INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Efficient Conformer with Prob-Sparse Attention Mechanism for End-to-End Speech Recognition

Xiong Wang[1*], *Sining Sun*[2*], *Lei Xie*[1†], *Long Ma*[2]

[1]Audio, Speech and Language Processing Group,
School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2]Tencent, Beijing, China

xwang@npu-aslp.org, siningsun@tencent.com, lxie@nwpu.edu.cn, malonema@tencent.com

Abstract

End-to-end models are favored in automatic speech recognition (ASR) because of their simplified system structure and superior performance. Among these models, Transformer and Conformer have achieved state-of-the-art recognition accuracy in which self-attention plays a vital role in capturing important global information. However, the time and memory complexity of self-attention increases squarely with the length of the sentence. In this paper, a prob-sparse self-attention mechanism is introduced into Conformer to sparse the computing process of self-attention in order to accelerate inference speed and reduce space consumption. Specifically, we adopt a Kullback-Leibler divergence based sparsity measurement for each query to decide whether we compute the attention function on this query. By using the prob-sparse attention mechanism, we achieve impressively 8% to 45% inference speed-up and 15% to 45% memory usage reduction of the self-attention module of Conformer Transducer while maintaining the same level of error rate.

Index Terms: speech recognition, prob-sparse attention mechanism

1. Introduction

In recent years, end-to-end (E2E) automatic speech recognition (ASR) technology has made great progress with its simplified architecture and competitive performance. Transducer [1, 2, 3, 4, 5, 6] and attention based encoder-decoder (AED) [7, 8, 9] are two popular E2E frameworks. For both kinds of models, the encoder is crucial for good performance. The Transformer architecture with self-attention has recently become the core block of most E2E models. Compared with RNN with long-short term memory (LSTM) units, Transformer block gives better accuracy and efficient computation because of its ability of modeling long-range global context. However, vanilla self-attention based Transformer block is less capable of capturing local information, which is also essential to speech recognition. To take advantage of both local and global contexts, Conformer [10] block is proposed, which combines convolution with self-attention, where the convolution is used to capture the local information while the self-attention for the global information consideration. Much work has shown that Conformer can obtain better speech recognition accuracy with fewer model parameters.

For both Transformer and Conformer blocks, self-attention plays a very important role. With the self-attention mechanism, each output is the weighted combination of the whole sequence, which makes it have the ability to model the global information. However, there are two problems to apply global self-attention to the ASR task. The first one is the computation efficiency. The time complexity of self-attention can reach $O(T^2)$ concerning the sequence length T. With the increase of the input sequence length, the time complexity will increase in a quadratic way. For the ASR task, the speech signal can last from seconds to minutes and the input length varies from dozens to thousands in terms of frames. For example, in the popular benchmark dataset LibriSpeech [11], the audio clips last from about 1.5 to 20 seconds. Therefore, reducing computational complexity is essential, especially for decoding long utterances. The second problem is that the self-attention mechanism does not consider the information redundancy issue. It is clear that considering the whole input can benefit the current output. However, the speech signal is highly structured and there can be redundant frames that are useless for a specified input.

Many researchers have recently attempted to optimize the above mentioned computation and memory consumption of self-attention mechanism. For example, Longformer [12] was proposed to handle the long document processing tasks, which combines local windowed attention with a task motivated global attention to process documents of thousands of tokens or longer. As for the ASR task, Shi et al. recently proposed Emformer [13], which converts a long-range history context into an augmentation memory bank to reduce the computation complexity of self-attention. Although these previous attempts can reduce the computation complexity, they have not considered the second issue $-$ computation redundancy.

In this paper, inspired by the Informer [14] originally proposed for long sequence time-series forecasting, we propose an efficient Conformer with prob-sparse attention mechanism to tackle the above mentioned two problems. Specifically, the attention score between i-th query and all keys can be defined as a probability. If attention score of the i-th query over all keys is close to a uniform distribution, the self-attention degenerates into a trivial sum of all values, which is apparently redundant. Therefore, only the queries with attention score distribution far from uniform distribution play dominant roles. Under this probability distribution measurement, prob-sparse self-attention only allows each key to attend to the top-u queries. Compared with the vanilla self-attention mechanism, the query matrix is sparse because it only contains top-u dominant queries. The prob-sparse method will reduce the computation complexity into $O(uT)$ without performance degradation because this process retains effective queries and removes redundant computation. Furthermore, to reduce the extra computation of the sparse measurement, we also exploit "sparsity measurement sharing" strategies between layers. By using the prob-sparse attention mechanism with sparsity measurement sharing, we achieve im-

[*]equal contribution
[†]corresponding author

http://dx.doi.org/10.21437/Interspeech.2021-415

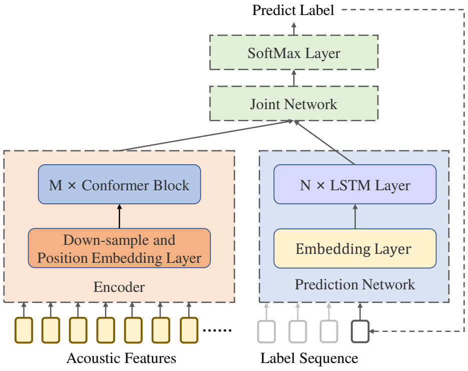

Figure 1: *Architecture of Conformer Transducer.*

pressively 10% to 40% inference speed-up of the Conformer Transducer while maintaining the same level of CER/WER on AISHELL-1 [15] and LibriSpeech [11] corpora.

2. Conformer Transducer

2.1. Architecture of Transducer

In this paper, we use Conformer Transducer to verify our proposed method. The Transducer model can establish the alignment between acoustic features $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T]$ and label sequences $\mathbf{y} = [y_1, y_2, \cdots, y_U]$, where T is the number of feature frames and U is the length of the label sequence. Figure 1 shows the transducer structure used in this paper, which mainly includes an encoder, a prediction network and a joint network. Specifically, the encoder converts the acoustic features into a high-dimensional representation $\mathbf{h}^{enc}$ which can be expressed as

$$\mathbf{h}^{enc} = \text{Encoder}(\mathbf{x}). \tag{1}$$

The main purpose of prediction network is to generate a higher-dimensional representation h_u^{pred} of last label y_{u-1} as shown in Eq. (2). To avoid the sparsity caused by directly using one-hot label as input, an embedding layer is arranged before N layers of LSTM.

$$h_u^{pred} = \text{Prediction}(y_{u-1}) \tag{2}$$

The joint network is represented by several fully connected layers, and finally a softmax layer is used to predict the probability $P(k|t,u)$ of the next label, as shown in

$$h_{t,u} = \mathbf{W}^{joint}\tanh(\mathbf{U}h_t^{enc} + \mathbf{V}h_u^{pred} + b) + b_{joint} \tag{3}$$

$$P(k|t,u) = \text{softmax}(h_{t,u}) \tag{4}$$

where $\mathbf{U}$ and $\mathbf{V}$ are the projection matrices to combine h_t^{enc} and h_u^{pred}, and $\mathbf{W}^{joint}$ is used to project the output dimension to the number of labels. During training, transducer uses the forward-backward algorithm [16] to maximize the posterior of $\mathbf{y}$ given $\mathbf{x}$.

2.2. Conformer Encoder

Recently, Conformer has achieved excellent performance in speech recognition [10]. Compared with Transformer, Conformer adds an additional convolution module to make up for the lack of local feature extraction ability of single self attention. A standard Conformer block consists of four modules: feed-forward module, self-attention module, convolution module and another feed-forward module. The Conformer block used in this paper does not use relative positional embedding in the self-attention block. The specific composition of each module is described in [10].

As shown in Figure 1, when using Conformer as the encoder of Transducer, the input feature $\mathbf{x}$ first passes through the down-sampling layer and position embedding layer and then passes through M Conformer blocks. Each Conformer block needs to go through a macaron-like feed-forward module (FFN), multi-head self attention module (MHSA), convolution module (Conv), and another FFN with layernorm in order to get a higher-dimensional representation. This process can be described in Eq. (5).

$$\begin{aligned} x_{FFN_1} &= x + \frac{1}{2}\text{FFN}(x) \\ x_{MHSA} &= x_{FFN_1} + \text{MHSA}\left(x + \frac{1}{2}\text{FFN}(x)\right) \\ x_{Conv} &= x_{MHSA} + \text{Conv}(x_{MHSA}) \\ x_{FFN_2} &= \text{ LayerNorm }\left(x_{Conv} + \frac{1}{2}\text{FFN}(x_{Conv})\right) \end{aligned} \tag{5}$$

And for the prediction network, it still has N-layer LSTM commonly used by Transducer, while joint network is also implemented by the standard fully-connection layer.

3. Prob-Sparse Self-Attention

3.1. Self-attention Mechanism and Sparsity Analysis

Given input matrix $\mathbf{X} \in \mathbb{R}^{L\times d_x}$, where L and d_x are sequence length and dimension of input respectively. The self-attention mechanism first projects $\mathbf{X}$ into query matrix $\mathbf{Q} = \mathbf{X}\mathbf{W}_q$, key matrix $\mathbf{K} = \mathbf{X}\mathbf{W}_k$ and value matrix $\mathbf{V} = \mathbf{X}\mathbf{W}_v$, where $\{\mathbf{Q}, \mathbf{K}, \mathbf{V}\} \in \mathbb{R}^{L\times d}$ and $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v$ are projection matrices with appropriate shapes. Then attention in scaled dot-product form can be represented by

$$\mathcal{A}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}. \tag{6}$$

In order to describe prob-sparse self-attention, Eq. (6) will be reformulated as its vector form. Specifically, for the i-th query $\mathbf{q}_i$ from $\mathbf{Q}$, the attention score over the j-th key from $\mathbf{K}$ can be expressed as

$$p(\mathbf{k}_j|\mathbf{q}_i) = \frac{\mathrm{e}^{\mathbf{q}_i\mathbf{k}_j^\top}/\sqrt{d}}{\sum_{l=1}^{L}\mathrm{e}^{\mathbf{q}_i\mathbf{k}_l^\top}/\sqrt{d}}. \tag{7}$$

Here we define $p(\mathbf{k}_j|\mathbf{q}_i)$ as attention score of query $\mathbf{q}_i$ over key $\mathbf{k}_j$. Then the self-attention of $\mathbf{q}_i$ over $\mathbf{K}$ can be expressed as

$$\mathcal{SA}(\mathbf{q}_i,\mathbf{K},\mathbf{V}) = \sum_{j=1}^{L} p(\mathbf{k}_j|\mathbf{q}_i)\mathbf{v}_j. \tag{8}$$

In this process, the time complexity of attention mechanism is $O(L^2)$. In fact, there is a certain sparsity in the query matrix [14], which means that it is unnecessary to calculate attention for all queries. Therefore, vanilla self-attention will have a lot of redundant calculations. Generally speaking, if the distribution of $p(\mathbf{k}_j|\mathbf{q}_i)$ yields to uniform a distribution, that is $p(\mathbf{k}_j|\mathbf{q}_i) = 1/L$, the self-attention of Eq. (8) shrinks into a trivial average of all values. Therefore, only the queries whose attention scores over all keys are far from uniform distribution are important. From this aspect, the Kullback-Leibler(K-L) divergence between the real distribution P of $p(\mathbf{k}_j|\mathbf{q}_i)$ and the uniform distribution U can be used to measure the validity of each query, expressed by

$$KL(P||U) = \ln\sum_{j=1}^{L} e^{\frac{\mathbf{q}_i\mathbf{k}_j^\top}{\sqrt{d}}} - \frac{1}{L}\sum_{j=1}^{L}\frac{\mathbf{q}_i\mathbf{k}_j^\top}{\sqrt{d}} - lnL. \tag{9}$$

After removing the constant, we can get Eq. (10) and define it as sparsity measurement of $\mathbf{q}_i$. The queries with larger M_{sparse} values play more important roles in the self-attention mechanism. In this way, we can measure the overall sparsity of each query and ignore the queries with lower M_{sparse} values during self-attention calculation, which can speed up the whole calculation procedure.

$$M_{sparse}\left(\mathbf{q}_i, \mathbf{K}\right) = \ln \sum_{j=1}^{L_K} e^{\frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d}}} - \frac{1}{L} \sum_{j=1}^{L_K} \frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d}} \tag{10}$$

3.2. Prob-Sparse based Self-Attention

Although the sparsity measurement proposed in Eq. (10) works well in theory, it needs to traverse all queries to calculate the dot-product pairs, which is still computing intensive. According to [14], to further reduce the computation of this part, Eq. (10) can be approximated by Eq. (11) using sampling methods:

$$\bar{M}_{sparse}\left(\mathbf{q}_i, \tilde{\mathbf{K}}\right) = \max_j \left\{ \frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d}} \right\} - \frac{1}{\tilde{L}} \sum_{j=1}^{\tilde{L}} \frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d}} \tag{11}$$

where $\tilde{\mathbf{K}}$ is a random sampling key matrix and $\tilde{L}$ is sampling number defined as $\tilde{L} = r_{sample} \ln L$. Here, sampling rate r_{sample} is constantly used to control how many samples can be sampled. After obtaining $\bar{M}_{sparse}$ for each query, only top-L_{sparse} queries with high $\bar{M}_{sparse}$ value will be used to calculate self-attention, where $L_{sparse} = r_{sparse}L$ and $0 < r_{sparse} < 1$ where r_{sparse} is sparse rate. Finally, the self-attention will be calculated using

$$\mathcal{SA}(\mathbf{q}_i, \mathbf{K}, \mathbf{V}) = \begin{cases} \sum_{j=1}^{L} p(\mathbf{k}_j|\mathbf{q}_i)\mathbf{v}_j, & \text{if } i \in I_{sparse} \\ \mathbf{v}_i, & \text{else} \end{cases} \tag{12}$$

where I_{sparse} is the index of top-L_{sparse} queries. It is obvious that for multi-layer attention, a query whose index is not in I_{sparse} will directly output the value and attention calculation is no longer needed.

4. Experiments

4.1. Datasets

In this paper, we verify our proposed prob-sparse attention mechanism based Conformer Transducer on two open source datasets: AISHELL-1 and LibriSpeech. The AISHELL-1 corpus contains 178 hours of labeled Mandarin speech data, recorded on different devices, and internally divided into the training, development and test sets [1]. LibriSpeech contains 970 hours of labeled English speech data, internally divided into training, clean and other development and test sets [2].

4.2. Experimental Setup

For all experiments, we use 80-dimensional log mel-filterbanks plus the 3-dimensional pitch features extracted by the KALDI toolkit [17] as the acoustic features . We also apply SpecAugment [18] for acoustic data augmentation. A set of 4230 Chinese characters is used as the modeling units for AISHELL-1 and a set of 5000 word pieces is used for LibriSpeech. All models are trained using the ESPNet toolkit [19]. During training, we use the Noam optimizer with peak learning rate at 1e-3 and gradient clipping is set at 5.0 for all models. All the models for evaluation are obtained by averaging the best five models according to the loss on the development set. In different hyper-parametric experiments, the random seed and training steps are the same. All the experimental results are conducted on 4 NVIDIA Tesla V100 GPUs.

The Conformer encoder consists of 2 convolutional layers to do 4-time down-sampling with position embedding layer and 16 Conformer blocks. The attention dimension of each Conformer block is 256 with 4 attention heads. The kernel size of convolution module is 3 and the dimension of feed-forward module is 1024. The prediction network has 1 LSTM layer with 256 hidden units and the joint network has 512 hidden units. The size of embedding layer is 256. For the prob-sparse self-attention, it has two additional super parameters, one is the sampling rate r_{sample} used to calculate the sparsity measurement, and the other is the sparse rate r_{sparse} used to do scaled dot-product with sparse query. The influence of these two hyper-parameters will be analyzed in the later experiment.

Table 1: *CER comparison on AISHELL-1 test sets with different parameter initialization styles and query selection ways.* $r_{sample} = 5$ *for experiment with prob-sparse self-attention.*

ExpID	Param. Initialization	Selection of $\mathbf{q}_i$	r_{sparse}	CER(%)
B0	Random	All	1.0	6.7
B1	B0	All	1.0	6.7
E0	Random	Prob-sparse	0.5	7.2
E1	B0	Prob-sparse	0.5	**6.5**
E2	B0	Random selection	0.5	8.2

4.3. Training Strategy and Performance Analysis

Table 1 shows the results of our proposed prob-sparse self-attention based Conformer Transducer on AISHELL-1 test sets. B0 is our baseline model trained with vanilla self-attention from scratch, which means we compute the attention function on all the queries. E0 is trained with our proposed prob-sparse self-attention from scratch. Compared with baseline B0, E0 with prob-sparse self-attention obtains worse CER. The main reason behind the accuracy degradation is that prob-sparse mechanism depends on a relative reliable model to calculate M_{sparse} for each query. Therefore, in experiment E1, we initialize our model from B0 and retrain the model using prob-sparse self-attention mechanism with sparse rate $r_{sparse} = 0.5$, which means that we only compute the attention function on a half of queries for each Conformer block. Compared with B0, E1 obtains even better recognition performance with CER lowered from 6.7 to 6.5, which also proves that the information redundancy issue we mentioned before does exist. To further exclude that the performance improvement comes from more training steps, experiment B1 is conducted, which proves that more training steps can not give further improvement. Furthermore, in order to show the effectiveness of our prob-sparse self-attention method, we also implement experiment E2 by sampling queries with random style. We can see that the CER increases from 6.5 (E1) to 8.2 (E2), which means that choosing queries randomly will hurt the model performance because it can not select the most useful queries.

All the experiments in Table 1 use sparse rate $r_{sparse} = 0.5$. As analyzed in Section 3, small sparse rate means we compute the attention function on fewer queries. However, a small sparse rate also means more information will be discharged. To explore how sparse rate influence the model performance, we further conducted experiments with several r_{sparse} values.

[1] www.openslr.org/33/

[2] www.openslr.org/12/

Figure 2 shows how CER varies with respect to r_{sparse}. As we expected, when $r_{sparse} < 0.35$, the CER is worse than our baseline model, which means too much information is ignored. On the contrary, when r_{sparse} is between 0.35 to 0.5, the prob-sparse attention method can obtain consistent better results than baseline because the model can discard redundant information efficiently.

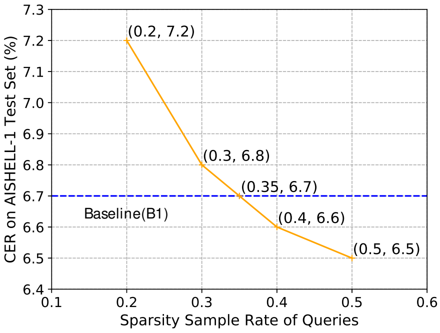

Figure 2: *Relationship between CER and r_{sparse} values on AISHELL-1 test set.*

4.4. Sparsity Measurement Share Through Layers

As we mentioned before, in order to calculate the sparsity measurement M_{sparse}, some extra computations in each layer are introduced as described in Eq. (11). It's worth noticing that the computation complexity will be further reduced if the sparsity measurement M_{sparse} can be shared through layers. To verify this hypothesis, we conduct experiments on AISHELL-1. Table 2 shows results of M_{sparse} sharing through layers. N_{share} means that M_{sparse} is calculated every N_{share} layers and it will be shared for the next $N_{share} - 1$ layers. In order to further reduce the calculation, we also try to reduce the sample number of keys in Eq. (11), which is controlled by r_{sample}. In Table 2, E1 is same as the one in Table 1, where M_{sparse} will be calculated in each layer. For S0, we first reduce the queries sample rate factor from 5 to 1, which brings a small CER increase (from 6.5 to 6.7). However, we can still obtain the same results as our baseline model B0 in Table 1 even when $r_{sample} = 1$. When we apply measurement share with $N_{share} = 4$ for this 16-layer Conformer, we can reduce 4 times of calculation with no CER increasment, which verifies that the sparsity measurement can be shared through layers. Even when we increase N_{share} to 8 in S2, there is only negligible CER increase. Therefore, sparsity measurement share is an efficient way to reduce extra calculation for our ASR model.

Table 2: *Comparison of the results on AISHELL-1 test sets in CER with different r_{sample} and N_{share} values with* $r_{sparse} = 0.5$.

Exp ID	r_{sample}	N_{share}	CER(%)
E1	5	1	6.5
S0	1	1	6.7
S1	1	4	6.7
S2	1	8	6.8

4.5. Experiments on LibriSpeech

We also carry out experiments on LibriSpeech with different hyper parameters to verify our proposed method. LibriSpeech English dataset has a larger amount of data than AISHELL-1, which can well verify the reliability of the experiments we have done above. The experimental results are shown in Table 3. Note that compared with the results reported in [10], our baseline results of M0 in Table 3 are worse because we did not spend too much effort on model refinement. Based on the results in Table 3, we can achieve consistent conclusions with those on AISHELL-1.

Table 3: *Experimental results on LibriSpeech. L0-L3 are initialized by M0.*

Exp ID	r_{sample}	r_{sparse}	N_{share}	dev clean	dev other	test clean	test other
M0	/	1	/	4.0	10.3	4.2	10.3
L0	5	0.5	4	4.0	10.5	4.3	10.5
L1	5	0.5	8	4.1	10.8	4.4	10.9
L2	1	0.5	4	4.1	10.6	4.3	10.7
L3	1	0.5	8	4.2	10.9	4.4	11.0

4.6. Efficiency and Complexity

From the above experiments, we verify that there is a certain sparsity of queries in the attention mechanism. After we use sparsity measurement to keep only the effective queries for calculation, we observe that the model performance will not degrade. At the same time, this operation can also greatly reduce the time and space complexity of self-attention. Specifically, we evaluate the time and memory consumption of the self-attention module for sentences with different length in the Conformer Transducer model during the decoding process. As shown in Figure 3, our proposed method can relatively increase the inference speed by 8% to 45% and reduce the memory usage by 15% to 45% in different sentence lengths. This advantage will be more obvious with the increase of sentence length.

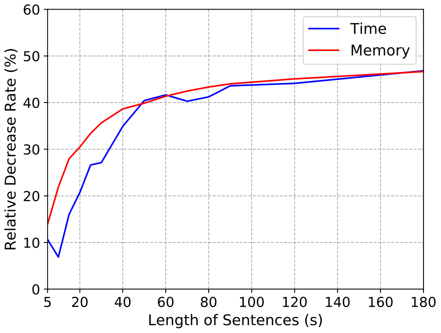

Figure 3: *Time and memory usage between standard and prob-sparse self-attention module of Conformer-Transducer on L2 in Table 3. All results are evaluated on a single thread of Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz.*

5. Conclusions

In this paper, we proposed an efficient Conformer with prob-sparse attention mechanism. Specially, K-L divergence is adopted as a sparsity measurement to evaluate the effectiveness of each query in the self-attention module of Conformer and only more effective queries contribute to the result. We achieve impressively 8% to 45% inference speed-up and 15% to 45% memory usage reduction of the self-attention module of Conformer Transducer while maintaining the same level of error rate.

6. References

[1] A. Graves, "Sequence transduction with recurrent neural networks," *Computer Science*, 2012.

[2] E. Battenberg, J. Chen, R. Child, A. Coates, Y. G. Y. Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, "Exploring neural transducers for end-to-end speech recognition," in *ASRU*, 2017, pp. 206–213.

[3] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. Chai Sim, T. Bagby, S. Chang, K. Rao, and A. Gruenstein, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP*, 2019, pp. 6381–6385.

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

[5] X. Wang, Z. Yao, X. Shi, and L. Xie, "Cascade RNN-transducer: Syllable based streaming on-device Mandarin speech recognition with a syllable-to-character converter," in *SLT*, 2021, pp. 15–21.

[6] Y. Zhang, S. Sun, and L. Ma, "Tiny transducer: A highly-efficient speech recognition model on edge devices," *ICASSP*, 2021.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[8] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016, pp. 4960–4964.

[9] L. Dong, S. Xu, and B. Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *ICASSP*, 2018, pp. 5884–5888.

[10] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *INTERSPEECH*, 2020, pp. 5036–5040.

[11] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*, 2015, pp. 5206–5210.

[12] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[13] Y. Shi, Y. Wang, C. Wu, C.-F. Yeh, J. Chan, F. Zhang, D. Le, and M. Seltzer, "Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition," *ICASSP*, 2021.

[14] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," *AAAI*, 2021.

[15] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline," in *COCOSDA*, 2017, pp. 1–5.

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

[17] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The KALDI speech recognition toolkit," in *ASRU*, 2011.

[18] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," *INTERSPEECH*, pp. 2613–2617, 2019.

[19] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPNet: End-to-end speech processing toolkit," *INTERSPEECH*, 2018.