

WNARS: WFST based Non-autoregressive Streaming End-to-End Speech Recognition

Zhichao Wang, Wenwen Yang, Pan Zhou, Wei Chen

AI Interaction Division, Sogou Inc., Beijing, P.R.China

{wangzhichao214232, yangwenwen, zhoupan, chenweibj8871}@sogou-inc.com

Abstract

Recently, attention-based encoder-decoder (AED) end-to-end (E2E) models have drawn more and more attention in the field of automatic speech recognition (ASR). AED models, however, still have drawbacks when deploying in commercial applications. Autoregressive beam search decoding makes it inefficient for high-concurrency applications. It is also inconvenient to integrate external word-level language models. The most important thing is that AED models are difficult for streaming recognition due to global attention mechanism. In this paper, we propose a novel framework, namely WNARS, using hybrid CTC-attention AED models and weighted finite-state transducers (WFST) to solve these problems together. We switch from autoregressive beam search to CTC branch decoding, which performs first-pass decoding with WFST in chunk-wise streaming way. The decoder branch then performs second-pass rescoring on the generated hypotheses non-autoregressively. On the AISHELL-1 task, our WNARS achieves a character error rate of 5.22% with 640ms latency, to the best of our knowledge, which is the state-of-the-art performance for online ASR. Further experiments on our 10,000-hour Mandarin task show the proposed method achieves more than 20% improvements with 50% latency compared to a strong TDNN-BLSTM lattice-free MMI baseline.

Index Terms: ASR, non-autoregressive, streaming end-to-end, CTC-attention, WFST

1. Introduction

Traditional automatic speech recognition (ASR) systems usually consist of three components: acoustic model (AM), pronunciation model (PM) and language model (LM). End-to-end (E2E) ASR systems provide an elegant way to combine these components into a single network and significantly simplify the training process. There are three major types of E2E architectures in ASR: connectionist temporal classification (CTC) [1, 2, 3], recurrent neural network transducer (RNN-T) [4, 5, 6] and attention based encoder-decoder (AED) [7, 8, 9]. Recent study proposed a hybrid CTC-attention architecture which combines CTC and AED into a single model and has achieved promising results in many ASR tasks [10, 11].

The CTC-attention architecture introduces a CTC objective to the encoder during training of AED models, which encourages the alignments between the inputs and outputs of AED models to be monotonic [10]. During inference, a one-pass decoding method is proposed, which combines CTC, attention and an additional recurrent neural network language model (RNN-LM) to generate final results [11]. While this hybrid architecture has made significant progress in ASR, there are three challenges to tackle in practical applications:

1. **Decoding efficiency:** The CTC-attention model is decoded in an autoregressive way. Each output token is

generated iteratively by conditioning on previously generated tokens. The decoder has to compute as many times as the output length, which is time consuming.

2. **LM integration:** Also, the CTC-attention model is inconvenient to take advantage of word-level LMs due to the autoregressive problem. In [11], a character-level RNN-LM is integrated to do one pass decoding. However, in speech recognition, word-level LMs are much more powerful than character-level LMs.
3. **Streaming decoding:** Since the input and output sequences of AED models are of different lengths, the attention module inside decoder has to pay attention to the entire encoder outputs to generate soft alignments between input and output sequences. This makes it difficult to do streaming decoding.

Previous work usually focused on solving one of these problems. For example, in [12], a two-pass decoding framework is proposed to achieve non-autoregressive decoding. The target sequence is initialized with the greedy CTC outputs and low-confidence tokens are masked based on the CTC probabilities. These masked tokens are then predicted by decoder conditioned on the entire input sequence in a few fixed steps. In [13, 14, 15], multiple LM integration schemes have been proposed for E2E models. All these works focus on the character-level LM integration and it is still not clear how to integrate additional word-level LMs into AED models. In [16, 17, 18, 19, 20], efforts have been made to convert full sequence soft attention inside the decoder into local chunk-wise attention, which is suitable for streaming decoding. The key element of previous streaming AED methods is predicting boundary in encoder output for local attention of each token [16, 17, 18, 20] or predicting number of tokens in each fix-length chunk [19] for sub-sentence level attention.

In this work, we adopt the CTC-attention architecture and propose a novel WFST-based non-autoregressive streaming speech recognition (WNARS) framework to solve the above three problems together. The CTC branch equipped with WFST for better word level language model integration, performs as a first-pass decoder, which is responsible for streaming decoding. Thus we convert a streaming local attention problem to a streaming CTC problem. The decoder branch then rescoring the generated hypotheses conditioned on the entire encoder outputs in non-autoregressive way. Experimental results on the AISHELL-1 and our 10,000-hour industrial Mandarin tasks indicate the effectiveness of our method.

2. Model architecture

The following subsections first give a brief review of the attention-based encoder-decoder architecture. Then the hybrid CTC-attention training loss is explained.

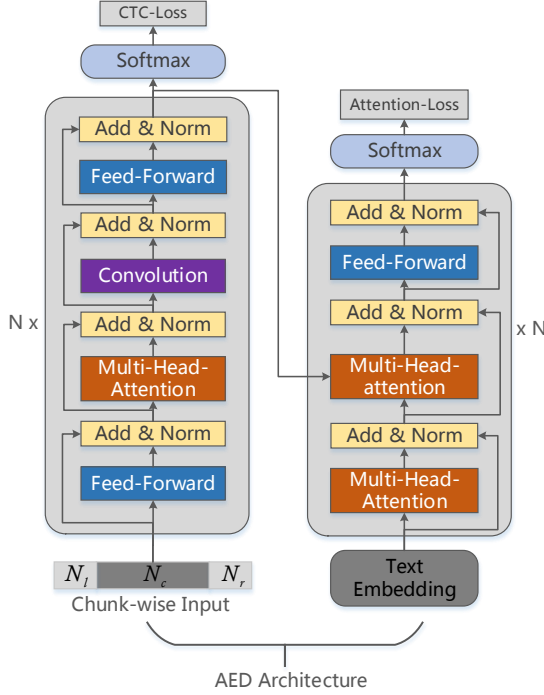


Figure 1: CTC-attention based AED model architecture.

2.1. Attention-based encoder-decoder

The AED architecture consists of an encoder and a decoder. The encoder learns a hidden representation $\mathbf{H} = (h_u | u = 1, \dots, U)$ from a speech feature sequence $\mathbf{X} = (x_t | t = 1, \dots, T)$, while the decoder generates an output sequence $\mathbf{Y} = (y_l | l = 1, \dots, L)$ based on \mathbf{H} and previously emitted tokens. It could be formulated as:

$$P_{att}(\mathbf{Y}|\mathbf{X}) = \prod_{l=1}^L p(y_l | y_1, \dots, y_{l-1}, \mathbf{H}) \quad (1)$$

There are many variants of the AED architecture [8, 21, 22]. Our work adopts the Conformer structure as shown in figure 1. A Conformer encoder-block is composed of four modules, namely, a feed-forward module, a multi-head self-attention module, a convolution module and a second feed-forward module in the end. The multi-head self-attention module captures the global context while the convolution module exploits the local correlations synchronously. A Conformer decoder-block consists of three modules, i. e., a feed-forward module, a multi-head self-attention module and a source-target attention module. The source-target attention module captures the alignments between input and output sequences. The encoder-block and decoder-block are stacked multiple times to achieve good performance.

2.2. Hybrid CTC-attention loss

In [10], researchers proposed a hybrid CTC-attention loss to improve the AED model. As shown in figure 1, a CTC objective function is attached to the shared encoder as an auxiliary task. The forward-backward algorithm of CTC can enforce monotonic alignment between speech and label sequences, which helps to reduce irregular alignments of AED models and leads to better performance. The hybrid training loss can be defined

as follows:

$$L_{CTC-attention} = \lambda L_{CTC} + (1 - \lambda) L_{attention} \quad (2)$$

where λ is a hyper-parameter: $0 \leq \lambda \leq 1$.

3. The proposed methods

In this section, we describe the proposed WNARS framework in detail.

3.1. WFST-based non-autoregressive decoding

The key to accelerate the decoding efficiency of AED models is to release the model from autoregressive decoding. Fortunately, the CTC-attention architecture gives us a chance to implement non-autoregressive decoding flexibly with a two-pass decoding framework. Like [12], we employ CTC to generate initial hypotheses and use attention-decoder to correct the results. Unlike [12], during CTC decoding at the first-pass stage, a n-gram word-level LM is integrated by a WFST to generate n-best hypotheses. Then, hypotheses are rescored by the attention-decoder in a teaching-forcing manner at the second pass stage. The two-pass hypotheses scores are combined using tunable hyper-parameters α and β to generate final results. Figure 2. shows the proposed WFST-based non-autoregressive decoding framework. With this framework, we not only bypass the autoregressive problem, but also explore a way to integrate a word-level language model into AED models.

The search graph we used in the first-pass decoding is built by three individual WFSTs as in [3]. A grammar WFST denoted as G encodes the permissible word sequences in a language (word-level language model). A lexicon WFST denoted as L encodes the mapping from sequences of lexicon units (Chinese character in this work) to words. A token WFST denoted as T maps a sequence of frame-level CTC labels to a single lexicon unit. After all, the search graph S is compiled as follows:

$$S = T \circ \min(\det(L \circ G)) \quad (3)$$

where \circ , \det and \min denote composition, determinization and minimization respectively.

3.2. Chunk-wise streaming processing

To tackle the streaming problem of AED models, the most common approach is introducing chunk-wise processing for both encoder and decoder. However, chunk-wise processing for decoder has to predict boundary in encoder output for local attention of each token or predict number of tokens in each chunk for sub-sentence level attention, which may introduce additional errors. With the proposed decoding framework described in section 3.1, we only need to do chunk-wise processing for encoder during CTC decoding, leaving the decoder to process global attention during rescoring. In our framework, the encoder is processed as in [23], the input sequence is split into isolated chunks of length N_c . To compensate contextual information, historical context N_l and future context N_r are appended to the left and right of N_c , as shown in Figure 1. The latency of the model is then controlled by $N_c + N_r$. Similar to [24], We use multi-chunk strategy, which means N_l , N_c , N_r are variable during training. In this way, we can obtain a streaming speech recognizer to satisfy different latency demands with by controlling N_c and N_r during inference.

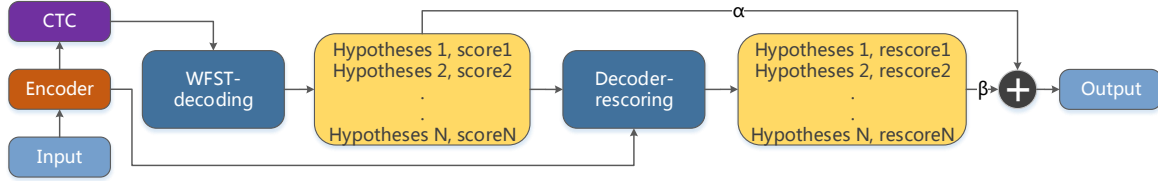


Figure 2: WFST-based non-autoregressive decoding framework.

4. Experiments

4.1. Datasets

To evaluate the effectiveness of our proposed methods, we carried out experiments on two Mandarin datasets, namely the open source AISHELL-1 corpus [25] and our own 10,000-hour Mandarin corpus. AISHELL-1 consists of three parts, a 150-hour training-set, a 10-hour development-set and a 5-hour test-set. The 10,000-hour corpus consists of 12 million utterances, collected from multi-domain, including dictation, voice-search, natural conversation, education, report et al. The data is anonymized without personal identifiable information. In the 10,000-hour corpus, 100 hours were randomly selected as the development set, a 25-hour close-talk test-set and a 10-hour far-field test-set were used for evaluation.

4.2. Experimental setup

We used the CTC-attention based Conformer model in all setups. The input acoustic features used for all experiments were 80-dimensional log-mel filter-banks computed on 25ms window with 10ms shift. Downsampling was performed through two convolutional layers with a factor of 4. Acoustic modeling units were the Chinese characters, which were 4233 and 6786 for AISHELL-1 and 10000-hour tasks respectively. For the training process, the Adam optimizer and the Noam [21] learning rate schedule were used, with warm-up steps of 25000. We also employed label smoothing and dropout regularization of 0.1 to prevent overfitting. Besides, the weight of the CTC loss was set to 0.3. All models were trained with 8 NVIDIA TESLA V100 GPUs, using the ESPNET toolkit [26] with the PyTorch backend.

4.3. AISHELL-1 Results

For AISHELL-1, speed-perturbation was applied to generate 3-fold versions of the data with factors of 0.9, 1.0 and 1.1. The model consisted of 12 encoder layers and 6 decoder layers. The feed-forward dimension was 2048 and the attention dimension was 256. Both the encoder and the decoder used multi-head of 4. During training, SpecAugment [27] was applied on-the-fly to improve the robustness of the model. Models were trained for 50 epochs and we averaged the 10-best models based on the accuracy on development-set to generate the final models. Besides, two language models were trained using the transcriptions of the training set. A character-level two-layer LSTM-LM (650 memory cells for each layer) was used for beam search decoding, while a 4-gram word-level LM was employed in our WFST-based decoding process.

4.3.1. Decoding performance

We first examined the performance of our proposed WFST-based non-autoregressive decoding approach on a non-streaming model. Our first experiment evaluated the impact

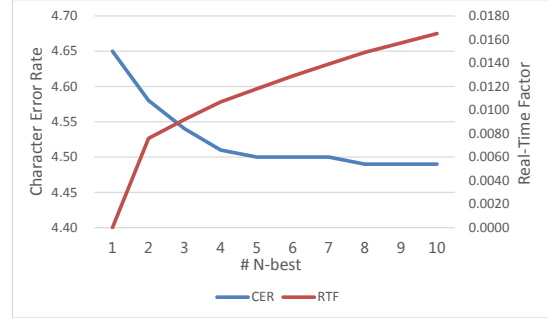


Figure 3: Curves of character error rate and real-time factor with the number of n-best.

Table 1: Decoding comparison

Decoding Strategy	CER	RTF
Joint CTC-attention Beam Search	4.7	0.74
WFST-based Non-autoregressive	4.5	0.1
CTC	5.2	-
CTC+LSTM-LM	4.8	-
CTC+WFST	4.65	-

of the number of n-best hypotheses used for attention-decoder rescoring. Our observations are illustrated in Figure 3. As we can see, the CER drops significantly when n-best increases from 1 to 5 (1 means no rescoring applied), while the CER stays stable when nbest is greater than 5. However, the rescoring time was proportional to the number of n-best. In the following experiments we chose 5-best to balance the recognition accuracy and decoding efficiency.

Table 1 shows the performance of the WFST-based non-autoregressive decoding approach on AISHELL-1 test-set. For comparison, the standard joint CTC-attention beam search method [11] was used as the baseline system. The character-level LSTM-LM was used during beam search progress in a shallow fusion manner. Hyper-parameters for beam search were: beam=10, ctc-weight=0.5, lm-weight=0.3. As can be seen in Table 1, the WFST-based non-autoregressive decoding approach achieves a CER of 4.5%, with a relative improvement of 4.3% compared to the baseline. Decoding efficiency of the two decoding strategy was also investigated in terms of RTF using a single thread with Intel(R) Xeon(R) E5-2630 v4 CPU, 2.20GHz. We can see the proposed method is 7.4x faster than the baseline.

To evaluate the effectiveness of WFST-based word-level LM integration approach, we discarded the attention-decoder and only used the CTC branch for decoding. As for comparison, two baseline systems were built: one used CTC prefix

Table 2: CERs with different Latency configuration on AISHELL-1 test-set

Model	N_l	N_c	N_r	latency(ms)	CER
Non-streaming	-	-	-	-	4.5
	160	64	32	960	5.15
	160	64	24	880	5.17
	160	64	16	800	5.21
	160	48	32	800	5.17
streaming	160	48	24	720	5.20
	160	48	16	640	5.25
	160	32	32	640	5.22
	160	32	24	560	5.26
	160	32	16	480	5.33

beam search without any LMs, another used shallow fusion to integrate the character-level LSTM-LM during prefix beam searching process. We can see from Table 1, both LM integration approaches can improve the performance in CER. The shallow fusion approach with LSTM-LM improved the CER by 7.7% in relative, while the word-level LM integrated with WFST brought a more improvement by 10.6%.

Results in Table 1 support the advantage of the WFST-based non-autoregressive decoding approach, both in speech recognition accuracy and efficiency.

4.3.2. Streaming processing

Our streaming model was initialized with the non-streaming model to accelerate convergence. We employed multi-chunk strategy to process the input sequence as mentioned in section 3.2 and the chunk-size configuration was: $\{N_l = (80, 100, 160), N_c = (32, 48, 64), N_r = (16, 24, 32)\}$. During training, $N_x(x = l, c, r)$ randomly selected a number from the candidates, and a total of 27 combinations of $\{N_l, N_c, N_r\}$ were generated.

In Table 2, we investigated our WNARS system by varying the latency configurations. During inference, we fixed $N_l = 160$ as it has nothing to do with the model latency. As we can see, our best streaming model achieved a CER of 5.15%, with a 960 ms latency and a 0.65% absolute CER degradation compared with the non-streaming baseline. From the results in Table 2, we can draw conclusions: 1) With the benefits of multi-chunk processing strategy, our WNARS system can achieve robust performance under different latency configurations. Only a 0.18% absolute CER degradation were observed when latency reduced from 960ms to 480ms. 2) With the same latency, using longer N_r can bring better performance, as the model can take use of more future information.

Finally, in Table 3, we compared our WNARS system with several published streaming E2E solutions on AISHELL-1. For a fair comparison, the latency of the systems listed in Table 3 is around 600 ms. We can see our proposed method achieved the best result. To our best knowledge, this is the state-of-the-art performance for streaming ASR systems on this task.

4.4. 10,000-hour Results

We verified the performance of the WNARS framework under a larger dataset, namely the 10,000-hour Mandarin dataset mentioned in 4.1. In this experiment, the attention dimension and the multi-head number of AED model was set to 384 and 6 re-

Table 3: Comparison with published online-E2E systems on AISHELL-1

Model	Latency(ms)	CER
SCAMA[19]	600	7.39
MMA[28]	640	6.60
U2[24]	640	5.42
WNARS	640	5.22

Table 4: Comparison with conventional ASR system on 10000-hour task

Model	Latency(ms)	close-talk	far-field
LFMMI-sMBR	1900	8.2	25.5
WNARS	960	6.4	20.4

spectively, while no speed-perturbation and SpecAugment were applied. As for comparison, we built a conventional ASR system with a TDNN-BLSTM lattice-free MMI (LFMMI) model [29]. The model consisted of 9 TDNN layers (1536 hidden nodes for each layer with splice [-3, 0, 3]) and 4 BLSTM layers (1280 memory cells and 512 projections for each direction), with 4528 2-state context dependent phones as modeling units. After LFMMI training, sMBR [30] training was applied to obtain a strong baseline. Both systems employed an external 5-gram word-level LM during WFST-decoding process.

As shown in Table 4, our WNARS system achieved a significant improvement in CER over the conventional ASR system, both on close-talk test-set and far-field test-set. Besides, the WNARS system has a much lower latency.

5. Conclusions

In this work, we propose a novel WNARS framework to solve decoding efficiency, LM integration and streaming decoding problems of AED end-to-end model. With the help of WFST and multi-chunk strategy, our WNARS is robust under different inference latency. The proposed framework reaches a CER of 5.22% with 640ms latency, which is the state-of-the-art streaming performance on AISHELL-1 task under comparable latency. Even with 50% latency, WNARS still obtains over 20% gain over a strong traditional industrial ASR system on a 10,000-hour Mandarin task, which certifies the effectiveness, practicality and generalization of our proposal.

6. Acknowledgements

The authors gratefully acknowledge the financial support provided by the National Key Research and Development Program of China (2020AAA0108004).

7. References

- [1] 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*, 2006, pp. 369–376.
- [2] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english

- and mandarin,” in *International conference on machine learning*. PMLR, 2016, pp. 173–182.
- [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] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.
 - [5] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang *et al.*, “Streaming end-to-end speech recognition for mobile devices,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6381–6385.
 - [6] K. Rao, H. Sak, and R. Prabhavalkar, “Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer,” in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 193–199.
 - [7] 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.
 - [8] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, “Listen, attend and spell,” *arXiv preprint arXiv:1508.01211*, 2015.
 - [9] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” *arXiv preprint arXiv:1506.07503*, 2015.
 - [10] S. Kim, T. Hori, and S. Watanabe, “Joint ctc-attention based end-to-end speech recognition using multi-task learning,” in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.
 - [11] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, “Advances in joint ctc-attention based end-to-end speech recognition with a deep cnn encoder and rnn-lm,” *arXiv preprint arXiv:1706.02737*, 2017.
 - [12] Y. Higuchi, S. Watanabe, N. Chen, T. Ogawa, and T. Kobayashi, “Mask ctc: Non-autoregressive end-to-end asr with ctc and mask predict,” *arXiv preprint arXiv:2005.08700*, 2020.
 - [13] S. Toshniwal, A. Kannan, C.-C. Chiu, Y. Wu, T. N. Sainath, and K. Livescu, “A comparison of techniques for language model integration in encoder-decoder speech recognition,” in *2018 IEEE spoken language technology workshop (SLT)*. IEEE, 2018, pp. 369–375.
 - [14] C. Shan, C. Weng, G. Wang, D. Su, M. Luo, D. Yu, and L. Xie, “Component fusion: Learning replaceable language model component for end-to-end speech recognition system,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5361–5365.
 - [15] W. Michel, R. Schlüter, and H. Ney, “Early stage lm integration using local and global log-linear combination,” *arXiv preprint arXiv:2005.10049*, 2020.
 - [16] C.-C. Chiu and C. Raffel, “Monotonic chunkwise attention,” *arXiv preprint arXiv:1712.05382*, 2017.
 - [17] R. Fan, P. Zhou, W. Chen, J. Jia, and G. Liu, “An online attention-based model for speech recognition,” *arXiv preprint arXiv:1811.05247*, 2018.
 - [18] K. Kim, K. Lee, D. Gowda, J. Park, S. Kim, S. Jin, Y.-Y. Lee, J. Yeo, D. Kim, S. Jung *et al.*, “Attention based on-device streaming speech recognition with large speech corpus,” in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 956–963.
 - [19] S. Zhang, Z. Gao, H. Luo, M. Lei, J. Gao, Z. Yan, and L. Xie, “Streaming chunk-aware multihead attention for online end-to-end speech recognition.”
 - [20] E. Tsunoo, Y. Kashiwagi, and S. Watanabe, “Streaming transformer asr with blockwise synchronous inference,” *arXiv preprint arXiv:2006.14941*, 2020.
 - [21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *arXiv preprint arXiv:1706.03762*, 2017.
 - [22] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, “Conformer: Convolution-augmented transformer for speech recognition,” *arXiv preprint arXiv:2005.08100*, 2020.
 - [23] H. Miao, G. Cheng, C. Gao, P. Zhang, and Y. Yan, “Transformer-based online ctc/attention end-to-end speech recognition architecture,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6084–6088.
 - [24] B. Zhang, D. Wu, Z. Yao, X. Wang, F. Yu, C. Yang, L. Guo, Y. Hu, L. Xie, and X. Lei, “Unified streaming and non-streaming two-pass end-to-end model for speech recognition,” *arXiv preprint arXiv:2012.05481*, 2020.
 - [25] 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 *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] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, “Espnet: End-to-end speech processing toolkit,” *arXiv preprint arXiv:1804.00015*, 2018.
 - [27] 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,” *arXiv preprint arXiv:1904.08779*, 2019.
 - [28] H. Inaguma, M. Mimura, and T. Kawahara, “Enhancing monotonic multihead attention for streaming asr,” *arXiv preprint arXiv:2005.09394*, 2020.
 - [29] Y. Qin, T. Lee, S. Feng, and A. P.-H. Kong, “Automatic speech assessment for people with aphasia using tdnn-blstm with multi-task learning,” in *INTERSPEECH*, 2018, pp. 3418–3422.
 - [30] B. Kingsbury, “Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling,” in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 3761–3764.