = \arg\max_{\mathbf{W}} Pr(\mathbf{W}|\mathbf{X}). \quad (6)$$

Many readers will notice that this "maximum a posteriori (MAP)"-based decoding framework is the one that has been used in traditional HMM-based decoding. We then transform Eq. 6 so as to fit CTC-AM, not HMM, [2] as follows.

$$\begin{aligned} \tilde{\mathbf{W}} &= \arg\max_{\mathbf{W}} Pr(\mathbf{W}|\mathbf{X}) && (7)\\ &= \arg\max_{\mathbf{W}} \sum_{\mathbf{s}} Pr(\mathbf{W}|\mathbf{s}) Pr(\mathbf{s}|\mathbf{X})^{\alpha} && (8)\\ &\simeq \arg\max_{\mathbf{W}} Pr(\mathbf{W}|\mathbf{s}) Pr(\mathbf{s}|\mathbf{X})^{\alpha} && (9) \end{aligned}$$

Here, $Pr(\mathbf{s}|\mathbf{X})$ is the CTC-AM probability and α is its scaling factor. Sequences $\mathbf{s}$ and $\mathbf{W}$ must satisfy Eq. 4, as in the conventional interpolation-based framework. In Eq. 9, the Viterbi approximation is used to remove the summation by $\mathbf{s}$.

The new term $Pr(\mathbf{W}|\mathbf{s})$ is calculated as follows.

$$Pr(\mathbf{W}|\mathbf{s}) = \frac{Pr(\mathbf{s}|\mathbf{W})Pr(\mathbf{W})}{Pr(\mathbf{s})^{\beta}}. \quad (10)$$

Here, $Pr(\mathbf{s})$ is a subword LM (SLM) probability and β is its scaling factor. SLM probability $Pr(\mathbf{s})$ can be calculated using conventional language modeling techniques like N-gram or RNN. SLM is trained by a subword corpus, which is easily created by applying a word-subword conversion to the text corpus.

The term $Pr(\mathbf{s}|\mathbf{W})$ is a word-subword conversion probability. Note that the conversion from word to subword is often a one-to-one mapping (e.g., a mapping from a word to characters). In such a case, $Pr(\mathbf{s}|\mathbf{W})$ becomes one under the constraint of Eq. 4, and Eq. 10 can be simplified to $Pr(\mathbf{W}|\mathbf{s}) = \frac{Pr(\mathbf{W})}{Pr(\mathbf{s})^{\beta}}$.

[2] In the HMM-based framework, Eq. 6 is transformed into $\tilde{\mathbf{W}} = \arg\max_{\mathbf{W}} \frac{Pr(\mathbf{X}|\mathbf{W})Pr(\mathbf{W})}{Pr(\mathbf{X})} \simeq \arg\max_{\mathbf{W}} \frac{Pr(\mathbf{X}|\mathbf{s})Pr(\mathbf{s}|\mathbf{W})Pr(\mathbf{W})}{Pr(\mathbf{X})}$, where $\mathbf{s}$ indicates an HMM-state sequence.

3.2. WFST-based implementation

WFST can be used to realize the search graph for Eq. 9, as in the case of conventional interpolation-based decoding. The search graph for the proposed decoding framework is created as follows.

$$T \circ min(det(S^{-\beta} \circ L \circ G)). \quad (11)$$

Here, T and G are the same as in Eq. 5. The term $S^{-\beta}$ is a new WFST that converts a subword sequence into the same subword sequence with weight $Pr(\mathbf{s})^{-\beta}$, which can be created in a similar manner to G [26]. [3] Finally, L is the lexicon WFST that converts subword sequence $\mathbf{s}$ into word sequence $\mathbf{W}$ with weight $Pr(\mathbf{s}|\mathbf{W})$. Compared with the conventional search graph (Eq. 5), the difference can be summarized in the additional composition of $S^{-\beta}$ and the incorporation of probabilistic score $Pr(\mathbf{s}|\mathbf{W})$ into L. [4]

4. Experiment

4.1. WSJ experiment

4.1.1. Experimental settings

The first experiment was conducted on the Wall Street Journal (WSJ) corpus, known as LDC93S6B and LDC94S13B. We followed the experimental settings in [23] by using the EESEN software [5] developed by the authors of the paper.

The training data was prepared according to the recipe in EESEN, which gave us 77.5 hours of training data with 3.8 hours of cross-validation data. A phoneme-based BLSTM with four hidden layers, each comprising 320 nodes, was trained on the 120-dimensional filter-bank features (40 filter-bank features, delta coefficients, and delta-delta coefficients) with mean and variance normalization. The BLSTM was trained from scratch based on the CTC-loss function. The initial learning rate and momentum parameter were set to 0.00004 and 0.95, respectively.

For the word-level LM, the WSJ standard pruned trigram LM was used. In addition, we trained a subword-level LM for the proposed MAP-based decoding. Here, a phoneme N-gram with Good-Turing smoothing was trained using the phoneme-converted transcription of the training data for AMs. When decoding, scaling factors α and β were optimally tuned. Basically speaking, α with a value of around 0.9 to 1.1 and β with a value of around 0.4 to 0.6 gave us good results.

4.1.2. Results

For consistency with previous works, we report the results on the "eval92" evaluation data. The results are listed in Table 1. The first two rows present the results of the DNN-HMM baseline (six hidden layers, each of 1024 nodes) and the CTC-BLSTM model, both were reported in [23]. The last three rows indicate the results with the CTC-BLSTM that we newly trained for this experiment. By comparing the second and the fourth rows, we confirm that we successfully reproduced the results reported in [23], with a slight improvement of the word error rate (WER). [6]

Table 1: *WER of various networks for WSJ eval92.*

Acoustic Model	Framework	WER (%)
DNN-HMM [23]	-	7.14
CTC-BLSTM [23]	EESEN prior [23]	7.87
CTC-BLSTM (reprod.)	Interpolation (Eq. 3)	8.56
	EESEN prior [23]	7.66
	MAP (proposed)	7.25

Table 2: *Relation between SLM type and WER on the WSJ eval92 dataset.*

SLM type	no use	1-gram	2-gram	3-gram
SLM perplexity	-	35.4	18.5	11.7
WER (%)	8.56	7.39	**7.25**	7.48

When we used the naive interpolation-based framework (the third row of Table 1), the WER was 8.56 %, which is much worse than that of the DNN-HMM baseline (7.14% WER). By applying the prior proposed in EESEN [23], the WER was improved to 7.66%; however the WER was still much worse than that of the DNN-HMM baseline. Finally, when we used our proposed MAP-based decoding framework with a 2-gram SLM, the WER was further improved to 7.25%, which is very close to the DNN-HMM result. The relative improvement from the interpolation-based method is 15.3%.

To understand the detailed effects of the incorporation of the SLM, we tested various N-gram orders of the SLM. The results are listed in Table 2. As shown in the table, incorporation of the SLM significantly improved the WER, and the 2-gram SLM obtained the best results. We noticed that the 3-gram SLM obtained a slightly worse result than that of the 2-gram, although the result was still much better than that of the conventional interpolation-based method. Our interpretation of this phenomenon is as follows: the SLM works as a coordinator between the CTC-AM and WLM, and the best setting is determined by the balance between the two models. In this experiment, CTC-AM would learn the 2-gram level knowledge from the training data; therefore, the 2-gram SLM achieved the best result.

4.2. CSJ experiment

4.2.1. Experimental settings

As an experiment with a larger set of training data, we also tested our proposed method on the "Corpus of Spontaneous Japanese (CSJ) [27]", which consists of over 600 hours of lecture recordings. The corpus contains three official evaluation sets (E1, E2, and E3), each comprising 10 lecture recordings. In addition to the three evaluation sets, we picked up 10 lecture recordings as the development set to tune the system parameters. Finally, the rest of the data in CSJ (586 hours of speaker-open lecture recordings) was used as the training data. The entire 586-hour data or its 240-hour subset was used for training.

As the baseline model, a DNN acoustic model with five hidden layers, each comprising 2048 nodes, was trained. The output layer had about 8,500 nodes [7], which corresponded to

[3] In most experiments, we used an *exact* representation for SLM WFST instead of an *approximated* representation with failure transitions [26]. Only in the experiment of 3-gram SLM on the CSJ testset, we used the approximated version to save the memory.

[4] As described in the previous section, $Pr(\mathbf{s}|\mathbf{W})$ is often one. In such a case, the difference between the conventional and proposed frameworks is summarized in the composition of $S^{-\beta}$.

[5] https://github.com/srvk/eesen

[6] Although the original momentum setting in EESEN was 0.9, we found that a momentum of 0.95 gave us slightly better results, which we report here. With a momentum of 0.9, we obtained a WER of 7.80%.

[7] The number of HMM states slightly varied depending on the train-

Table 3: *WER for the CSJ test set with 240-hour training data.*

Acoustic Model	Decoding Framework	WER (%)			
		E1	E2	E3	E (avg.)
DNN-HMM	-	**12.65**	**10.19**	14.28	12.37
CTC-BLSTM	Interpolation	13.90	10.95	14.70	13.18
	EESEN prior [23]	14.19	11.33	16.49	14.00
	MAP (proposed)	12.94	10.33	**13.11**	**12.13**

Table 4: *WER for the CSJ test set with 586-hour training data.*

Acoustic Model	Decoding Framework	WER (%)			
		E1	E2	E3	E (avg.)
DNN-HMM	-	12.19	9.80	11.01	11.00
CTC-BLSTM	Interpolation	12.81	9.75	10.67	11.08
	EESEN prior [23]	12.40	10.00	11.36	11.25
	MAP (proposed)	**11.90**	**9.24**	**9.65**	**10.26**

the context-dependent phoneme HMM states. As acoustic features, 72-dimensional filter-bank features (24 filter-bank features, delta coefficients and delta-delta coefficients) with mean and variance normalization per speaker were used. We concatenated the features of both the previous and following seven frames (15 frames of features in total) when inputting them to the DNNs. The DNN was initialized using the discriminative pre-training method [28] and was fine-tuned using the standard stochastic gradient descent based on the cross-entropy loss criterion.

The CTC-BLSTM was then trained based on the same 72-dimensional filter-bank features with no splicing. In this experiment, we used 263 Japanese syllables (known as "kana") for the recognition unit. A BLSTM with five hidden layers, each comprising 320 nodes, was used. The BLSTM was trained based on the CTC-loss function from scratch. The initial learning rate and momentum parameter were set to 0.00004 and 0.9, respectively.

In addition to the AMs, we trained a 4-gram WLM from the transcription of the 586 hours of training data, with Kneser–Ney smoothing [29]. The vocabulary size of the WLM was 77K. Finally, we trained an SLM for the proposed MAP-based decoding. A syllable N-gram was trained with Kneser–Ney smoothing from the syllable-level transcription of the training data. When decoding, we tuned the scaling factors α, β and word insertion penalty using the development set. The best parameters were then used to decode the evaluation sets.

4.2.2. Results

The results with the 240-hour training data and the entire 586-hour training data are listed in Tables 3 and 4, respectively. In these experiments, we used a 2-gram SLM for our proposed MAP-based decoding framework.

From the results in the tables, various things can be observed. First, as long as the conventional interpolation-based method was used, CTC-BLSTM could not surpass the DNN-HMM. However, the difference between DNN-HMM and CTC-BLSTM (with the interpolation-based decoding) became smaller when we used the entire 586 hours of training data, so one may expect that the CTC-BLSTM could surpass the DNN-HMM if additional training data was used.

Second, contrary to the experiment on the WSJ, the EESEN prior degraded the WER. There was only one exception for which the EESEN prior improved the WER (E1 test set with 586 hours of training data). We believe that these inconsistent results were caused by the lack of theory behind the interpolation-based method and the EESEN prior.

Lastly, our proposed MAP-based decoding method improved the WER from the interpolation-based method in all experimental settings, which finally obtained a better result than the strong DNN-HMM baseline. With 586 hours of training data, CTC-BLSTM achieved the best results for all test sets, showing an average of 6.7% better results than the DNN-HMM. It is important to note that the parameter size of the CTC-BLSTM was much smaller than that of the DNN-HMM because the number of output nodes was much smaller in CTC-BLSTM. It is also noteworthy that the training pipeline of CTC-BLSTM was much simpler than that of DNN-HMM; we only did the CTC-loss-based training from scratch.

We tested various N-gram orders for SLM. The results are listed in Table 5. Just as in the WSJ experiment, the 2-gram SLM achieved the best results, while the incorporation of SLM always improved the WER. A detailed analysis of this phenomenon is one of our future works.

Table 5: *Relation between SLM type and WER (avg.) on CSJ (586-hour training data).*

SLM type	no use	1-gram	2-gram	3-gram
SLM perplexity	-	63.1	32.7	18.1
WER (%)	11.08	10.40	**10.26**	10.62

5. Conclusion

In this paper, we proposed a novel decoding framework for CTC-AMs. In the proposed framework, a subword-based LM was newly introduced to coordinate the CTC-based AM and word-level LM scores. The search graph is represented by the WFST, so it is easy to extend the conventional interpolation-based framework to our framework. In the experiments on the WSJ and CSJ, our proposed framework consistently achieved 7.4 – 15.3 % improvements over the previous interpolation-based framework. In the CSJ experiment with 586 hours of training data, the CTC-based AM finally achieved a 6.7 % better WER than the the DNN-HMM baseline.

6. References

[1] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks." in *Proc. INTERSPEECH*, 2011, pp. 437–440.

[2] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. SAP*, vol. 20, no. 1, pp. 30–42, 2012.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. 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," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[4] N. Kanda, R. Takeda, and Y. Obuchi, "Elastic spectral distortion for low resource speech recognition with deep neural networks," in *Proc. ASRU*, 2013, pp. 309–314.

[5] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *Proc. ICASSP*, 2013, pp. 8619–8623.

[6] L. Deng, O. Abdel-Hamid, and D. Yu, "A deep convolutional neural network using heterogeneous pooling for trading acoustic in-

ing data set. It was 8,522 for the 240-hour subset data and 8,407 for the entire 586-hour training data.

variance with phonetic confusion," in *Proc. ICASSP*, 2013, pp. 6669–6673.

[7] O. Vinyals, S. V. Ravuri, and D. Povey, "Revisiting recurrent neural networks for robust ASR," in *Proc. ICASSP*, 2012, pp. 4085–4088.

[8] C. Weng, D. Yu, S. Watanabe, and B.-H. F. Juang, "Recurrent deep neural networks for robust speech recognition," in *Proc. ICASSP*, 2014, pp. 5532–5536.

[9] G. Saon, H. Soltau, A. Emami, and M. Picheny, "Unfolded recurrent neural networks for speech recognition," in *Proc. INTERSPEECH*, 2014.

[10] N. Kanda, M. Tachimori, X. Lu, and H. Kawai, "Training data pseudo-shuffling and direct decoding framework for recurrent neural network based acoustic modeling," in *Proc. ASRU*, 2015, pp. 15–21.

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

[12] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *arXiv e-prints*, 2014.

[13] H. Sak, O. Vinyals, G. Heigold, A. Senior, E. McDermott, R. Monga, and M. Mao, "Sequence discriminative distributed training of long short-term memory recurrent neural networks," *entropy*, vol. 15, no. 16, pp. 17–18, 2014.

[14] J. T. Geiger, Z. Zhang, F. Weninger, B. Schuller, and G. Rigoll, "Robust speech recognition using long short-term memory recurrent neural networks for hybrid acoustic modelling," in *Proc. INTERSPEECH*, 2014.

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

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

[17] A. L. Maas, A. Y. Hannun, D. Jurafsky, and A. Y. Ng, "First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns," *arXiv preprint arXiv:1408.2873*, 2014.

[18] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deepspeech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[19] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *Proc. INTERSPEECH*, pp. 1468–1472.

[20] 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.

[21] A. Senior, H. Sak, F. de Chaumont Quitry, T. N. Sainath, and K. Rao, "Acoustic modelling with CD-CTC-SMBR LSTM RNNs," in *Proc. ASRU*, 2015, pp. 604–609.

[22] A. L. Maas, Z. Xie, D. Jurafsky, and A. Y. Ng, "Lexicon-free conversational speech recognition with neural networks," in *Proc. NAACL HTL*, 2015.

[23] 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.

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

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

[26] C. Allauzen, M. Mohri, and B. Roark, "Generalized algorithms for constructing statistical language models," in *Proc. ACL*, 2003, pp. 40–47.

[27] K. Maekawa, "Corpus of spontaneous japanese: Its design and evaluation," in *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[28] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. ASRU*, 2011, pp. 24–29.

[29] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech & Language*, vol. 13, no. 4, pp. 359–393, 1999.