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

Joint Decoding of CTC Based Systems for Speech Recognition

Jiaqi Guo[1], *Yongbin You*[2], *Yanmin Qian*[1], *Kai Yu*[1]

[1]MoE Key Lab of Artificial Intelligence
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
[2]AISpeech Ltd, Suzhou China

{guojiaqi, yanminqian, kai.yu}@sjtu.edu.cn, yongbin.you@aispeech.com

Abstract

Connectionist temporal classification (CTC) has been successfully used in speech recognition. It learns the alignments between speech frames and label sequences automatically without explicit pre-generated frame-level labels. While this property is convenient for shortening the training pipeline, it may become a potential disadvantage for the frame-level system combination due to inaccurate alignments. In this paper, a novel Dynamic Time Warping (DTW) based position calibration algorithm is proposed for joint decoding on two CTC based acoustic models. Furthermore, joint decoding for CTC and conventional hybrid NN-HMM models is also explored. Experiments on a large vocabulary Mandarin speech recognition task show that the proposed joint decoding of both CTC based and CTC-Hybrid based systems can achieve a significant and consistent character error rate reduction.

Index Terms: Joint decoding, Dynamic time warping, Connectionist temporal classification, Hybrid system, System combination

1. Introduction

End-to-end speech recognition is a recently proposed approach that directly transcribes speech to text without requiring predefined alignment between acoustic frames and characters [1, 2, 3, 4, 5, 6]. Works on end-to-end speech recognition can be categorized into two main approaches: Connectionist Temporal Classification (CTC) [7, 1, 2, 3] and attention-based encoder-decoder [8, 4, 5, 6]. Both methods learn a mapping between variable-length input and output sequences. The key idea of CTC is to use an intermediate label representation allowing repetitions of labels and occurrences of blank labels to identify no output label. The attention-based encoder-decoder directly learns a mapping from acoustic frames to corresponding character sequences. At each output time step, the model emits a character conditioned on the inputs as well as the history of the target character.

Compared to the attention-based encoder-decoder model, CTC differs in two aspects: i) the lengths of neural network outputs are the same among different systems, which equals to the length of acoustic frames. ii) the alignments between input frames and labels are strictly monotonic as in hidden markov models (HMM). These two properties enable an acoustic probability fusion between two CTC based systems as well as a fusion between CTC and HMM based systems. However, the fusion of any two systems is not straightforward, since the posterior sequences are not synchronized in addition to different modelling units in case of CTC and HMM. In order to alleviate these problems, this work firstly introduces a novel method based on Dynamic Time Warping (DTW) to synchronize different score sequences. Further, an acoustic score fusion method is proposed to combine aligned sequences. Furthermore for CTC based and hybrid NN-HMM system combination, a state-phone mapping is introduced to unify the model units. Lastly, an attenuated method to solve the absence of "blank" label outputs in a hybrid HMM system is evaluated.

Many works have been proposed for system combination. In ROVER [9], the 1-best word sequence of several speech recognizers are aligned and a single word transcription network (WTN) is built. The best scoring word is selected at each node. In [10], the generated lattice from each system is compressed into a structure called confusion network where the most likely word is picked at each position. The approach in [11], multiple lattices generated for the same utterance from multiple systems are combined. The optimization procedure is conducted to minimize the averaged Bayes Risk with respect to the Levenshtein distance over multiple systems, then 1-best path for each utterance is generated.

In comparison with approaches described above, joint decoding is more efficient. Instead of requiring a decoding procedure for each system individually, it performs only one single decoding stage with fused acoustic scores. In addition, the performance of the weighted combination in state-level acoustic log likelihoods is even better than 1-best or lattice level fusion as reported in our early works[12, 13].

The rest of this paper is organized as follows: In Section 2, we revisit CTC speech recognition systems and explain why unsynchronization between different CTC based systems exists. In Section 3, we present how DTW is adapted to the joint decoding of CTC-based systems. The combination between CTC and hybrid based systems is described in Section 4. In Section 5, the proposed approaches are evaluated and experimental results and analysis are given. Finally, we draw our conclusions in Section 6.

2. Connectionist Temporal Classification

The key idea of CTC [7] is to use intermediate label representation $\boldsymbol{\pi} = (\pi_1, ..\pi_T)$, allowing repetitions of labels and occurrences of a blank label $(-)$, which represents the special emission without labels, i.e., $\pi_t \in \{1, ..., K\} \cup \{-\}$. CTC trains the model to maximize $P(\boldsymbol{l}|\boldsymbol{x})$, the probability distribution over all possible label sequences $\mathcal{B}^{-1}(\boldsymbol{l})$:

$$P(\boldsymbol{l}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\boldsymbol{l})} P(\boldsymbol{\pi}|\boldsymbol{x}) \tag{1}$$

where $\mathcal{B}$ is a many-to-one map: $\mathcal{L'}^T \mapsto \mathcal{L}^{\leq T}$, where $\mathcal{L}^{\leq T}$ is the set of possible labellings (i.e. the set of sequences of length

http://dx.doi.org/10.21437/Interspeech.2019-2026

less than or equal to T over the original alphabet $\{1, ..., K\}$). The mapping is done by removing all blanks and repeated labels from the paths(e.g. $\mathcal{B}(a - ab-) = \mathcal{B}(-aa - abb) = aab$).

CTC is generally applied on top of Recurrent Neural Networks (RNNs). Each RNN output unit is interpreted as the probability of observing the corresponding label at a particular time. The probability of label sequence $P(\boldsymbol{\pi}|\boldsymbol{x})$ is modeled as being conditionally independent to the network outputs product:

$$P(\boldsymbol{\pi}|\boldsymbol{x}) \approx \prod_{t=1}^{T} P(\pi_t|\boldsymbol{x}) = \prod_{t=1}^{T} y_{\pi_t}^t \tag{2}$$

where $y_{\pi_t}^t$ denotes the softmax activation of label π_t within the RNN output y at time t.

The objective function to be minimised is defined as the negative log likelihood of the ground truth label sequence $\boldsymbol{l}^*$. i.e.,

$$\mathcal{L}_{CTC} \triangleq -\ln P(\boldsymbol{l}^*|\boldsymbol{x}) \tag{3}$$

The probability distribution $P(\boldsymbol{l}|\boldsymbol{x})$ can be computed efficiently using the forward-backward algorithm as

$$P(\boldsymbol{l}|\boldsymbol{x}) = \sum_{u=1}^{|\boldsymbol{l'}|} \frac{\alpha_t(u)\beta_t(u)}{y_{l'_u}^t} \tag{4}$$

where l' is a modified label sequence of l, which is made by inserting blank symbols between each label and the beginning and the end of a sequence allowing for blanks in the output (i.e., $l = (c, a, t)$, $l' = (-, c, -, a, -, t, -)$). $\alpha_t(u)$ is the forward variable, representing the total probability of all possible prefixes ($l'_{1:u}$) that end with the u-th label, and $\beta_t(u)$ is the backward variable of all possible suffixes ($l'_{u:U}$) that start with the u-th label. The network can then be trained with standard backpropagation by taking the derivative of the loss function with respect to y_k^t for any k label including the blank.

Since the label probabilities used for CTC are assumed to be conditioned on the entire input sequence, therefore in cases where the network is unidirectional it must wait until after a given input segment is sufficiently complete to be identified before emitting the corresponding label [14]. Thus with various emitting confidence of current input segments, different acoustic neural networks will activate their outputs at different frames. Consequently, this unsychronization becomes an obstacle for frame-level posterior based system combination in speech recognition.

3. Joint decoding of CTC based systems

3.1. Dynamic Time Warping algorithm

Dynamic Time Warping (DTW) algorithm is a popular processing method in automatic speech recognition, time series analysis, and many other pattern matching applications. It measures the similarity between two temporal sequences, which may vary in speed, and "warp" the time axis of one (or both) sequences to achieve an optimal alignment [15]. Figure 1 shows a simple example of the DTW algorithm.

3.2. Joint decoding of CTC based systems using DTW

Here, we adapt DTW to address the unsychronization issue in Section 2. Instead of other applications of DTW to detect desired patterns from target temporal sequence, we use DTW to generate acoustic posterior alignments, against which the combination is performed to deliver the sharing posterior sequence, between two acoustic models. The DTW algorithm is adapted in a multi-dimension and local constraint way as described in Algorithm 1

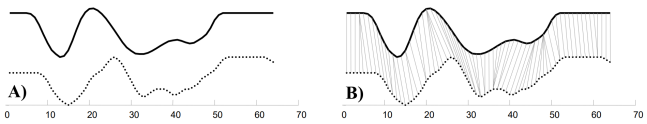

Figure 1: *An example of the utility of dynamic time warping. A) The sequences have an overall similar shape, but they are not aligned in the time axis. The distance between the i^{th} point on one sequence and the i^{th} point on the other sequence is large. B) DTW can efficiently find an alignment between the two sequences leading to a more accurate distance measure [16].*

Algorithm 1 Dynamic Time Warping algorithm with locality constraint for joint decoding of CTC based systems

```
distance := DISTANCE_MEASURE
w := CONSTRAINT_WINDOW_SIZE
DTW := array [0..n, 0..n]
for i := 0 to n do
    for j := 0 to n do
        DTW[i, j] := inifinity
    end for
end for
DTW[0,0] := 0
for i := 1 to n do
    for j := max(1, i − w) to min(n, i + w) do
        cost := distance(S[i], T[j])
        DTW[i, j] := cost + minimum(DTW[i − 1, j],
        DTW[i, j − 1], DTW[i − 1, j − 1])
    end for
end for
```

where S, T are acoustic posterior sequences from different CTC systems.

$$S = s_1, s_2, ..., s_i, ..., s_n \tag{5}$$

$$T = t_1, t_2, ..., t_j, ..., t_n \tag{6}$$

And *distance* and *w* are two hyperparamters. For *distance*, symmetric KullbackLeibler divergence is utilized in this paper. The constraint window size w was set to restrict matching within range $[-w, +w]$ with respect to the current position.

By backtracing the DTW matrix, an alignment between two posterior sequences can be achieved. Then, element-wise fusion can be performed. Compared to the straight forward joint decoding method, there is only one mapping(*one-to-one* mapping) between the elements to be combined, here are three alternative mappings in achieved DTW alignment, *one-to-one*, *one-to-many*, *many-to-one* for the element of one sequence with respect to its peer(s) in the other sequence. Considering all these possible DTW alignment mappings, we propose a score combination approach in a *compact* way, as illustrated in Figure 2, where elements with *many-to-one* mapping are equally averaged with its siblings to meet their sharing peer in the other sequence. The length of newly generated sequence will be shorter than the originals.

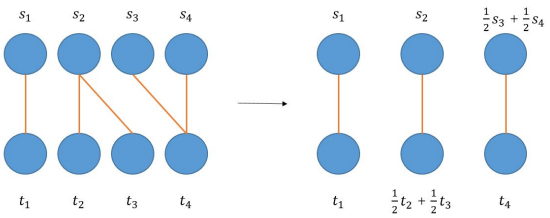

Figure 2: *Acoustic score merge method according to DTW alignments on the CTC system outputs in a compact way*

4. Joint decoding of CTC-Hybrid based systems

Compared to joint decoding among CTC-based systems, the acoustic combination between CTC and hybrid systems is slightly more complicated, because the acoustic modeling units differ between these two systems. Phone and character units are usually used in the CTC model directly, while tied senone states are commonly utilized for hybrid systems. However, if we properly design the decision tree of the state clustering, letting the tree root correspond to an individual phone, the mapping between states and phone set becomes many to one, i.e. several senone states may map to the same phone. Here we calculate the posterior of particular phone in hybrid neural networks by choosing the maximum probability among all its corresponding candidate states. For example, there are K senone states affiliated with phone y_u, and then the selected posterior representing the phone y_u is

$$P_{hyb}(y_u|\mathbf{x}) = max\{P(s_1^{y_u}|\mathbf{x}), P(s_2^{y_u}|\mathbf{x}), ...P(s_K^{y_u}|\mathbf{x})\} \quad (7)$$

Though averaging across candidate states to represent the phone y_u is also applicable, we deem maximum value of these states preferable. Suppose there are only two phones, i.e. m, n in phone set with 3 and 4 candidate states respectively. And the probabilities of their candidate states are:

$$P(\mathbf{s}^{y_m}|\mathbf{x}) = \{0.15, 0.18, 0.16\} \quad (8)$$
$$P(\mathbf{s}^{y_n}|\mathbf{x}) = \{0.01, 0.02, 0.43, 0.05\} \quad (9)$$

It is obvious that the output y is more like a phone n, since the confidence of its third states is much larger than any state's confidence in phone m. If we use the averaged probability, $P(y_m|\mathbf{x})$ would be larger. While if we use maximum, phone n would be more likely picked up. So maximum should be the more reasonable choice and the results in the experiments also confirm our conjunction.

In addition to the units mapping, we also have to address the symbol *blank* in CTC, which does not exist in the hybrid model. Inspired by [17], we use the *blank* probability in CTC as a "gate" to control the mapped phone posterior from the hybrid system, and the acoustic score combination is only performed on the non-blank phones. The acoustic combination between CTC and hybrid acoustic models is illustrated in Figure 3, where the newly generated posterior is calculated as:

$$P_{new}(\mathbf{y}|\mathbf{x}) = \frac{P_{ctc}(\mathbf{y}|\mathbf{x}) + \alpha \cdot (1 - P_{ctc}(\mathbf{y}_{blk}|\mathbf{x})) \cdot P_{hyb}(\mathbf{y}|\mathbf{x})}{N} \quad (10)$$

where the $P_{ctc}(\mathbf{y}|\mathbf{x})$ is the original phone acoustic score from the CTC model, $P_{hyb}(\mathbf{y}|\mathbf{x})$ is the mapped phone probability from the hybrid system, α is its weight in combination. In this way, the acoustic score $P_{hyb}(\mathbf{y}|\mathbf{x})$ from the hybrid system can be properly attenuated with $(1 - P_{ctc}(\mathbf{y}_{blk}|\mathbf{x}))$. And N is L1 normalization factor to make newly generated score a probability.

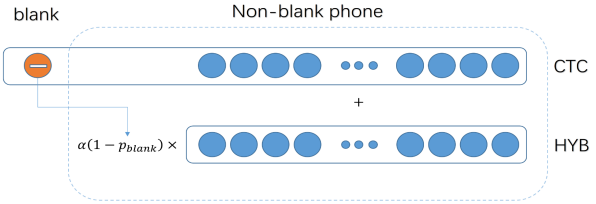

Figure 3: *Illustration of using CTC blank probability as a "gate" to control the phone posteriors from the hybrid system.*

5. Experiments

5.1. Data and Experiment Setup

In this work, we evaluated our proposed joint decoding methods on a large vocabulary Mandarin speech recognition task. The training set for all involved systems is 2000 hours of transcribed data extracted from an online speech recognition service. For evaluation, we construct 3 data sets: *test-A* which composes of transcribed data from the previously mentioned online speech recognition service, consisting of 6382 utterances. *test-B*, a data set collected from in-car voice assistant applications with 5240 utterances, and *test-C*, a fast test set with 667 utterances in total, in which speakers were asked to speak faster than usual. We also constructed a development set from the online service as *test-A*, including 2127 utterances, to tune α in Equation 10 and the weights of different recognition systems for all combination methods in our experiments. During decoding stages, all acoustic models are decoded with a single pass WFST-based decoder which uses a 3-gram language model.

For experiments, we prepared four recognition systems, CTC-DFSMN [18], CTC-CLD [19], CTC-LSTM [7] and HYB-LSTM [20, 21]. The input features are 40-dimensional log-mel filterbanks computed on a 25ms window with 10ms shift. The outputs for all CTC based systems are 121 in dimension with 120 phone and a blank as targets. The output for hybrid system is size of 9663, indicating the probability over context dependent state targets.

CTC-DFSMN is trained with an $11 \times 40 - 8 \times [2048 - 512(20; 0; 2; 0)] - 3 \times 2048 - 512 - 121$ architecture. The input is spliced with a context window of 11(5+1+5). The Layers are 8 DFSMN components, 3 full-connected ReLU layers and 1 projection softmax layer size of 121. For a more detailed description of the architecture, we refer the reader to [18]. For CTC-CLD, the same context window as CTC-DFSMN is used as input. The network layers are a convolution layer with filter size of 9×8 and 256 feature maps, a maxpooling layer with window size of 1×3 followed by 4 projection LSTM layers [22] with 1536 cells and 320 in project dimension, sequentially, 2 full-connected ReLU layers and 1 softmax layer size of 121 as output. For CTC-LSTM and HYB-LSTM, the inputs are 40-dimensional log-mel filterbanks, the layers are 3 projection LSTM layers and 1 softmax layer with 121 and 9663 in dimension respectively. The lower frame rate [23] of 30ms is adapted in all neural networks. All these models were trained

using KALDI [24] and EESEN [3]. The performance of these baseline systems are shown as Table 1.

Table 1: *CER (%) comparison of the CTC and hybrid baselines*

Baseline Model	test-A	test-B	test-C
CTC-DFSMN	14.24	13.56	21.24
CTC-CLD	14.71	14.35	22.59
CTC-LSTM	15.54	15.10	23.22
HYB-LSTM	14.22	13.59	22.18

5.2. Evaluation on joint decoding of CTC-based Systems

The proposed joint decoding approach is performed and denoted as the symbol $\otimes$, the w parameter in DTW is chosen as 1 since we assume the match window should be around [-30ms, 30ms]. For comparison, the normal Kaldi minimum Bayes risk (MBR) lattice combination and straight forward joint decoding are also applied and denoted as the symbol $\oplus$ and $naive$ respectively. The experiment results are shown as Table 2.

Table 2: *CER (%) comparison of different system combination approaches between CTC systems. $\oplus$ indicates the normal MBR lattice combination using Kaldi, and $\otimes$ indicates the joint decoding with different modes*

Model	Comb Mode	test-A	test-B	test-C
CTC-DFSMN	-	14.24	13.56	21.24
CTC-DFSMN $\oplus$ CTC-CLD	MBR lattice-comb	13.60	**12.62**	19.89
CTC-DFSMN $\otimes$ CTC-CLD	naive	13.58	12.93	20.35
	DTW	**13.46**	12.75	**19.88**
Model	**Comb Mode**	**test-A**	**test-B**	**test-C**
CTC-CLD	-	14.71	14.35	22.59
CTC-CLD $\oplus$ CTC-LSTM	MBR lattice-comb	14.24	**13.14**	20.17
CTC-CLD $\otimes$ CTC-LSTM	naive	14.49	13.65	20.80
	DTW	**13.91**	13.22	**20.10**

In *test-A*, the performance of *naive* mode joint decoding of CTC-DFSMN and CTC-CLD is better than MBR lattice-comb method. In our hypothesis, there are two factors, firstly, the data source of *test-A* and the training data are identical. Secondly, the context window of input in these two systems are the same. These make their alignments more accurate and the sychronization between them fairly fine. However, the proposed DTW joint decoding achieves even better CER to 13.46, which is 18% (0.66 to 0.78) relative improvement compared with *naive* mode. In *test-B* and *test-C*, DTW alignments could consistently help improve the performance of the *naive* mode, and its effect is more significant in the fast *test-C*, in which the confidence of activation in each frame is more various between two systems. Besides that, when the "vision" of two CTC based system are not same, such as CTC-CLD and CTC-LSTM having individual input context windows, the sychronization provided by DTW presents to be more effective in their joint decoding as shown in the lower part of Table 2.

5.3. Evaluation on joint decoding of CTC-Hybrid based systems

As to the joint decoding between CTC and Hybrid based systems, firstly we derived the mapping between the tied states and phone set from the transition model trained by Kaldi, and then perform the mapping and acoustic score fusion as described in Section 4. The parameter α in Equation 10 is investigated across the range $[0.1 - 0.3]$ and tuned on the development set. All the experimental results are listed in Table 3.

Table 3: *CER (%) comparison of different system combination approaches between CTC & Hybrid systems. $\oplus$ indicates the normal MBR lattice combination using Kaldi, and $\otimes$ indicates the joint decoding with different modes*

Model	Mode	Map	test-A	test-B	test-C
Single Best	-	-	14.22	13.56	21.24
CTC-DFSMN $\oplus$ HYB-LSTM	MBR lattice-comb	-	14.04	**12.94**	**20.43**
CTC-DFSMN $\otimes$ HYB-LSTM	naive	max	**13.83**	13.18	21.23
		ave	14.18	13.18	21.38

As shown in Table 3, in *test-A* and *test-B*, the joint decoding between CTC and hybrid based systems delivered a considerate improvement on CER, while it took little effective in *test-C*. And the *maximum* is proven to be the more premium mapping in comparision with the *average* solution.

6. Conclusion

In this work, we proposed a system combination method using joint decoding between CTC-based and CTC-Hybrid based systems. For the CTC-based systems, a DTW algorithm is firstly performed to align the CTC outputs, and then an appropriate acoustic probability combination method is proposed to generate a new acoustic score sequence for decoding. Moreover, the joint decoding between CTC & Hybrid systems is also designed with modeling units mapping and accurate acoustic calculation. Experimental results show that the newly proposed system combination approach between two CTC based systems can get significant and consistent improvements compared to the straight forward joint decoding, and it is also competitive with the conventional MBR lattice combination. And joint decoding method between CTC and hybrid system could be applied to leverage already finetuned conventional hybrid acoustic neural networks to improve the performance of CTC recognition systems.

7. Acknowledgements

This work has been supported by the National Key Research and Development Program of China (Grant No.2017YFB1002102), the China NSFC project (No. 61603252 and No. U1736202). Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

8. References

[1] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning*, 2014, pp. 1764–1772.

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

[3] Y. Miao, M. Gowayyed, and F. Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 167–174.

[4] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end

continuous speech recognition using attention-based recurrent nn: First results," *arXiv preprint arXiv:1412.1602*, 2014.

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

[6] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[7] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[9] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*. IEEE, 1997, pp. 347–354.

[10] G. Evermann and P. Woodland, "Posterior probability decoding, confidence estimation and system combination," in *Proc. Speech Transcription Workshop*, vol. 27. Baltimore, 2000, p. 78.

[11] H. Xu, D. Povey, L. Mangu, and J. Zhu, "Minimum bayes risk decoding and system combination based on a recursion for edit distance," *Computer Speech & Language*, vol. 25, no. 4, pp. 802–828, 2011.

[12] Y. Qian and P. C. Woodland, "Very deep convolutional neural networks for robust speech recognition," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 481–488.

[13] T. Tan, Y. Qian, H. Hu, Y. Zhou, W. Ding, and K. Yu, "Adaptive very deep convolutional residual network for noise robust speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 8, pp. 1393–1405, 2018.

[14] A. Graves, "Supervised sequence labelling with recurrent neural networks," *Stud Comput Intell*, vol. 385, 2012.

[15] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370.

[16] E. Keogh and M. Pazzani, "Derivative dynamic time warping," *SIAM International Conference on Data Mining 2001*, 2001.

[17] S. Zhang and M. Lei, "Acoustic modeling with dfsmn-ctc and joint ctc-ce learning," *Proc. Interspeech 2018*, pp. 771–775, 2018.

[18] S. Zhang, M. Lei, Z. Yan, and L. Dai, "Deep-fsmn for large vocabulary continuous speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5869–5873.

[19] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4580–4584.

[20] D. J. Kershaw, A. J. Robinson, and M. Hochberg, "Context-dependent classes in a hybrid recurrent network-hmm speech recognition system," in *Advances in Neural Information Processing Systems*, 1996, pp. 750–756.

[21] D. K. T. R. M. Hochberg, "Context-dependent classes in a hybrid recurrent network-hmm speech recognition system."

[22] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth annual conference of the international speech communication association*, 2014.

[23] G. Pundak and T. Sainath, "Lower frame rate neural network acoustic models," 2016.

[24] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.