

# NON-AUTOREGRESSIVE TRANSFORMER ASR WITH CTC-ENHANCED DECODER INPUT

Xingchen Song<sup>1,†</sup>, Zhiyong Wu<sup>1,2,‡</sup>, Yiheng Huang<sup>3</sup>, Chao Weng<sup>3</sup>, Dan Su<sup>3</sup>, Helen Meng<sup>1,2</sup>

<sup>1</sup>Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen

<sup>2</sup>The Chinese University of Hong Kong, Hong Kong

<sup>3</sup>Tencent AI Lab, Shenzhen

sxc19@mails.tsinghua.edu.cn<sup>†</sup>, zyw@se.cuhk.edu.hk<sup>‡</sup>

## ABSTRACT

*Non-autoregressive* (NAR) transformer models have achieved significantly inference speedup but at the cost of inferior accuracy compared to *autoregressive* (AR) models in automatic speech recognition (ASR). Most of the NAR transformers take a fixed-length sequence filled with MASK tokens or a redundant sequence copied from encoder states as decoder input, they cannot provide efficient target-side information thus leading to accuracy degradation. To address this problem, we propose a CTC-enhanced NAR transformer, which generates target sequence by refining predictions of the CTC module. Experimental results show that our method outperforms all previous NAR counterparts and achieves **50x** faster decoding speed than a strong AR baseline with only 0.0 ~ 0.3 **absolute** CER degradation on Aishell-1 and Aishell-2 datasets.

**Index Terms**— non-autoregressive, transformer, etc

## 1. INTRODUCTION

Recently, the self-attention based encoder-decoder framework, called transformer [1], has achieved very promising results for automatic speech recognition [2] when comparing to traditional hybrid models [3] and CTC based models [4]. However, such model suffers from a high latency during the inference process as it translates a source sequence in an *autoregressive* manner, i.e.(see Fig.1(a)), it generates a target sentence character by character from left to right and the generation of  $t$ -th token  $y_t$  depends on previously generated tokens  $y_{1:t-1}$  and encoded audio representations  $\mathcal{E}(x)$ :

$$y_t = \mathcal{D}(y_{1:t-1}, \mathcal{E}(x)) \quad (1)$$

where  $\mathcal{E}(\cdot)$  and  $\mathcal{D}(\cdot)$  denote the encoder and decoder part of the model respectively,  $x$  is the input audio features and  $\mathcal{E}(x)$  is the output of encoder, i.e., a series of encoded hidden states at the top layer of the encoder.

To speed up the inference of speech recognition, *non-autoregressive* transformers have been proposed [5, 6, 7, 8], which generate all target tokens simultaneously [5, 6] or iteratively [7, 8]. We notice that the encoder of AR transformers and that of NAR transformers are the same thus the dif-

ferences lie in the decoder. More specifically, instead of using previously generated tokens as in AR decoders, NAR decoders take other global signals derived from either empirical statistics [5, 8] or encoded speech sequence [6].

For example, the NAR decoders in *listen attentively and spell once* (LASO) [5] and *listen and fill in missing letter* (LFML) [8] take fixed-length sequence filled with MASK tokens as input to predict target sequence:

$$y_t = \mathcal{D}(\text{MASK}_{1:L}, \mathcal{E}(x)) \quad (2)$$

where  $L$  is a constant number that was predefined and used for all sentences. The settings of this predefined length is usually based on empirical analysis of durations of audio and the final output length is determined by the position where the first end-of-sequence (EOS) token appears. To guarantee the performance of the model, the predefined length  $L$  is often much longer than the actual length of the target sequence  $T$ . This results in extra calculation cost and affects the inference speed. In order to estimate the length of the target sequence accurately and accelerate the inference speed, [6] proposed a *spike-triggered non-autoregressive* (ST-NAR) model, which introduces a CTC module [4] to predict the target length and takes a copy of encoder hidden states  $\hat{\mathcal{E}}(x)$  as the decoder input, and the copy process is guided by CTC spikes, which represent positions of the state that should be copied. After that all target tokens are simultaneously predicted:

$$y_t = \mathcal{D}(\hat{\mathcal{E}}(x), \mathcal{E}(x)) \quad (3)$$

where  $\hat{\mathcal{E}}(x) = (\mathcal{E}(x)_{p_1}, \mathcal{E}(x)_{p_2}, \dots, \mathcal{E}(x)_{p_{T'}})$  are the copied encoder states.  $p_i$  is the position where CTC spike occurs and  $T'$  is the number of CTC spikes, it can be seen as the target length predicted by CTC.

While ST-NAR models significantly reduce the inference latency, they suffer from large accuracy degradation compared with their AR counterparts. We notice that in AR models, the generation of  $t$ -th token  $y_t$  is conditioned on previously generated tokens  $y_{1:t-1}$ , which provides target side context information. In contrast, ST-NAR [6] models generate tokens conditioned on the copied encoder states  $\hat{\mathcal{E}}(x)$  guided by spikes, such information is redundant and indirect because the copied states are still in the domain of audio features.

<sup>†</sup> Work performed during an internship at Tencent AI Lab.

<sup>‡</sup> Corresponding author.

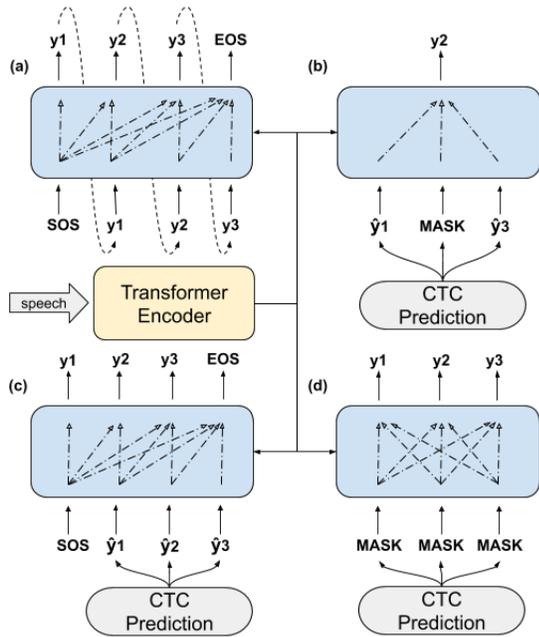
Consequently, the decoder of ST-NAR model has to handle the recognition task conditioned on less and weaker information compared with its AR counterparts, thus leading to inferior accuracy. In this paper, we aim to enhance the decoder inputs of NAR models so as to reduce the difficulty of the task that the decoder needs to handle. Our basic idea is to directly feed greedy CTC predictions  $\hat{y} = CTC(\mathcal{E}(x))$  to the decoder. We conjecture that the CTC outputs can provide more closed information to the target-side tokens  $y$ , thus reducing the gap between AR and NAR models. Experimental results on two corpora demonstrate that our model outperforms all compared NAR models and exceeds strong AR baseline by **50x** faster speed with comparable performance.

## 2. METHODOLOGY

### 2.1. Related Work

Recently, [7] trains a Transformer encoder-decoder model with both Mask-Predict [9] and CTC objectives (MP-CTC). As shown in Fig.1(b), during inference, the input of decoder is initialized with the greedy CTC prediction  $\hat{y}$  and low-confidence tokens are masked based on the CTC probabilities. The masked low-confidence tokens  $y_l$  are then predicted by the decoder conditioning on the high-confidence tokens  $\hat{y}_h$ :

$$y_l = \mathcal{D}(\hat{y}_h, \mathcal{E}(x)) \quad (4)$$



**Fig. 1.** Comparison between AR and NAR transformer **during inference**. The blue boxes indicate Transformer Decoder and the dash-dotted lines inside them indicate the dependency between tokens. (a) AR Transformer. The dashed line limits the emission of  $y_t$  by waiting for the end of generating  $y_{t-1}$ . (b) MP-CTC [7]. (c) proposed NAR Transformer. (d) up-graded version of LASO [5].

While this approach use CTC predictions as decoder input as we do, the Mask-Predict [9] manner that was originally proposed for neural machine translation (NMT) task may not be suitable for ASR as it suffers from two drawbacks: 1) As pointed out in [10], the observed tokens (high-confidence CTC predictions) are *not* always correct and the masked tokens (low-confidence CTC predictions) are *not* always wrong, this may further increase mis-recognitions when re-decoding low-confidence but correct tokens to wrong tokens or making predictions based on high-confidence but incorrect tokens. 2) Previous works [11, 12] show that the decoder is more sensitive to target-side information than source-side information and ASR is more serious than NMT due to the different characteristics between the input acoustic signals and the output linguistic symbols [11]. This suggests that given the full context of audio features  $\mathcal{E}(x)$  and the full context of characters  $\hat{y}_h$  as in Eq.4, the mask-based character prediction can be easily filled out only with the target context without using any speech input (i.e., the network only learns the decoder). In this situation, the training-inference discrepancy, where the ground truth is used as decoder input during training and replaced by CTC prediction during inference, will largely affect the performance and this is verified by our study in Fig.2.

### 2.2. Proposed Method

As shown in Fig.1(c), we address the above problems by limiting the target context to the past thus forcing the model to pay more attention to source-side context  $\mathcal{E}(x)$ :

$$y_t = \mathcal{D}(\hat{y}_{1:t-1}, \mathcal{E}(x)) \quad (5)$$

It is interesting to note that Eq.5 is similar to its AR baseline described in Eq.1. The difference is that we can get all target tokens  $\hat{y}_{1:T'}$  before the decoder starts to translate while AR model cannot. This allows us to make the decoding process fully paralleled while maintaining the AR properties for each generated token and this is the reason that we can achieve **50x** speedup while capturing the performance of the AR counterpart.

Here, we propose two concrete methods to train transformer model to support the decoding type described in Eq.5:

1. *Teacher Forcing (CM)*: we feed ground truth to the decoder during training and CTC prediction during inference. CM means we use Causal Mask (CM) as decoder attention mask to discard target-side future context to obey the left-to-right rule.
2. *CTC Sampling (CM)*: during training, we perform on-the-fly CTC decoding to get predicted length  $T'$  and adjust the decoder input according to it. Specifically, if  $|T - T'| \leq 2$ , we use CTC prediction  $\hat{y}$ , otherwise we use ground truth as input.

### 2.3. Variations for Comparison

Beside the Causal Mask, we also replace it with Padding Mask (PM), which only masks the padded tokens in a mini-batch and is also used in MP-CTC [7], for *CTC Sampling* to remove the context restrictions to validate our conjecture that the full context of targets may lead to performance degradation due to the sensitivity of decoder towards target-side information. This can be formalized as:

$$y_t = \mathcal{D}(\hat{y}_{1:T'}, \mathcal{E}(x)) \quad (6)$$

To make a thorough comparison, we also proposed an upgraded version of LASO [5]. As shown in Fig.1(d) and Eq.7, we replace the predefined length  $L$  in Eq.2 with CTC predicted length  $T'$ :

$$y_t = \mathcal{D}(MASK_{1:T'}, \mathcal{E}(x)) \quad (7)$$

During training, ground truth is used as decoder input and all target tokens are replaced with MASK. In this case, target-side information only contains the length information and the semantic context can be retrieved only from the source-side information  $\mathcal{E}(x)$  [5]. We name this strategy as *Mask Forcing (PM)*. Note that during inference, the length information will be provided by CTC.

## 3. EXPERIMENTS

In this work, all experiments are conducted using ESPnet transformer [13]<sup>1</sup> with CTC joint training [2]. The model and experiment setups are the same as [2] except that we use SpecAugment [14] and set *total\_training\_epoch* = 200 for all experiments. Due to space limitations, the details are omitted and interested readers can refer to [2].

### 3.1. 178-hours Corpus: AISHELL-1

We first conduct experiments and analyses on a public mandarin speech corpus AISHELL-1 [15]. As shown in Table 1, our best strategy speeds up the AR baseline (beam=10) by **50x** times with only 0.0 ~ 0.2 absolute CER degradation and achieves **5x** faster speed than the greedy result (beam=1) with even better CER. We find that both *Teacher Forcing (CM)* and *MP-CTC (PM)* perform better than *Mask Forcing (PM)*, this is partly because the latter one cannot provide efficient target-side information. Besides, we also find that strategies with CM always outperform those with PM, i.e., *Teacher Forcing (CM)* and *CTC Sampling (CM)* work better than *MP-CTC (PM)* and *CTC Sampling (PM)*, respectively. This suggests that given the full context of speech features  $\mathcal{E}(x)$ , target-side context should be limited to avoid excessive reliance on it. Such reliance is caused by the aforementioned sensitivity of decoder and we validate it in Fig.2, where we

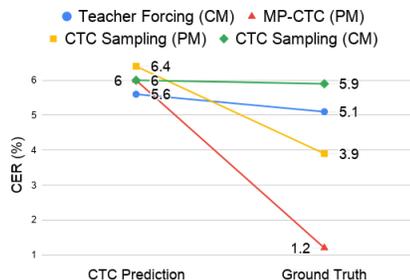
<sup>1</sup> <https://github.com/espnet/espnet>

<sup>2</sup> We re-implement MP-CTC [7] using official code retrieved from <https://github.com/espnet/espnet/pull/2070> and set the number of total decoding iterations to 5 as suggested in [7].

**Table 1.** The character error rate (CER) of the systems on AISHELL-1. Real-time factor (RTF) is computed as the ratio of the total inference time to the total duration of the dev set. Inference is done with *batchsize* = 8, on an NVIDIA Tesla P40 GPU.

| Training Strategy                      | Decode Type     | Beam | Dev / Test       | RTF           |
|--|-----------------|------|------------------|---------------|
| <i>Autoregressive Transformer</i>      |                 |      |                  |               |
| Baseline [2] (with CTC Joint Training) | Fig.1(a) & Eq.1 | 1    | 5.6 / 6.6        | 0.0186        |
|  |                 | 10   | <b>5.6 / 6.1</b> | 0.1703        |
| <i>Non-autoregressive Transformer</i>  |                 |      |                  |               |
| CTC Sampling (PM)                      | -- & Eq.6       | -    | 6.4 / 7.2        | 0.0037        |
| CTC Sampling (CM)                      | Fig.1(c) & Eq.5 | -    | 6.0 / 6.8        | 0.0037        |
| Mask Forcing (PM)                      | Fig.1(d) & Eq.7 | -    | 6.0 / 6.8        | 0.0037        |
| MP-CTC (PM) [7] <sup>2</sup>           | Fig.1(b) & Eq.4 | -    | 6.0 / 6.6        | 0.0134        |
| Teacher Forcing (CM)                   | Fig.1(c) & Eq.5 | -    | <b>5.6 / 6.3</b> | <b>0.0037</b> |

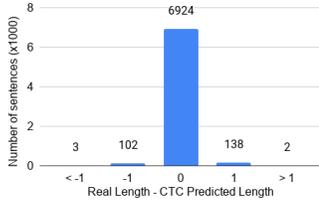
change the decoder input during inference to see the performance gap between ideal and reality, i.e., feed ground truth or CTC prediction to the decoder respectively. By comparing *CTC Sampling (CM)* and *CTC Sampling (PM)* in Fig.2, we can clearly see that PM based decoder is more sensitive to the input. As for *MP-CTC (PM)* and *CTC Sampling (PM)*, although both of them use PM as attention mask, the performance of the former one is more affected by the input. This is because the decoder of *MP-CTC (PM)* can only see ground truth during training thus suffers more from training-inference discrepancy. In contrast, *Teacher Forcing (CM)*, which limits the target-side context to obey the autoregressive manner, performs more stable and much closer to the upper bound of the trained model, indicating the effectiveness of our proposed method and we will use it for all our following experiments.



**Fig. 2.** The CER drop when change the decoder input. Note that for MP-CTC [7], the ground truth is obtained by replacing the non-masked tokens in CTC prediction with the corresponding correct tokens.

To study how CTC contributes to our NAR models, we analyze the predicted length by CTC. As shown in Fig.3, the accuracy of length prediction is higher than 96%, which means that  $T'$  is accurate enough to lower the computation cost when we upgrade Eq.2 to Eq.7

Besides, we also provide a comparison between CTC prediction and NAR Decoder prediction in Table 2. Due to the



**Fig. 3.** The analysis of the predicted length on test set. The histogram record the difference between the target length  $T$  and the CTC predicted length  $T'$ . It will not cause irreversible effects when the value is less then or equal to zero, as the decoder is still able to correct the CTC output by predicting EOS token at the end of sentence.

strong conditional independence assumption, CTC prediction has a poor performance. However, the decoder can still correct the input based on both speech context and partial target context, indicating that the target-side information provided by CTC is useful and efficient for our NAR decoder.

**Table 2.** CER comparison between CTC prediction and decoder prediction for proposed NAR transformer.

| Result                 | Dev / Test       |
|------------------------|------------------|
| CTC Prediction         | 6.0 / 6.7        |
| NAR Decoder Prediction | <b>5.6 / 6.3</b> |

Lastly, we give a CER comparison with other approaches in Table 3. To make a fair comparison, we also add Speed-Perturb [16] based augmentation to our training and approximate the number of parameters based on the description in the previous studies. It can be seen that our strong AR baseline reaches the state-of-the-art performance as it outperforms all previously published results and our proposed NAR model also beats all previous NAR counterparts. Considering the **50x** faster decoding speed, the gap between AR and NAR Transformer is tolerable.

**Table 3.** CER comparison with Hybrid and End-to-End models on Aishell-1 [15]. We use beam=10 for our AR model and keep it unchanged in our following experiments.  $\blacktriangle$ : SpecAugment [14] is used.  $\blacklozenge$ : Speed-Perturb [16] is used.

| Model                                | Dev / Test       | RTF           | Params          |
|--------------------------------------|------------------|---------------|-----------------|
| <i>End-to-End Autoregressive</i>     |                  |               |                 |
| LAS [17]                             | - / 8.7          | -             | $\approx$ 156 M |
| Transformer [5]                      | 6.1 / 6.6        | -             | $\approx$ 58 M  |
| <i>End-to-End Non-autoregressive</i> |                  |               |                 |
| CTC [6] $\blacktriangle$             | 7.8 / 8.7        | -             | -               |
| ST-NAR [6] $\blacktriangle$          | 6.9 / 7.7        | -             | $\approx$ 31 M  |
| MP-CTC [7] $\blacktriangle^2$        | 6.0 / 6.7        | 0.0134        | 29.7 M          |
| LASO(big) [5] $\blacklozenge$        | 5.8 / 6.4        | -             | $\approx$ 105 M |
| <i>Traditional Hybrid</i>            |                  |               |                 |
| Kaldi/nnet3 [3] $\blacklozenge$ + LM | - / 8.6          | -             | -               |
| Kaldi/chain [3] $\blacklozenge$ + LM | - / 7.5          | -             | -               |
| <i>Ours</i>                          |                  |               |                 |
| AR-Transformer $\blacktriangle$      | 5.6 / 6.1        | 0.1703        | 29.7 M          |
| NAR-Transformer $\blacktriangle$     | 5.6 / 6.3        | 0.0037        | 29.7 M          |
| AR-Transformer $\blacklozenge$       | <b>5.2 / 5.7</b> | 0.1703        | 29.7 M          |
| NAR-Transformer $\blacklozenge$      | 5.3 / 5.9        | <b>0.0037</b> | 29.7 M          |

### 3.2. 1000-hours Corpus: AISHELL-2

We then conduct our experiments on AISHELL-2 [18], which is by far the largest free corpus available for Mandarin ASR research and contains 1000 hours of reading speech for training. As far as we know, none of the previous NAR methods have been tested on such a large dataset thus we are the first to evaluate the effectiveness of NAR models on industrial level Large Vocabulary Continuous Speech Recognition (LVCSR).

In Table 4, it is worth noting that the state-of-the-art CER achieved by CIF [19] model uses nearly three times more parameters than ours and the AR baseline we used is still competitive since it performs much better than other models. By comparing the results of AR and NAR transformer, we can draw a similar conclusion as in Section 3.1 that our NAR model can achieve much faster decoding speed while maintaining the performance of its AR counterpart.

**Table 4.** CER comparison with Hybrid and End-to-End models on 1000 hours of reading corpus Aishell-2 [18].  $\blacktriangle$ : SpecAugment [14] is used.  $\blacklozenge$ : Speed-Perturb [16] is used.  $\blacktriangledown$ : Adversarial-Data-Augmentation [20] is used.

| Model                               | IOS / Mic / Android | RTF           | Params      |
|-------------------------------------|---------------------|---------------|-------------|
| <i>End-to-End Autoregressive</i>    |                     |               |             |
| LAS [20] $\blacklozenge$            | 9.2 / 10.3 / 9.7    | -             | -           |
| S2S [21] + LM                       | 8.5 / - / -         | -             | -           |
| CIF [19] $\blacktriangle$ + LM      | 5.8 / 6.3 / 6.2     | -             | 67 M + 22 M |
| <i>Traditional Hybrid</i>           |                     |               |             |
| tri3(LDA+MLLT) [18] $\blacklozenge$ | 19.8 / 21.1 / 21.0  | -             | -           |
| Chain-TDNN [18] $\blacklozenge$     | 8.8 / 10.9 / 9.6    | -             | -           |
| <i>Ours</i>                         |                     |               |             |
| AR-Transformer $\blacklozenge$      | 6.8 / 7.7 / 7.8     | 0.1703        | 29.7 M      |
| NAR-Transformer $\blacklozenge$     | 7.1 / 8.0 / 8.1     | <b>0.0037</b> | 29.7 M      |

## 4. CONCLUSION

In this paper, we proposed a novel Non-autoregressive Transformer with CTC-enhanced decoder input, which generates a sequence by refining the CTC prediction. The experimental comparisons demonstrate that on Aishell-1 and Aishell-2 tasks, our method can surpass all previous NAR models and approach the performance of AR models while achieving **50x** faster decoding speed.

## 5. ACKNOWLEDGEMENTS

This work was conducted when the first author was an intern at Tencent AI Lab, and is supported by National Natural Science Foundation of China (NSFC) (62076144) and joint research fund of NSFC-RGC (Research Grant Council of Hong Kong) (61531166002, N\_CUHK404/15). We would also like to thank Tencent AI Lab Rhino-Bird Focused Research Program (No. JR202041, JR201942) and Tsinghua University - Tencent Joint Laboratory for the support.

## 6. REFERENCES

- [1] Linhao Dong, Shuang Xu, and Bo Xu, “Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition,” in *ICASSP 2018*, Calgary, Canada, 2018, pp. 5884–5888, IEEE.
- [2] Shigeki Karita, Xiaofei Wang, and Shinji Watanabe, “A comparative study on transformer vs rnn in speech applications,” in *ASRU 2019*, Singapore, 2019, pp. 449–456, IEEE Signal Processing Society.
- [3] Povey Daniel, Ghoshal Arnab, Boulianne Gilles, Burget Lukas, Glembek Ondrej, Goel Nagendra, Hannemann Mirko, Motlicek Petr, Qian Yanmin, Schwarz Petr, Silovsky Jan, Stemmer Georg, and Vesely Karel, “The kaldi speech recognition toolkit,” in *ASRU 2011*, Hawaii, USA, 2011, pp. 1–4, IEEE Signal Processing Society.
- [4] Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *ICML 2006*, Pittsburgh, USA, 2006, pp. 369–376, ACM.
- [5] Ye Bai, Jiangyan Yi, Jianhua Tao, Zhengkun Tian, Zhengqi Wen, and Shuai Zhang, “Listen attentively, and spell once: Whole sentence generation via a non-autoregressive architecture for low-latency speech recognition,” *arXiv preprint*, 2020.
- [6] Zhengkun Tian, Jiangyan Yi, Jianhua Tao, Ye Bai, Shuai Zhang, and Zhengqi Wen, “Spike-triggered non-autoregressive transformer for end-to-end speech recognition,” *arXiv preprint*, 2020.
- [7] Yosuke Higuchi, Shinji Watanabe, Nanxin Chen, Tetsuji Ogawa, and Tetsunori Kobayashi, “Mask CTC: non-autoregressive end-to-end ASR with CTC and mask predict,” *arXiv preprint*, 2020.
- [8] Nanxin Chen, Shinji Watanabe, Jesús Villalba, and Najim Dehak, “Listen and fill in the missing letters: Non-autoregressive transformer for speech recognition,” *arXiv preprint*, 2019.
- [9] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer, “Mask-predict: Parallel decoding of conditional masked language models,” in *EMNLP-IJCNLP 2019*, Hong Kong, China, 2019, pp. 6111–6120, ACL.
- [10] Marjan Ghazvininejad, Omer Levy, and Luke Zettlemoyer, “Semi-autoregressive training improves mask-predict decoding,” *arXiv preprint*, 2020.
- [11] Yi Ren, Jinglin Liu, Xu Tan, Zhou Zhao, Sheng Zhao, and Tie-Yan Liu, “A study of non-autoregressive model for sequence generation,” in *ACL 2020*, Online, 2020, pp. 149–159, ACL.
- [12] Tianyu He, Xu Tan, and Tao Qin, “Hard but robust, easy but sensitive: How encoder and decoder perform in neural machine translation,” *arXiv preprint*, 2019.
- [13] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplín, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai, “Espnet: End-to-end speech processing toolkit,” in *Interspeech 2018*, Hyderabad, India, 2018, pp. 2207–2211, ISCA.
- [14] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Interspeech 2019*, Graz, Austria, 2019, pp. 2613–2617, ISCA.
- [15] 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 *O-COCOSDA 2017*, Seoul, South Korea, 2017, pp. 1–5, IEEE.
- [16] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, “Audio augmentation for speech recognition,” in *Interspeech 2015*, Dresden, Germany, 2015, pp. 3586–3589, ISCA.
- [17] Changhao Shan, Chao Weng, Guangsen Wang, Dan Su, Min Luo, Dong Yu, and Lei Xie, “Component fusion: Learning replaceable language model component for end-to-end speech recognition system,” in *ICASSP 2019*, Brighton, United Kingdom, 2019, pp. 5631–5635, IEEE.
- [18] Jiayu Du, Xingyu Na, Xuechen Liu, and Hui Bu, “AISHELL-2: transforming mandarin ASR research into industrial scale,” *arXiv preprint*, 2018.
- [19] Linhao Dong and Bo Xu, “CIF: continuous integrate-and-fire for end-to-end speech recognition,” in *ICASSP 2020*, Barcelona, Spain, 2020, pp. 6079–6083, IEEE.
- [20] Sining Sun, Pengcheng Guo, Lei Xie, and Mei-Yuh Hwang, “Adversarial regularization for attention based end-to-end robust speech recognition,” *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 11, pp. 1826–1838, 2019.
- [21] Cheng Yi, Feng Wang, and Bo Xu, “Etc-docd: An end-to-end structure with CTC encoder and OCD decoder for speech recognition,” in *Interspeech 2019*, Graz, Austria, 2019, pp. 4420–4424, ISCA.