Bidirectional Temporal Convolution with Self-Attention Network for CTC-Based Acoustic Modeling

Jian Sun, Wu Guo, Bin Gu* and Yao Liu†
* University of Science and Technology of China
National Engineering Laboratory for Speech and Language Information Processing, Hefei, China
E-mail: sjian17@mail.ustc.edu.cn, guowu@ustc.edu.cn, bin2801@mail.ustc.edu.cn
† China General Technology Research Institute
E-mail: liuyao88@mail.ustc.edu.cn

***Abstract*—Connectionist temporal classification (CTC) based on recurrent (RNNs) or convolutional neural networks (CNNs) is a method for end-to-end acoustic modeling. Inspired by the recent success of the self-attention network (SAN) in machine translation and other domains such as images, we apply the SAN to CTC acoustic modeling in this paper. SAN has powerful capabilities for capturing global dependencies, but it cannot model the sequential information and local interactions of utterances. The bidirectional temporal convolution with self-attention network (BTCSAN) is proposed in order to capture both the global and local dependencies of utterances. Furthermore, the down- and upsampling strategies are adopted in the proposed BTCSAN in order to achieve computational efficiency and high recognition accuracy. Experiments are carried out using the King-ASR-117 Japanese corpus. The proposed BTCSAN can obtain a 15.87% relative improvement in the CER over the BLSTM-based CTC baseline.**

***Index Terms*—connectionist temporal classification, bidirectional temporal convolution, self-attention**

I. Introduction

There is growing interest in developing end-to-end models for large vocabulary continuous speech recognition (LVCSR). Compared with the traditional hidden Markov model (HMM)/ neural network systems [1, 2, 3, 4, 5], the end-to-end (E2E) approach avoids the need for linguistic resources such as a pronunciation dictionary or phonetic context-dependency trees, and this greatly simplifies the training and decoding process. There are three major types of end-to-end architectures for LVCSR: RNN-Transducers [6], attention-based encoder-decoder methods [7, 8, 9, 10, 11] and connectionist temporal classification (CTC)-based frameworks [12, 13, 14, 15, 16]. Generally, CTC models can achieve better performance than other E2E frameworks, and thus we adopt the CTC framework in this paper.

The CTC loss function with bidirectional long short-term memory (BLSTM) networks can achieve comparable or better results than the HMM systems on most speech recognition tasks. However, the training speed can be very slow and the training process is tricky for BLSTM modeling. To solve these issues, some researchers explored applying convolutional neural networks (CNNs) to CTC [17, 18]. Krishna et al. [19] adopted all-convolutional architectures that were trained using the CTC loss function. Although the training speed can be greatly improved, CNNs may perform poorly due to not having a sufficiently large receptive field. To broaden the receptive fields of CNNs and enhance their sequence modeling ability, researchers proposed a new kind of convolution architecture, temporal convolution networks (TCN), consisting of causal convolutions, dilated convolutions and residual connections [20]. TCN models outperform generic recurrent architectures in synthetic stress tests, polyphonic music modeling, character and word-level language modeling.

Recently, self-attention has obtained impressive performance improvements in neural machine translation [21]. It is also applied to speech recognition coupled with encoder-decoder architectures, which can obtain comparable recognition accuracy to mainstream systems [22, 23, 24]. Self-attention can extract global information, but it lacks the ability to model the local contextual information of sequence signals such as speech. Ref. [25] proposed QANet in order to combine self-attention and convolution, in which self-attention models the global interactions and convolution models the local interactions. QANet has showed its superiority in the fields of machine reading and question answering.

Inspired by the work mentioned above, we propose the bidirectional temporal convolution with self-attention network (BTCSAN) and apply it to the CTC framework. Compared with QANet encoders, we replace the CNN modules with bidirectional temporal convolution networks (BTCNs) to capture local contextual information of speech. Compared with the TCN, the BTCN can capture both the future and the past contextual information in speech. In addition, to prevent the GPU memory from overflowing and to maintain the effectiveness of CTC training, we design down- and upsample modules for our acoustic model.

In this paper, we apply BTCSAN to Japanese speech recog-

This work was partially funded by the National Key Research and Development Program of China (Grant No. 2016YFB100 1303) and the National Natural Science Foundation of China (Grant No. U1836219).

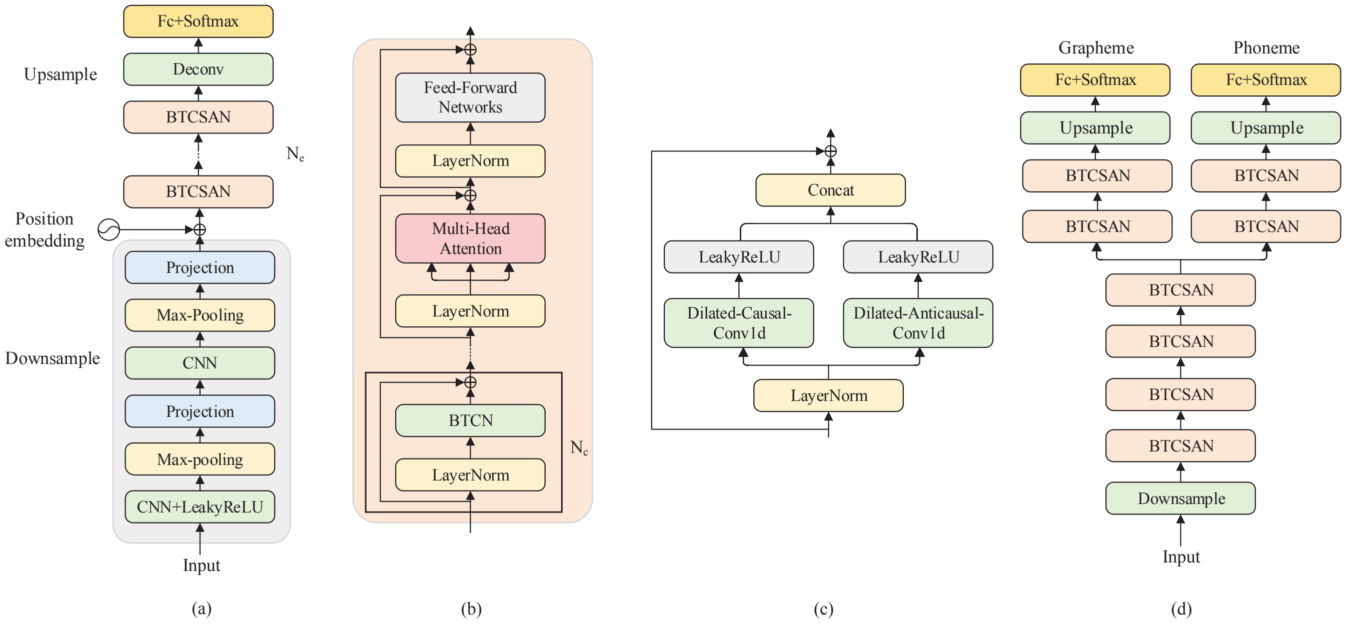

Fig. 1. (a) The overview of the architecture for CTC training, which consists of a downsample module, an upsample module, a position encoding layer, an output layer and several BTCSAN modules. (b) For one BTCSAN module in our proposed model, we use a multihead self-attention layer, a feed-forward layer and multiple bidirectional temporal convolution network layers. (c) For a BTCN layer, 1-D CNN structures are used with causal convolutions, anticausal convolutions and dilated convolutions. (d) The MTL framework for Japanese speech recognition is shown here.

nition. Japanese has over 2000 graphemes, including Kanji, Hiragana and Katakana, which are a natural fit for selection as the modeling units in CTC systems. Considering that a single Kanji may have different pronunciations (phonemes), we integrate the phoneme information into the CTC framework using a multitask learning (MTL) strategy in which the primary task adopts the graphemes as outputs, and the auxiliary task adopts phonemes as outputs. The experimental results demonstrate that the MTL framework is effective.

The rest of this paper is organized as follows. We introduce our proposed architecture for CTC training, especially for BTCSAN, in Section II. Section III gives a brief description of the multitask training framework. Section IV shows our experimental setup and other details, including the experimental results. Finally, the conclusion is presented in Section V.

II. The Module

The overall architecture of the proposed framework is depicted in Fig. 1(a), which mainly contains five components: a downsample module, an upsample module, a position encoding layer, an output layer and a stack of N_e identical BTCSAN modules. We will describe the details of these components in the rest of this section.

A. BTCSAN

The stacked BTCSAN modules are the core components of the proposed CTC training architecture. Each BTCSAN module consists of a multihead self-attention layer, a position-wise feed-forward layer, and N_c BTCN layers, as depicted in Fig. 1(b). We employ a residual connection around each of the two sublayers, which is followed by layer normalization.

1) Multihead Self-Attention: Our self-attention model closely follows the SAN model that is presented in [21]. To obtain information from the different representation subspaces, we adopt multihead attention as in [21, 22]. Multihead attention first linearly projects the queries, keys and values h times with different, learned projections. Then, these outputs are concatenated and projected again in order to obtain the final results.

$$\text{Multihead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Concat}(\text{head}_1,...,\text{head}_\text{h})\mathbf{W}^o \quad (1)$$

$$\text{where head}_\text{i} = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \quad (2)$$

where $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ stand for queries, keys and values, the parameters $\mathbf{W}_i^Q \in \mathbb{R}^{d_{model} \text{ x } d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{model} \text{ x } d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_{model} \text{ x } d_v}$ and $\mathbf{W}_i^O \in \mathbb{R}^{d_{model} \text{ x } d_{model}}$ are projection matrices, and $d_k = d_v = d_{model}/h$.

2) Position-wise Feed-Forward Network: The position-wise feed-forward network (FFN) consists of two linear transformations with an ReLU activation in between, which will introduce additional depth and nonlinearities.

$$\text{FFN}(\mathbf{x}) = \max(0, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2} \quad (3)$$

where $\mathbf{W_1} \in \mathbb{R}^{d_{model} \text{ x } d_{ff}}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_{ff} \text{ x } d_{model}}$, $\mathbf{b_1}$ and $\mathbf{b_2}$ are bias. And in this paper, we set $d_{ff} = 2d_{model}$.

3) Bidirectional Temporal Convolution Network: The proposed BTCN is depicted in Fig. 1(c). The TCN has a wider receptive field than the conventional CNN, and this is fit for sequencing signals such as speech. Since the SAN lacks information about the relative or absolute positions of frames in the speech sequence, we use the BTCN with causal and anticausal convolutions in order to capture the position information. A causal convolution produces an output at time t, which is convolved only with the elements from time t and earlier in the previous layer. Conversely, an anticausal convolution uses elements from time t and after that in the previous layer.

In our model, the BTCN only adopts the 1D-CNN so that the network can produce an output that is the same length as the input. To make the network more memory efficient, we use depthwise separable convolutions.

To obtain more historic or future information from the sequence, we employ dilated convolutions in the network. For a 1-D sequence $\mathbf{x}$ and a filter $f : \{0, ..., k-1\}$, the dilated convolution operation at time-step t of the sequence is defined as:

$$\text{Dilated\_causal}(\mathbf{x}, d, k) = \sum_{j=0}^{k-1} f(j) \cdot \mathbf{x}_{t-d \cdot j} \tag{4}$$

$$\text{Dilated\_anticausal}(\mathbf{x}, d, k) = \sum_{j=0}^{k-1} f(j) \cdot \mathbf{x}_{t+d \cdot j} \tag{5}$$

where d is the dilation factor and k is the filter size.

For the $i_{th}(1 \leq i \leq N_c)$ BTCN in one BTCSAN module, the output is computed as follows:

$$\mathbf{x\_fw} = \text{Dilated\_causal}(\text{layernorm}(\mathbf{x}), d_i, k) \tag{6}$$

$$\mathbf{x\_bw} = \text{Dilated\_anticausal}(\text{layernorm}(\mathbf{x}), d_i, k) \tag{7}$$

$$\mathbf{O} = \text{concat}(\mathbf{x\_fw}, \mathbf{x\_bw}) + \mathbf{x} \tag{8}$$

where d_i is the dilation factor, $d_i = 2^{i-1}$, $\mathbf{x} \in \mathbb{R}^{d_{model} \text{ x } T}$, T is length of the acoustic sequence and $\mathbf{x\_fw}, \mathbf{x\_bw} \in \mathbb{R}^{d_{model}/2 \text{ x } T}$. After each dilated CNN layer, we adopt the LeakyReLU as the nonlinear function.

B. Downsample and Upsample

During the training process, the GPU memory may overflow because self-attention requires large memory in order to store the attention scores of every two frames of the acoustic sequence, and the number grows quadratically with the sequence length. Taking this issue into account, we design a downsample module in order to reduce the length of speech and make our model more memory-efficient.

We stack two blocks in the downsample module. For each block, there are two 2-D CNN layers and a max-pooling layer between CNNs. The max-pooling operation is executed among the time dimensions. A reshaping operation is applied after the second max-pooling layer, which is followed by a projected operation on the flattened feature map outputs in order to obtain the vectors of dimension d_{model}.

After two max-pooling operations in the downsample module, the size of the speech sequences becomes a quarter of the original size. On the other hand, the deconvolution operation before the output layer must keep the same size as the original speech sequences in order to achieve high recognition accuracy. We use a 1x4 deconvolution in order to fulfill this upsampling operation.

水

On-yomi: すい(sui)

Kun-yomi: みず(mizu)

Fig. 2. Different pronunciations of the kanji character '水'

C. Position encoding

In order to further enhance the sequence modeling ability of our acoustic model, we add the position encoding to the input encoding as in [21, 22].

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \tag{9}$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \tag{10}$$

where pos is the position and i is the dimension.

III. Multitask Learning Architecture

We conducted experiments on the Japanese ASR task. Japanese has a complex writing system, including Kanji and two syllabaries: hiragana and katakana. These graphemes can be used as model units in CTC-based systems. Hiragana and katakana are simplified from kanji and have a consistent one-to-one match pronunciation in Japanese, but most kanji characters have two or more pronunciations. A simple example is illustrated in Fig. 2, where the kanji character '水' can have two pronunciations as 'すい' and 'みず'.

Considering the polyphone problem, we combine the phoneme information into the grapheme-based CTC system in order to train the acoustic model. As depicted in Fig. 1(d), the architecture takes the grapheme-based CTC system as the primary task and the phoneme-based CTC system as an auxiliary task by sharing the hidden layers of the same network.

In the training procedure, the objective is presented as (11), and it includes an adjustable parameter α.

$$O_{all} = (1-\alpha)O_{grapheme} + \alpha O_{phoneme} \tag{11}$$

where $0 \leq \alpha \leq 1$. $O_{grapheme}$ and $O_{phoneme}$ present the objectives of the two tasks. α determines the influence of the secondary task on the model. We only use grapheme-branch outputs for decoding.

IV. Experiments

A. Database

The experiments are conducted using the King-ASR-117 corpus. This corpus is a Japanese speech database that was collected by the Speechocean Corporation[1] . All the speech files are sampled at 16K Hz with 16 bits. The transcripts contain 122,847 (approx.) utterances in total with approximately 145 h of speech. In the experiments, we randomly selected 123 h, 6 h and 4 h of speech data as the training, dev and test sets, respectively.

[1] www.speechocean.com

B. Experiment setup

The Pytorch and Eesen [26] toolkits are used in our model training process. The acoustic feature is 108-dimensional filter-bank of features (36 filter-bank features, delta coefficients, and delta-delta coefficients) with mean and variance normalization. According to the statistical information of the transcripts, there are 2794 different graphemes (Kanji, hiragana and katakana) in the training set. Along with the added blank, 2795 modeling units are used in the grapheme-based CTC system. The trigram language model is used in the decoding procedure.

C. Baselines

We build three types of CTC systems as our baselines. All the networks are optimized by Adam [27], and the initial learning rate is set to 0.0004, which is halved if the performance when using the cross-validated data (dev set) degrades.

- The first system is a classical BLSTM-CTC system, which contains 3 layers with 1024 nodes in each layer.
- The second system is a CNN-CTC system, which was proposed in [19]. In this system, traditional 1-D CNN structures are adopted. The input acoustic sequences are convolved and followed by max-pooling across time in the first layer. After the subsequent ResBlocks (composed of a pair of convolution layers and a residual connection), the last two layers before the output nodes are fully connected layers with 512 hidden units. The batch normalization and a nonlinear ReLU are also added after every convolution layer, which has output channels. The kernel size of the 1-D CNN is set to 5, which is the same as the setting used in [19].
- The last is a SAN-CTC system, which explores the strength of self-attention to acoustic modeling. In the experiments, SAN-CTC adopts the same down- and upsample modules as our proposed framework, and the details are as shown in Table I. Meanwhile, we set both the attention dropout and residual dropout to 0.1 in the SAN system [22]. For the LeakyRelu function in our model, the negative slope is set to 0.1.

The character error rate (CER) is used as an evaluation criterion for our systems. The performance of the three baseline systems are listed in Table II. Though the CNN system has much fewer parameters than the BLSTM system, the performance is not satisfactory. The SAN system can outperform BLSTM and be more memory-efficient, which proves the SAN's ability of capturing global information . When we set $head = 8$ and $N_e = 6$, the best performance with $\mathrm{CER} = 10.98\%$ can be obtained, and this result is used as our baseline performance.

TABLE I
THE CONFIGURATION OF THE DOWN- AND UPSAMPLE MODULES IN SAN-CTC SYSTEMS. PARAM(A,B,C,D) REPRESENTS THE INPUT CHANNELS, OUTPUT CHANNELS, AND THE KERNEL SIZE AMONG THE DIMENSION AND TIME AXES, RESPECTIVELY, AND STRIDE(E,F) DENOTES THE STRIDE AMONG THE DIMENSION AND TIME AXES, RESPECTIVELY.

Module	Layer	Param	Stride
Down-sample	Conv	(1,64,9,3)	(1,1)
	Max-Pooling	(-,-,1,2)	(1,2)
	Projection	(64,64,1,1)	(1,1)
	Conv	(64,64,3,3)	(1,1)
	Max-Pooling	(-,-,1,2)	(1,2)
	Reshape	-	-
	Projection	(108*64,d_{model},1,1)	(1,1)
Upsample	Deconv	(1,1,1,4)	(1,4)

TABLE II
COMPARISONS OF THE BASELINES (BLSTM, CNN AND SAN) WITH CER% USING THE KING-ASR-117 CORPUS.

Module	N_e	$head$	d_{model}	#Weights	CER%
BLSTM	3	-	1024	65.3M	11.28
CNN	14	-	256	10.3M	12.85
	17	-	256	12.3M	12.98
	28	-	256	19.5M	12.90
SAN	5	4	256	5.16M	13.31
	5	8	512	15.5M	11.20
	6	8	512	17.6M	**10.98**
	7	8	512	19.7M	11.49

D. BTCSAN-CTC

For the proposed BTCSAN-CTC system, the settings are the same as the baseline SAN-CTC system in Table I, and we also set $head = 8$ and $N_e = 6$ for the BTCSAN system in order to compare the CERs. The performances of the BTCSAN-CTC system are listed in Table III. It can be observed that the best performance with a CER 9.88% is obtained when $N_c = 2$ and $k = 3$. Compared with the SAN baseline, a 1.1% absolute CER reduction can be obtained. However, we continue increase N_c, and the performance degrades because of the network depth. In order to further investigate the influence of the BTCN layer, we establish two systems that only use causal convolutions or anticausal convolutions instead of both of them, which are denoted as TCSAN and r-TCSAN, respectively. The last two rows of Table III list the results of the TCSAN and r-TCSAN, and their performances are much worse than the BTCSAN system. The reason may be that BTCSAN can learn the contextual information more effectively through the causal convolutions and anticausal convolutions.

E. Multitask learning framework

To further improve the performance, the MTL training framework is adopted, and the mono-phones are used as the modeling units of the auxiliary task. According to the lexicon of Speech-ocean Corporation, there are 864 phoneme variants because of word-position dependency and 1 blank unit used in the experiments. In the MTL framework, the two branches share the first four encoder layers, and the other settings are same as the BTCSAN-CTC system in Section 4.4. And the parameter α in the MTL is adjusted using the dev set. As presented in Table IV, when α is 0.4, the system obtains the best performance with a CER of 9.49% for test set. Compared with the best results in Table III, a 0.39% reduction can be obtained, which means that the phoneme information is helpful

TABLE III
THE CER% OF BTCSANS WITH VARIATIONS IN THE ARCHITECTURE.

Module	N_C	K	#Weights	CER%
SAN	-	-	17.6M	10.98
BTCSAN	1	3	19.2M	10.44
	2	3	20.8M	**9.88**
	3	3	22.5M	10.10
	4	3	24.1M	10.44
	2	5	20.9M	9.97
	2	7	20.9M	10.20
TCSAN	2	3	20.8M	10.84
r-TCSAN	2	3	20.8M	10.62

TABLE IV
THE CER% OF MTL FRAMEWORKS WITH DIFFERENT α

Module	α	dev set	test set
MTL-BTCSAN	0.1	3.95	9.54
	0.2	3.96	9.64
	0.3	3.92	9.58
	0.4	3.91	**9.49**
	0.5	4.06	9.66

for CTC acoustic modeling.

V. CONCLUSIONS

In this paper, we propose the bidirectional temporal convolution with self-attention network and explore its capabilities for CTC-based acoustic modeling. BTCSAN takes full advantage of the long-range dependencies and local information on acoustic sequences. In addition, causal and anticausal convolution operations enhance the ability of the network for sequence modeling. Furthermore, we apply it to the multitask learning framework using a Japanese corpus, and the proposed network gives a 15.87% relative reduction in the CER over the BLSTM baseline system.

REFERENCES

[1] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Large vocabulary continuous speech recognition with context-dependent DBN-HMMS,” *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP),* IEEE, 2011, pp. 4688-4691.

[2] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly et al., “Deep neural networks for acoustic modeling in speech recognition,” *IEEE Signal processing magazine,* vol. 29, 2012.

[3] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” *2013 IEEE international conference on acoustics, speech and signal processing(ICASSP),* IEEE, 2013, pp. 6645-6649.

[4] A. Graves, N. Jaitly, and A.-r. Mohamed, “Hybrid speech recognition with deep bidirectional LSTM,” *2013 IEEE workshop on automatic speech recognition and understanding,* IEEE, 2013, pp. 273-278.

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

[6] A. Graves, “Sequence transduction with recurrent neural networks,” *International Conference of Machine Learning (ICML) 2012 Workshop on Representation Learning,* 2012.

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

[8] R. Prabhavalkar, T. N. Sainath, B. Li, K. Rao, and N. Jaitly, “An Analysis of ”Attention” in Sequence-to-Sequence Models,” *Interspeech,* 2017, pp. 3702-3706.

[9] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N.Jaitly, “A Comparison of Sequence-to-Sequence Models for Speech Recognition,” *Interspeech,* 2017, pp. 939-943.

[10] C. Weng, J. Cui, G. Wang, J. Wang, C. Yu, D. Su et al., “Improving attention based sequence-tosequence models for end-to-end english conversational speech recognition,” *Interspeech,* 2018, pp. 761-765.

[11] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen et al., “State-of-the-art speech recognition with sequence-to-sequence models,” *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE, 2018, pp. 4774-4778.

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

[13] A. Maas, Z. Xie, D. Jurafsky, and A. Ng, “Lexicon-free conversational speech recognition with neural networks,” *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* 2015, pp. 345-354.

[14] T. Zenkel, R. Sanabria, F. Metze, and A. Waibel, “Subword and Crossword Units for CTC Acoustic Models,” *Interspeech,*2018, pp. 396-400.

[15] S. Li, X. Lu, R. Takashima, P. Shen, T. Kawahara, and H. Kawai, ‘Improving CTC-based Acoustic Model with Very Deep Residual Time-delay Neural Networks,” *Interspeech,* 2018, pp. 3708-3712.

[16] J. Li, G. Ye, A. Das, R. Zhao, and Y. Gong, “Advancing acoustic-to-word CTC model,” *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* IEEE, 2018, pp. 5794-5798.

[17] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. Laurent, Y. Bengio et al., “Towards End-to-End Speech Recognition with Deep Convolutional Neural Networks,” *Interspeech,* 2016, pp. 410-414.

[18] Y. Wang, X. Deng, S. Pu, and Z. Huang, “Residual convolutional CTC networks for automatic speech recognition,” *arXiv preprint arXiv:1702.07793,* 2017.

[19] K. Krishna, L. Lu, K. Gimpel, and K. Livescu, “A Study of All-Convolutional Encoders for Connectionist Temporal Classification,” *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* IEEE, 2018, pp. 5814-5818.

[20] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv preprint arXiv:1803.01271,* 2018.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N.Gomez et al., “Attention is all you need,” *Advances in Neural Information Processing Systems,* 2017, pp. 5998-6008.

[22] L. Dong, S. Xu, and B. Xu, “Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition,” *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* IEEE, 2018, pp. 5884-5888.

[23] S. Zhou, L. Dong, S. Xu, and B. Xu, “Syllable-Based Sequence-to-Sequence Speech Recognition with the Transformer in Mandarin Chinese,” *Interspeech,* 2018, pp. 791-795.

[24] M. Sperber, J. Niehues, G. Neubig, S. Stuker, and A. Waibel, “Self-Attentional Acoustic Models,” *Interspeech,* 2018, pp. 3723-3727.

[25] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi et al., “Qanet: Combining local convolution with global self-attention for reading comprehension,” *International Conference on Learning Representations (ICLR),* 2018.

[26] Y. Miao, M. Gowayyed, and F. Metze, “EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding,” *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)* IEEE, 2015, pp. 167-174.

[27] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980* 2014.