THE SPEECHTRANSFORMER FOR LARGE-SCALE MANDARIN CHINESE SPEECH RECOGNITION

Yuanyuan zhao *Jie Li* *Xiaorui Wang* *Yan Li*

Kwai, Beijing, P.R. China

{zhaoyuanyuan, lijie03, wangxiaorui, liyan}@kuaishou.com

ABSTRACT

Attention-based sequence-to-sequence architectures have made great progress in the speech recognition task. The SpeechTransformer, a no-recurrence encoder-decoder architecture, has shown promising results on small-scale speech recognition data sets in previous works. In this paper, we focus on a large-scale Mandarin Chinese speech recognition task and propose three optimization strategies to further improve the performance and efficiency of the SpeechTransformer. Our first improvement is to use a much lower frame rate, which is shown very beneficial to not only the computation efficiency but also the model performance. The other two strategies are scheduled sampling and focal loss, which are both very effective to reduce the character error rate (CER). On a 8,000 hours task, the proposed improvements yield 10.8%-26.1% relative gain in CER on four different test sets. Compared to a strong hybrid TDNN-LSTM system, which is trained with LF-MMI criterion and decoded with a large 4-gram LM, the final optimized SpeechTransformer gives 12.2%-19.1% relative CER reduction without any explicit language models.

Index Terms— SpeechTransformer, Much Lower Frame Rate, Scheduled Sampling, Focal Loss, Large-scale Speech Recognition

1. INTRODUCTION

There has been growing interest in building an end-to-end speech recognition system, which folds all necessary components into a single neural framework, such as acoustic model, language model, pronunciation model, etc [1]. Comparing to a conventional hybrid system, such an end-to-end system typically has several advantages, including a simpler training process, allowing a joint optimization among components and a compact model size. Current end-to-end speech recognition systems can be categorized into two types: connectionist temporal classification (CTC) based [2, 3, 4, 5, 6] and attention based [7, 8, 9, 10, 11].

Attention-based sequence-to-sequence system was first introduced into speech recognition in [8]. Later on, a model namely listen, attend and spell (LAS), was examined on a large-scale speech task [10]. More recently, it shows a superior performance to a conventional hybrid system [11]. In the LAS model, recurrent neural networks (RNNs) play an essential role when generating sequential hidden representations (encoding) and emitting characters according to soft alignment at different time steps (decoding). Unfortunately, RNNs suffer from slow computation speed and may not be able to optimally exploit long-range context. These become especially severe for speech recognition task since speech sequences are commonly quite long.

Recently, a model called Transformer was proposed on machine translation (ML) task [12], which eschews recurrence and relies entirely on self-attention to compute representations of its input and output sequences. The self-attention mechanism in this model relates different positions in a single sequence to extract a higher level representation. This mechanism is quite attractive for two characteristics. The first one is its high computational efficiency. It can be efficiently implemented through batched tensor multiplication. The second one is its modeling power of context relevance. It allows direct conditioning on both short-range and long-range context, without the need to pass information step by step as is the case with RNNs. These two features are exactly needed in the attention-based end-to-end speech recognition systems, due to the long speech sequences.

The Transformer was first introduced into speech recognition in [13]. This model was called *SpeechTransformer* and was examined on the WSJ task. Then in [14, 15], the authors investigated this model on Mandarin Chinese ASR task (HKUST dataset) with different modeling units, and found the character based model performs best. However, the amount of training data in these three works is relatively small (less than 200 hours). The SpeechTransformer's performance on large-scale ASR task is still an open question.

This paper focuses on a large-scale Mandarin Chinese ASR task containing 8000 hours data, and makes three improvements to the SpeechTransformer model. The first one is much lower frame rate (mLFR). We show that reducing the frame rate from 33.3 Hz in [14, 15] to a much lower one, e.g, 16.7 Hz, is quite beneficial to the performance. It is achieved by downsampling the frames to the desired rate after features are extracted and stacked. This feature processing reduces the length of input sequence, thus improves the computational efficiency significantly. What's more, it makes both the *encoder self-attention* and *decoder-encoder attention* much easier. The second improvement is scheduled sampling (SS) [16], which feeds the previous label prediction during training rather than ground-truth. SS was applied to LAS model in [2, 11], and in this work, it's introduced into the SpeechTransformer successfully. As for the third one, we include the focal loss (FL) [17] during the training process, which is a dynamically scaled cross entropy loss and down-weights the loss assigned to well-classified examples. FL can address the class imbalance problem, which is quite serious with Chinese characters as modeling units, even with 8000 hours training data. Overall, all these three improvements bring 10.8% to 26.1% relative gain in character error rate (CER). The final SpeechTransformer model shows 12.2% to 19.1% relative CER reduction compared to a strong hybrid baseline, whose acoustic model is a TDNN-LSTM trained with lattice-free MMI (LF-MMI) objective function [18, 19]. It should be noted that there is no extra language model (LM) in the SpeechTransformer, while the hybrid baseline system is decoded with a 4-gram LM containing 41M N-grams. These experimental results show the great potential of the SpeechTransformer.

The remainder of this paper is organized as follows. Section 2

describes three improvements to the SpeechTransformer in details. The related work is introduced in Section 3. We report our experiments in Section 4 and 5 and conclude this work in Section 6.

2. IMPROVEMENTS TO SPEECHTRANSFORMER

In this section, we will first introduce the model structure of the SpeechTransformer, and then describe the three improvements to it in details.

2.1. SpeechTransformer

The architecture of the SpeechTransformer is basically belongs to the attention-based encoder-decoder structure. It stacks multi-head attention (MHA) and position-wise, fully connected layers for both the encoder and decoder. The encoder is composed of a stack of N identical layers, and each one has two sub-layers: one is a MHA, and the other is a position-wise feed-forward network. Residual connections are employed around each of the two sub-layers, followed by a layer normalization. The decoder's structure is similar to the encoder except inserting a third sub-layer to perform multi-head attention over the output of the encoder stack. To prevent leftward information flow and preserve the auto-regressive property in the decoder, all the values corresponding to illegal connections are masked out in the self-attention sub-layers of the decoder. In addition, positional encodings are added to the input of encoder and decoder, injecting some position information in the sequence.

In the SpeechTransformer, attention occurs in three different places. The first one is in *decoder-encoder attention* layers. This allows every position in the decoder to attend over all positions in the input sequence. The other two happen in *encoder self-attention* and *decoder self-attention* layers respectively, where attention relates different positions in the input or the output sequence to extract a more expressive representation. It's obvious that the computation cost of these three attentions is determined by the length of input and output sequences. Unfortunately, speech sequences are commonly quite long. Thus our first improvement to the SpeechTransformer is to use a much lower frame rate to reduce the length of input sequence.

2.2. Much Lower Frame Rate

Lower frame rate (LFR) modeling is not new. It has been applied to conventional hybrid ASR system [18, 19, 20], CTC end-to-end models [21] and attention-based end-to-end models [11, 14, 15]. The typical rate in previous LFR works is 33.3 Hz, while in this work we show that reducing this number to a much lower one, e.g, 16.7 Hz, is quite beneficial to the SpeechTransformer. It is achieved by downsampling the frames to the desired rate after features are extracted and stacked, just as Figure 1 showing, where n is the subsampling factor and the resulting frame rate is $100/n$.

We believe that this much lower frame rate (mLFR) has three advantages. The first one is it can improve the computation efficiency significantly, which is very straightforward. The second one is that it makes the *encoder self-attention* much easier. Compared with the word sequence in ML, the speech feature sequence in ASR is much longer. Moreover, the speech frames evolves rather slowly (the features are typically computed every 10ms), and there is no clear boundaries between the adjacent ones. It is therefore more difficult for the encoder self-attention to compute the similarity of each pair of frames. The mLFR processing, feature stacking and downsampling, will produce more sparse but more informative features, thus can alleviate this problem. As for the third advantage, mLFR is also conducive to the *decoder-encoder attention*, which becomes easier over the much shorter sequence of encoder output. In this work, we tried different combinations of n and m, and the results are shown in the experiments part.

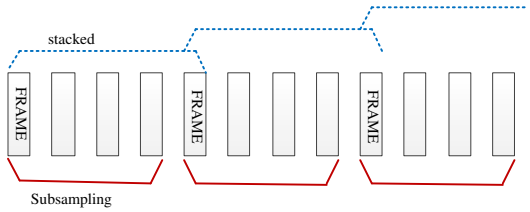

Fig. 1. mLFR processing with subsampling factor $n = 4$ and stacking number of frames $m = 5$.

2.3. Scheduled Sampling

Scheduled sampling (SS) [16] is a sampling mechanism that will randomly decide whether to use the ground-truth label or the previous prediction. This process helps to reduce the gap between training and inference behavior. SS has been applied to LAS model in [2, 11], and in this work, we introduce it into the SpeechTransformer and investigate three kinds of schedule to ramp up the sampling probability:

- Linear:

$$\varepsilon_{(i)} = \min(\varepsilon_{\max}, \varepsilon_{\max} \cdot \frac{i - N_{st}}{N_{ed} - N_{st}}) \tag{1}$$

- Exponential:

$$\varepsilon_{(i)} = \exp\left(\frac{\log(0.01)}{N_{ed} - N_{st}}\right)^{\max(N_{ed}-i, 0.0)} \tag{2}$$

- Sigmoid:

$$\varepsilon_{(i)} = 1.0 - \frac{N_{ed} - N_{st}}{\exp(^{(i-N_{st})}/_{(N_{ed}-N_{st})}) + (N_{ed} - N_{st})} \tag{3}$$

where $0 < \varepsilon_{\max} \leq 1$ is the maximum probability to sample the prediction, N_{st} is the starting step to employ scheduled sampling and N_{ed} is the step that reaches $\varepsilon_{\max}$. $i \geq N_{st}$ is the current training step and $\varepsilon_{(i)}$ represents the current sampling rate.

2.4. Focal Loss

The modeling units in this work are Chinese characters, which are very simple and intuitive. Unfortunately, the frequency distribution of characters in the training data approximately follows a Zipf's law [22]: most of the tokens in a corpus are accounted for by a small number of high frequency characters and there are many low frequency ones. As a result, the class imbalance problem is still quite serious even though our training corpus contains 8000 hours data. To alleviate it, we include focal loss (FL) [17] into the training process, which is a reshaped cross entropy (CE) that down-weights the loss assigned to well-classified examples and thus focuses training on hard ones. FL is defined as:

$$FL(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \tag{4}$$

where $\alpha_t \in [0, 1]$ represents the weighting factor, and $\gamma \in [0, 5]$ is the tunable focusing parameter. As far as we know, this is the first time that FL is introduced into the ASR task.

3. RELATED WORK

The SpeechTransformer was first proposed by [16] and then it was studied on Mandarin Chinese ASR task in [14, 15]. An important difference of this work from the previous ones is that we focus on a large-scale ASR task, which contains 8000 hours training data. In addition, we propose three improvements to further promote the performance and efficiency of the SpeechTransformer.

Self-attention has been applied to acoustic modeling in the conventional hybrid system [23], where a restricted self-attention layer was introduced into TDNN and TDNN-LSTM models. It has also been applied in the LAS model architecture [24], where the RNN encoder was partially replaced with self-attention. In these two papers, self-attention can be regarded as an improvement to the original models. While in the SpeechTransformer, self-attention plays a very important role and fully eschews recurrence.

4. EXPERIMENTAL SETUPS

4.1. Data Sets

We focus on a Mandarin ASR task, of which the training set contains 8000 hours mobile recording data. No foreign words appear in this corpus, and the code-switching ASR with the SpeechTransformer will be left for future work. The performance is evaluated on four test sets, two of them are public-available, and the remaining two come from our real traffic:

- AiShell_dev: the development set of the released corpus AiShell-1 [25], containing 14326 utterances.
- AiShell_test: the test set of the released corpus AiShell-1, containing 7176 utterances.
- LiveShow: a real traffic test set from live show, containing 5766 utterances.
- VoiceComment: a real traffic test set from voice comment, containing 5998 utterances.

4.2. Hybrid Baseline

The acoustic model in our hybrid baseline system is a TDNN-LSTM [19], trained with LF-MMI criterion computed on 33.3 Hz outputs. The structure of TDNN-LSTM follows [19], containing 7 TDNN layers and 3 LSTM layers. The dimension of each TDNN layer, and the the cell number of each LSTM layer, are both set to 1024. For each LSTM, a recurrent projection layer is added with a dimension of 512. 40-dimensional MFCCs features without cepstral truncation [26] are extracted with a 25ms window and shifted every 10ms. The language model used in the hybrid system is a 4-gram LM with 41M N-grams, trained with a large amount of text data. TDNN-LSTM has been shown to outperform a Bidirectional LSTM (BLSTM) model [19], thus our hybrid system is a quite strong baseline.

Table 1. CERs (%) of hybrid baseline system and the basic SpeechTransformer model. In the table, *ST* is short for the SpeechTransformer and *VoiceCom* is short for VoiceComment.

Models	Test Sets			
	AiShell_dev	AiShell_test	LiveShow	VoiceCom
Hybrid	4.18	4.99	31.37	8.39
ST	4.54	5.51	30.89	8.61

4.3. SpeechTransformer Baseline

All the SpeechTransformer models in this work are trained with the same features as the hybrid baseline. Global mean subtraction and variance normalization are applied for the raw features. For the SpeechTransformer baseline, the features are firstly stack with 3 frames to the left and then downsampled to 33.3 Hz frame rate, or equivalently, 30ms frame advance, same as [14, 15]. There are 6002 output units in total, which contains 5998 Chinese characters, plus 4 extra tokens, i.e., an unknown token <UNK>, a padding token <PAD>, and sentence start and end <S>/<\S>.

The SpeechTransformer contains 6 encoder and 4 decoder layers, with a configuration of $d_{model} = 512$, 16 attention heads, and 512 feed forward inner-layer dimension. During training, the samples are firstly randomly shuffled and then batched together, and each batch contains 512 sequences. We use the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\varepsilon = 10^{-9}$ and alter the learning rate over the course of training, according to the formula:

$$lrate = k \cdot d_{model}^{-0.5} \cdot \min(n^{-0.5}, n \cdot warmup\_n^{-1.5}) \quad (5)$$

where n is the step number, k is a tunable scalar, and the learning rate increases linearly for the first $warmup\_n$ training steps and decreases thereafter proportionally to the inverse square root of the step number. During training, the label smoothing of value $\varepsilon_{ls} = 0.2$ is employed [27]. After trained, the last 15 checkpoints are averaged to make the performance more stable [12]. For evaluation, we used beam search with a beam size of 3 and length penalty $\alpha = 0.6$ [28].

Table 1 gives the character error rate (CER) of the hybrid and the SpeechTransformer baselines. We can see that the performance of basic SpeechTransformer model is worse than the hybrid baseline system.

5. EXPERIMENTAL RESULTS

In this section, we explore the three improvements discussed in Section 2 with different configurations. The results and analysis will be depicted in detail.

5.1. Much Lower Frame Rate

Our first set of experiments focus on various settings of different lower frame rates, which are obtained by first stacking with several frames then downsampling to the desired frame rate. We guarantee that there is one to two overlapping frames between two adjacent generated inputs. The results are summarized in Table 2.

Table 2. CERs (%) of different lower frame rate settings.

Stack_LFR	Test Sets			
	AiShell_dev	AiShell_test	LiveShow	VoiceCom
Left3_33.3	4.54	5.51	30.89	8.61
Left5_25.0	4.52	5.49	30.01	8.75
Left7_16.7	**4.41**	**5.22**	**29.7**	**8.13**
Left9_12.5	4.58	5.69	31.81	9.68
CERR	2.86%	5.26%	3.85%	5.57%

The first column of Table 2 gives the mLFR settings, for example, *Left5_25.0* means that for each current frame, the features are stacked with 5 frames to the left (6 frames in total) and downsampled to 25.0 Hz (with a frame subsample factor of 4, resulting 40ms

frame advance). The first line in the table, *Left3_33.3* is the SpeechTransformer baseline, and the last line, *CERR* is the relative CER reduction of the best mLFR setting over the basic SpeechTransformer.

According to Table 2, it's very clear that with the frame rate decreasing, the performance of the SpeechTransformer firstly get improved and then decreased. The best performance is obtained at the frame rate of 16.7 Hz, with a relative 2.86%-5.57% CER reduction compared to the SpeechTransformer baseline (33.3 Hz) on the four test sets. We think the reason is that, much lower frame rate results in much shorter speech sequence length, making both the *encoder self-attention* and *decoder-encoder attention* much easier. What's more, the mLFR is quite beneficial to the decoding speed. Our experiments show that the SpeechTransformer with 16.7 Hz can achieve about 1.5 times speedup in the real time factor (RTF) over the 33.3 Hz baseline.

5.2. Scheduled Sampling

Based on the best mLFR setting (*Left7_16.7*), the scheduled sampling method is explored in this subsection. The first schedule we tried is *Linear* (Equation 1), with three different maximum sampling probabilities $\varepsilon_{\max}$ being 0.4, 0.6 and 1.0.

Intuitively, at the beginning, the sampling probability of the prediction should be rather small since the model is not well trained (otherwise it will add to too much noise and lead to very slow convergence). While at the later training, this probability should be quite large to sample enough useful examples since the prediction at this stage is almost all correct. To better simulate this process, we tried another two forms of schedules, *Exponential* (Equation 2) and *Sigmoid* (Equation 3), which gives smaller values at the beginning and larger ones near the end compared to *Linear* function. Table 3 gives all the results.

Table 3. CERs (%) of various scheduled sampling methods.

Models	Test Sets			
	AiShell_dev	AiShell_test	LiveShow	VoiceCom
Left7_16.7	4.41	5.22	29.7	8.13
+Lin_0.4	3.91	4.73	28.76	7.61
+Lin_0.6	**3.84**	**4.60**	**28.03**	**7.49**
+Lin_1.0	3.91	4.78	29.12	7.65
+Exp	3.98	4.74	28.98	7.75
+Sigmoid	3.95	4.78	28.44	7.71
CERR	12.93%	11.88%	5.62%	7.87%

Several observations can be found in Table 3. Firstly, the three schedule forms, *Linear* with $\varepsilon_{\max} = 1.0$, *Exponential* and *Sigmoid*, which have the same maximum sampling probability, give comparable performance and are better than the model trained without SS. Secondly, among the three settings of *Linear* form, the best performance can be obtained with $\varepsilon_{\max} = 0.6$, with relative gains of 5.62% to 12.93% on four test sets over the SpeechTransformer trained without SS (the *CERR* in the last line of Table 3).

5.3. Focal Loss

Based on the best configuration of mLFR and SS, we investigate the focal loss in this subsection. FL is introduced into the training process after the training accuracy achieves a high value, with the focusing parameter $\gamma = 2$ and the weighting factor $\alpha = 0.25$ (Equation 4). With this setting, if the prediction probability of a training example on the correct label is 0.90, the corresponding focal loss will be $100\times$ lower than the standard CE. This will in turn increase the importance of the misclassified examples. The results of FL and the final results compared to the hybrid and the SpeechTransformer baseline are listed in Table 4.

Table 4. CERs (%) of focal loss and the final results compare with the hybrid and speech transformer baseline.

Model	Test Sets			
	AiShell_dev	AiShell_test	LiveShow	VoiceCom
Left7_16.7	4.41	5.22	29.7	8.13
+Lin_0.6	3.84	4.60	28.03	7.49
+FL	**3.38**	**4.07**	**27.54**	**7.35**
CERR	11.98%	11.52%	1.57%	1.87%
vs. Hybrid	19.14%	18.44%	12.21%	12.40%
vs. ST	25.55%	26.13%	10.84%	14.63%

In Table 4, the last third line is the relative gain of FL over the model of *Lin_0.6*. While the bottom two lines are the relative CER reduction of the final improved SpeechTransformer model compared to the hybrid and the SpeechTransformer baselines, respectively. According to the table, FL gives 1.5%-11.9% relative gains, demonstrating its effectiveness.

Overall, the proposed three improvements to the SpeechTransformer bring 10.8%-26.1% relative gains on the four test sets. Compared to the strong hybrid baseline system, the resulting improved SpeechTransformer gives 12.2%-19.1% relative CER reduction, without any explicit language model. These experimental results show the great potential of the SpeechTransformer.

6. CONCLUSIONS

The SpeechTransformer, with self-attention mechanism, is a newer attention-based encoder-decoder architecture that has been examined on small-scale speech recognition tasks. It integrates acoustic, language and pronunciation model into a single neural network and has no-recurrence. In this work, we explore several optimization strategies to further improve the performance and efficiency of the SpeechTransformer on a large-scale Mandarin Chinese speech recognition task. Cumulatively, the proposed three optimization mechanism yield 10.8%-26.1% relative improvements in CER over the basic SpeechTransformer. The final optimized model shows 12.2%-19.1% relative CER reduction compared to a strong hybrid TDNN-LSTM system trained with LF-MMI criterion decoded with a large 4-gram LM.

We note, however, the SpeechTransformer has a very high latency, with a limitation that the entire utterance must be seen by the encoder, before any labels can be decoded. Therefore, an important next step is to revise this model with an streaming attention-based model, such as Neural Transducer [29]. We also find that the SpeechTransformer is more hungry for training data than the hybrid model (not shown in this work). We plan to add more data to train the model. Moreover, integrating the language model into the SpeechTransformer is also a valuable research direction.

7. REFERENCES

[1] Chao Weng, Jia Cui, Guangsen Wang, Jun Wang, Chengzhu Yu, Dan Su, and Dong Yu, "Improving attention based sequence-to-sequence models for end-to-end english conversational speech recognition," in *Proc. Interspeech 2018*, 2018, pp. 761–765.

[2] Alex Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[3] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al., "Deep speech: Scaling up end-to-end speech recognition," *arXiv:1412.5567*, 2014.

[4] Hagen Soltau, Hank Liao, and Hasim Sak, "Neural speech recognizer: Acoustic-to-word lstm model for large vocabulary speech recognition," *arXiv preprint arXiv:1610.09975*, 2016.

[5] Kartik Audhkhasi, Brian Kingsbury, Bhuvana Ramabhadran, George Saon, and Michael Picheny, "Building competitive direct acoustics-to-word models for english conversational speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4759–4763.

[6] Jinyu Li, Guoli Ye, Rui Zhao, Jasha Droppo, and Yifan Gong, "Acoustic-to-word model without oov," in *Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 111–117.

[7] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "End-to-end continuous speech recognition using attention-based recurrent nn: first results," *arXiv preprint arXiv:1412.1602*, 2014.

[8] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.

[9] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.

[10] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *International Conference onAcoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[11] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[13] Linhao Dong, Shuang Xu, and Bo Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.

[14] Shiyu Zhou, Linhao Dong, Shuang Xu, and Bo Xu, "Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese," in *Proc. Interspeech 2018*, 2018, pp. 791–795.

[15] Shiyu Zhou, Linhao Dong, Shuang Xu, and Bo Xu, "A comparison of modeling units in sequence-to-sequence speech recognition with the transformer on mandarin chinese," *arXiv preprint arXiv:1805.06239*, 2018.

[16] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.

[17] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[18] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi.," in *Interspeech*, 2016, pp. 2751–2755.

[19] Vijayaditya Peddinti, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur, "Low latency acoustic modeling using temporal convolution and lstms," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 373–377, 2018.

[20] Golan Pundak and Tara N Sainath, "Lower frame rate neural network acoustic models.," in *Interspeech*, 2016, pp. 22–26.

[21] Haşim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," *arXiv preprint arXiv:1507.06947*, 2015.

[22] Olivier Siohan, "Ctc training of multi-phone acoustic models for speech recognition," *Proc. Interspeech 2017*, pp. 709–713, 2017.

[23] Daniel Povey, Hossein Hadian, Pegah Ghahremani, Ke Li, and Sanjeev Khudanpur, "A time-restricted self-attention layer for asr," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5874–5878.

[24] Matthias Sperber, Jan Niehues, Graham Neubig, Sebastian Stüker, and Alex Waibel, "Self-attentional acoustic models," in *Proc. Interspeech 2018*, 2018, pp. 3723–3727.

[25] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*. IEEE, 2017, pp. 1–5.

[26] Daniel Povey, Xiaohui Zhang, and Sanjeev Khudanpur, "Parallel training of dnns with natural gradient and parameter averaging," *arXiv preprint arXiv:1410.7455*, 2014.

[27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[28] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[29] Tara N Sainath, Chung-Cheng Chiu, Rohit Prabhavalkar, Anjuli Kannan, Yonghui Wu, Patrick Nguyen, and ZhiJeng Chen, "Improving the performance of online neural transducer models," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5864–5868.