



# A Lexical-aware Non-autoregressive Transformer-based ASR Model

Chong-En Lin<sup>1</sup>, Kuan-Yu Chen<sup>1,2</sup>

<sup>1</sup>Department of Computer Science and Information &  
<sup>2</sup>Empower Vocational Education Research Center,  
National Taiwan University of Science and Technology, Taiwan  
celin@nlp.csie.ntust.edu.tw, kychen@mail.ntust.edu.tw

## Abstract

Non-autoregressive automatic speech recognition (ASR) has become a mainstream of ASR modeling because of its fast decoding speed and satisfactory result. To further boost the performance, relaxing the conditional independence assumption and cascading large-scaled pre-trained models are two active research directions. In addition to these strategies, we propose a lexical-aware non-autoregressive Transformer-based (LA-NAT) ASR framework, which consists of an acoustic encoder, a speech-text shared encoder, and a speech-text shared decoder. The acoustic encoder is used to process the input speech features as usual, and the speech-text shared encoder and decoder are designed to train speech and text data simultaneously. By doing so, LA-NAT aims to make the ASR model aware of lexical information, so the resulting model is expected to achieve better results by leveraging the learned linguistic knowledge. A series of experiments are conducted on the AISHELL-1, CSJ, and TEDLIUM 2 datasets. According to the experiments, the proposed LA-NAT can provide superior results than other recently proposed non-autoregressive ASR models. In addition, LA-NAT is a relatively compact model than most non-autoregressive ASR models, and it is about 58 times faster than the classic autoregressive model.

**Index Terms:** Non-autoregressive, speech recognition, lexical-aware, linguistic knowledge

## 1. Introduction

Automatic speech recognition (ASR) is one of the important research in the context of natural language processing, where the goal is to convert a speech into a corresponding text sequence. In recent years, end-to-end based ASR [1, 2, 3] has created a dominant paradigm, and the school of research can be further divided into autoregressive (AR) and non-autoregressive (NAR) manners according to different decoding philosophies. AR models predict each token by considering all the previous historical information, can be combined with a language model to enhance the performance easily, and usually achieve acceptable performances. However, they often struggle with the decoding speed barrier caused by autoregressive one-by-one decoding. Orthogonal to the AR models, NAR methods assume that all the tokens are conditionally independent, so the decoding results can be generated in parallel at once. In short, the design allows NAR models to deliver faster-decoding speeds in practice, but their recognition performance is usually only comparable to AR models.

In order to boost performance, various research has been devoted to improving NAR ASR. A summary of these models shows two main directions for improvement. On the one hand, some studies concentrate on relaxing the conditional indepen-

dence assumption of the learning objectives from a theoretical perspective [4, 5]. As a result, the improved models are expected to equip the ability to take contextual information into account during decoding. On the other hand, many researchers turn to leverage pre-trained speech and/or language models for building the NAR models. By distilling the knowledge or directly concatenating the pre-trained models to make the resulting ASR models inherit the merits of these pre-trained models [6, 7, 8]. Although classical NAR techniques have progressed with these developments, they usually have a large set of model parameters and are impractical for real-world applications.

Aside from relaxing the conditional independence assumption and using pre-trained models, we propose a framework to enhance NAR ASR by explicitly making the model aware of the lexical information. Accordingly, a lexical-aware non-autoregressive Transformer-based ASR (LA-NAT) model, which consists of an acoustic encoder, a speech-text shared encoder, and a speech-text shared decoder, is introduced. The acoustic encoder is used to process the speech input and produce a set of acoustic features as usual. In order to make the ASR model learn the lexical information naturally, our idea is to construct a shared network for training text and speech data simultaneously. As a result, the model is expected to equip the linguistic knowledge because it must be able to mitigate text-oriented tasks. At the same time, the model is also used for speech recognition, so it would be a lexical-aware ASR model. The speech-text shared encoder and decoder are designed to make the idea work. As the proposed LA-NAT does not cascade large-scaled pre-trained models, its lightweight property is yet another advantage. Extensive experiments are conducted on the AISHELL-1 [9], CSJ [10], and TEDLIUM 2 [11] datasets. The proposed LA-NAT not only demonstrates superior results on these corpora as expected but also reveals the lightweight property compared with other complicated models and achieves about 58 times faster than the classic autoregressive model.

## 2. Related Work

Autoregressive (AR) ASR models aim to generate an output token sequence  $Y = \{y_1, \dots, y_L\}$  for a given speech utterance  $O$  by referring to  $P(Y|O)$ . According to the chain rule, a common simplification strategy is to decompose  $P(Y|O)$  into a series of conditional probabilities, as

$$Y^* = \arg \max_Y P(Y|O) = P(y_1|O) \prod_{l=2}^L P(y_l|y_{<l}, O), \quad (1)$$

where  $y_{<l}$  denotes the partial token sequence before  $y_l$ . Obviously, AR models are designed to generate tokens by considering all previously generated ones and acoustic statistics. Even

though the philosophy promotes these models to take plenty of clues into account for making accurate predictions, there are usually several associated challenges. First, the left-to-right nature of the AR models limits the efficiency of the parallel computation and increases the difficulty of enhancing the inference speed. Second, during training, ground truth history tokens are used to predict the next one, but at the inference stage, only the generated sequence can be used as a condition to predict the upcoming token, which may contain recognition errors. This mismatch problem between training and inference is a common challenge for AR models [12]. A few such representative models are Listen, Attend and Spell (LAS) [1], Self-attention Transducer (SA-T) [13], hybrid CTC/attention architecture [2], Speech-Transformer [14], and Memory equipped Self-attention (SAN-M) [15].

Non-autoregressive (NAR) ASR models assume that each token is conditionally independent of others that are given a speech signal. Formally, for a given speech utterance  $O$ , the NAR modeling decodes the corresponding token sequence  $Y$  by referring to:

$$Y^* = \arg \max_Y P(Y|O) = \prod_{l=1}^L P(y_l|O). \quad (2)$$

Compared with (1), without the dependency between the current and previous tokens, the NAR modeling can generate tokens in parallel to achieve low latency. Besides, the mismatch issue between training and inference no longer occurs. Despite their apparently much simpler designs, more and more NAR-based ASR models demonstrate rapid inference speed and competitive performances with conventional AR methods. In spite of the advantages NAR models seem to have over AR models, a major challenge of NAR modeling is determining the output length. Specifically, during inference, AR modeling stops generating tokens when an end-of-sentence token (e.g.,  $\langle \text{EOF} \rangle$ ) is produced, while the NAR modeling requires additional efforts to post-process the predicted result [3, 16, 17] or to estimate/adjust the length of the recognition sequence [18, 19]. A few of the widely used models include Connectionist Temporal Classification (CTC) [3], Listen and Fill in the Missing Letters (LFML) [16], Listen Attentively, and Spell Once (LASO) [18], Pre-trained Language Model-based NAR ASR (PLM-NAR-ASR) [7] and Mask CTC [17]. Comprehensive comparisons between AR and NAR ASR models are found in the literature [20, 21, 22].

### 3. Proposed Method

The proposed LA-NAT is composed of an acoustic encoder, a speech-text shared encoder, and a speech-text shared decoder. The model architecture is illustrated in Figure 1.

#### 3.1. Acoustic Encoder

Each training example consists of a speech signal  $O$  and its corresponding token sequence  $Y$ , where the speech signal is quantized into a series of  $T'$  acoustic feature frames  $\{o_1, \dots, o_{T'}\}$  and  $Y$  denotes a sequence of  $L$  tokens  $\{y_1, \dots, y_L\}$ . The acoustic encoder uses two layers of the 2D convolutional neural networks to encode temporal and spectral vicinity statistics of acoustic feature frames and downsample the resolution in time. Next, the sinusoidal positional embedding is added for each feature to retain the order in the line [23, 24]. Afterward, we stack a series of  $N_a$  Transformers to reformulate the acoustic charac-

teristics because the self-attention mechanism can blend short-term and long-term information together. Consequently, a set of acoustic features  $H^O = \{h_1^O, \dots, h_{T'}^O\} \in R^{d \times T'}$  is derived. A layer normalization layer, a linear layer, and a softmax activation function are sequentially stacked on  $H^O$ , and a CTC objective  $\mathcal{L}_{\text{CTC}}$  is used to guide the model training toward minimizing the differences between the prediction and the ground-truth.

#### 3.2. Speech-Text Shared Encoder

In order to make the ASR model learn lexical information naturally, a meticulous speech-text shared module, including an encoder and a decoder, is introduced. To be more specific, for each training example, the acoustic encoder converts  $O$  into a set of acoustic features  $H^O$ , while a simple table lookup is applied to translate ground truth text  $Y$  into a series of token embeddings. Without loss of generality, the positional embedding is added for each token embedding to reveal the ordered information in line, and the set of representations is denoted as  $H^Y \in R^{d \times L}$ . A speech-text shared encoder is employed to adjust the two modality features first, and then organize them into a set of representative information. Following the idea, a series of  $N_b$  Transformer layers are applied to adjust acoustic- and text-level features (i.e.,  $H^O$  and  $H^Y$ ), and a layer normalization layer is adopted to stable the distribution of statistics. Accordingly, two set of features  $H_{se}^O \in R^{d \times T'}$  and  $H_{se}^Y \in R^{d \times L}$  are obtained for speech and text inputs, respectively.

Next, inspired by the memory network [25], we initialize  $M$  memory slots to act as a set of anchors to rearrange and reorganize speech and text information. Concretely, the memory slots and a set of features (i.e.,  $H_{se}^O$  or  $H_{se}^Y$ ) are fed into a Transformer, where the former is used to query the latter. Subsequently,  $N_c$  Transformers, where the queries are output from the previous Transformer, while the keys and values are identical to the original set of features (i.e.,  $H_{se}^O$  or  $H_{se}^Y$ ), are used to iteratively rearrange and reorganize the statistics to obtain a set of acoustic-based composite representations  $H_{com}^O = \{h_{com_1}^O, \dots, h_{com_M}^O\} \in R^{d \times M}$  and a set of lexical-based composite representations  $H_{com}^Y = \{h_{com_1}^Y, \dots, h_{com_M}^Y\} \in R^{d \times M}$ . Since the content of speech and text input is the same, the resulting acoustic-based and lexical-based composite representations should be able to align one-to-one. Therefore, a contrastive objective  $\mathcal{L}_{\text{Cont}}$  is used to maximize the matching degree between each pair of composite representations:

$$\begin{aligned} \mathcal{L}_{\text{Cont}} = & - \sum_{i=1}^M \log \frac{e^{\cos(H_{com_i}^O, H_{com_i}^Y)/\tau}}{\sum_{j=1}^M e^{\cos(H_{com_i}^O, H_{com_j}^Y)/\tau}} \\ & - \sum_{j=1}^M \log \frac{e^{\cos(H_{com_j}^O, H_{com_j}^Y)/\tau}}{\sum_{i=1}^M e^{\cos(H_{com_i}^O, H_{com_j}^Y)/\tau}}, \end{aligned} \quad (3)$$

where  $\cos(\cdot, \cdot)$  denotes the cosine similarity function, and  $\tau$  is a scaling factor. It is worth noting that memory slots are model parameters that should be updated by training. Besides, the memory network-like procedure can not only deal with the length mismatch between acoustic- and text-level features (i.e.,  $H_{se}^O$  and  $H_{se}^Y$ ) but also work as a set of pivots to align and capsule information from the two modalities of features.

#### 3.3. Speech-Text Shared Decoder

After the speech-text shared encoder, a speech-text shared decoder is performed at the last step. The decoder is employed to do speech recognition with acoustic-based representations,

while it also has to reconstruct the original token sequence with lexical-based representations. In detail, for the former, we initialize a set of positional embeddings  $H_{PE} \in R^{d \times L}$  as queries to retrieve information from acoustic representations  $[H_{se}^O; H_{com}^O] \in R^{d \times (T+M)}$  by using  $N_d$  Transformer layers. Except for the first one Transformer, the queries are output from the previous Transformer, and the keys and values are always identical to the concatenated acoustic features  $[H_{se}^O; H_{com}^O]$ . It should be emphasized that we use a fusion mask to constrain the attention scope in each Transformer [26]. The Viterbi algorithm is used to match the CTC alignment with the ground truth into a trigger mask as the left half of the fusion mask, and elements in the right-hand side of the fusion mask are all set to one. By doing so, each query can capture local information from features produced by the speech-text shared encoder  $H_{se}^O$  and global information from acoustic-based composite features  $H_{com}^O$  by the cross-attention mechanism. In other words,  $H_{se}^O$  is a set of low-level fine-grained acoustic features, while  $H_{com}^O$  can be regarded as a set of coarse-grained auxiliary information. After that, the resulting features are passed through a sequence of  $N_e$  Transformers to refine the statistics. A linear layer and a softmax activation function are sequentially stacked to construct an ASR model. Consequently, the cross-entropy loss  $\mathcal{L}_{CE}^O$  is used to guide the model training toward minimizing the differences between the prediction and the ground-truth. During training, the teacher forcing strategy is used in our implementation, so the number of positional embeddings (i.e.,  $L$ ) is determined by referring to the ground truth.

Despite the speech part, the speech-text shared decoder is also designed to reconstruct the original token sequence by feeding lexical representations. Again, a set of positional embeddings is created to query the lexical-based statistics  $[H_{se}^Y; H_{com}^Y]$  and reorganize them with the corresponding attention scores by cross-attention mechanism. Except for the first Transformer, the queries are generated by the previous Transformer, and the keys and values are always identical to the concatenated features  $[H_{se}^Y; H_{com}^Y]$ . Following, the resulting features are passed through a set of Transformers to manipulate the statistics, and then a linear layer and a softmax activation function are sequentially stacked. Finally, the cross-entropy loss  $\mathcal{L}_{CE}^Y$  is used to guide the model training toward minimizing the reconstruction errors.

During inference, we use the CTC branch to calculate the fusion mask and decide the length of the recognition result. It is worth noting that text input is only used for training, and the proposed ASR model is performed in a non-autoregressive manner. In a nutshell, the proposed LA-NAT introduces a speech-text shared encoder and a speech-text shared decoder to train speech and text input simultaneously while with different tasks. Thereby, LA-NAT can learn linguistic knowledge from the text data so as to enhance the performance of ASR.

### 3.4. Training Method

Since the proposed LA-NAT has to face two modality features, the training procedure must be carefully designed to stable the model training and obtain a better model.

**Step1:** First of all, we collect a large set of text data to train the speech-text shared encoder and decoder with the reconstruction task (i.e., the objective function is  $\mathcal{L}_{CE}^Y$ ) only. By doing so, linguistic knowledge can be learned and stored in the speech-text shared encoder and decoder models.

**Step2:** Following, we optimize all the model parameters, except the set of word embeddings used for translating text in-

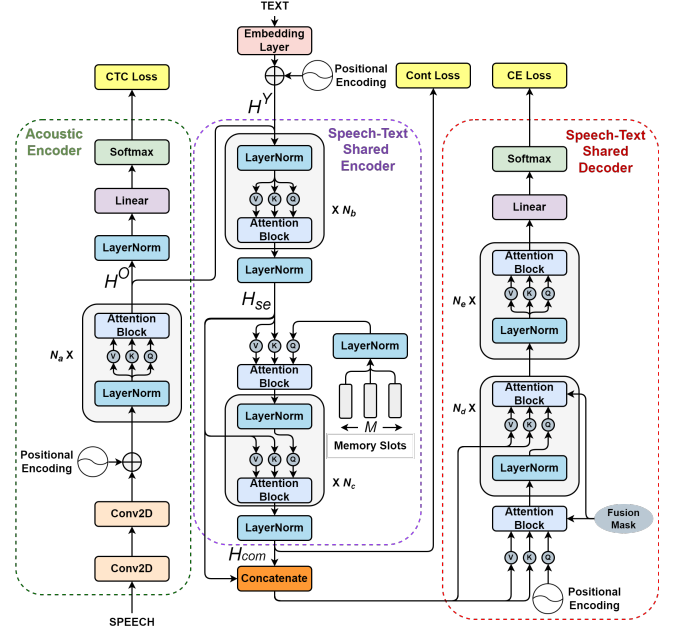


Figure 1: The model architecture of the proposed LA-NAT.

put to a sequence of token embeddings, by using speech data only toward minimizing  $\lambda_{CTC} \mathcal{L}_{CTC} + \lambda_{CE}^O \mathcal{L}_{CE}^O$  (cf. Sections 2.1 and 2.3). The hyper-parameters  $\lambda_{CTC}$  and  $\lambda_{CE}^O$  are set to 0.3 and 1.0, respectively. On top of the first stage, the speech-text shared encoder and decoder have been pre-trained with text data, so we expect the ASR model can be more robust base on the learned linguistic knowledge.

**Step3:** In the last step, both speech and text data are used to update the entire model. The training objective is to minimize all the loss functions  $\lambda_{CTC} \mathcal{L}_{CTC} + \lambda_{Cont} \mathcal{L}_{Cont} + \lambda_{CE}^Y \mathcal{L}_{CE}^Y + \lambda_{CE}^O \mathcal{L}_{CE}^O$ . The weighting factors  $\lambda_{CTC}$ ,  $\lambda_{Cont}$ ,  $\lambda_{CE}^Y$ , and  $\lambda_{CE}^O$  are set to 0.3, 1.0, 0.3, and 1.0, respectively.

It's interesting to note that the text data used in the first step can come from either the speech corpus, another selected collection, or a combination of them. Since the text data is easy to collect, the proposed LA-NAT can thus enjoy the advantage of inheriting linguistic knowledge from a large set of text data and becoming a lexical-aware Transformer-based NAR ASR model.

## 4. Experiments

### 4.1. Experimental setup

The experiments are conducted on three datasets, including AISHELL-1, CSJ, and TEDLIUM 2. For pre-training, we use the AISHELL-2 [34], the training set in CSJ, and LibriSpeech [35] to initialize the speech-text shared modules. The lexicon sizes are 4, 231, 3, 260, and 5, 000 for Chinese, Japanese, and English, respectively. We use 80-dimensional log Mel-filter bank features with 3-dimensional pitch features, computed every 10ms with a window size of 25ms, and the SpecAugment is applied for data augmentation. Speed perturbation is also used for AISHELL-1 and TEDLIUM 2. The hyper-parameters  $\{N_a, N_b, N_c, N_d, N_e\}$  are set to  $\{6, 6, 5, 2, 4\}$  for AISHELL-1, while  $N_a$  is set to 12 for CSJ and TEDLIUM 2. We build a base and a large models with hidden layer sizes  $d$  of 256 and 512, respectively. Each CNN layer in the acoustic encoder has 256 or 512 filters, depending on whether it is a base or a large

Table 1: Experimental results on AISHELL-1 (CER%), CSJ (CER%) and TEDLIUM 2 (WER%), respectively. The model parameters and real-time factors (RTF) are calculated by using the AISHELL-1 dataset.

MODEL		AISHELL-1			CSJ		TEDLIUM 2		#Param.	RTF
		Dev	Test	Eval1	Eval2	Eval3	Dev	Test		
AR	LAS [1]	-	8.7	-	-	-	-	-	156M	-
	SA-T [13]	8.3	9.3	-	-	-	-	-	-	-
	ESPnet (Transformer) [27]	6.0	6.7	6.37	4.76	5.4	12.6	10.2	58M	0.362
	SAN-M [15]	5.7	6.5	-	-	-	-	-	-	-
	CAT [28]	-	6.3	-	-	-	-	-	-	-
NAR	ST-NAT [29]	6.4	7.0	-	-	-	-	-	31M	-
	Mask CTC [17]	6.0	6.7	6.56	4.57	4.96	11.9	10.7	29.7M	-
	LASO- Big [18]	5.8	6.4	-	-	-	-	-	105.8M	0.0042
	CTC [17, 30]	5.7	6.2	6.51	4.71	5.49	12.8	12.2	-	-
	NAR-Transformer [31]	5.3	5.9	-	-	-	-	-	29.7M	-
	CASS-NAT [26]	5.3	5.8	-	-	-	-	-	33.2M	-
	HANAT-Big [32]	5.1	5.6	-	-	-	-	-	87M	-
	Improved CASS-NAT [33]	4.9	5.4	-	-	-	-	-	38.3M	-
	<b>LA-NAT Base</b>	<b>4.58</b>	<b>4.96</b>	<b>5.91</b>	<b>4.11</b>	<b>5.05</b>	<b>11.6</b>	<b>11</b>	31.5M	0.0062
	<b>LA-NAT Large</b>	<b>4.51</b>	<b>4.88</b>	<b>5.55</b>	<b>3.73</b>	<b>4.5</b>	<b>10.9</b>	<b>10.6</b>	79.8M	0.0067

Table 2: Ablation study on AISHELL-1 dataset with only SpecAugment.

Training Recipe	Dev	Test
From scratch	5.21	5.83
Step 3	5.35	5.74
Steps 1 and 3	5.22	5.62
Steps 1, 2 and 3	4.84	5.33

model, with a kernel size of 3 and a stride of 2. The intermediate dimension of the feedforward neural network is 2,048 with 8 attention heads, and the activation function is GLU. During the schedule learning, the model is trained with 40 epochs in the first stage and 130 epochs in the next two stages, and we average the model parameters of the last 10 epochs as the final model. Each training batch consists of 100 seconds of speech and accumulates gradients of 12 steps.

## 4.2. Experimental results

Table 1 summarizes the experimental results of the proposed LA-NAT and several SOTA models. Information on the real-time factors (RTF) and the model sizes are also presented in Table 1. The RTF results are measured for inference on the test set of the AISHELL-1 dataset using an NVIDIA RTX 3090 with a batch size of 1. Several observations can be drawn from the results. At first glance, the results reveal that the NAR models have caught up to the efficiency and effectiveness of the AR ones, thereby demonstrating the potential of NAR ASR modeling. Second, the proposed LA-NAT can deliver better results and provide a faster decoding speed than the AR models. Taking LA-NAT Base and ESPnet (Transformer) as an example, regardless of the improvement in recognition performance, the former is not only nearly half the size of the latter, but the decoding speed is at least 58 times faster. Third, compared with NAR models, LA-NAT not only can give satisfactory results but also is a relatively exquisite model. Although LA-NAT only obtains competitive results on the TEDLIUM 2 dataset, we have proved the proposed LA-NAT can be done in both logographic (i.e., Chinese and Japanese) and alphabetic (i.e., English) languages. To further boost the performance, one of the emergent future works is to search for better configurations for alphabetic languages. A possible way is to leverage enhanced CTC-based objectives for training the acoustic encoder [4, 5, 30]. Finally, it should be mentioned that the performance gap between the base

and large models of the proposed LA-NAT is remarkable on the CSJ and TEDLIUM 2 corpora, but there is only a dearth of improvements on the AISHELL-1 dataset. A possible reason may be that both the CSJ and TEDLIUM 2 are more complicated in content than the AISHELL-1. In this case, scaling up the model size provides a straightforward way of increasing the capability and ability of the proposed LA-NAT.

## 4.3. Ablation Studies and Analysis

To analyze the impact of lexical information in the proposed LA-NAT, we examine the results step by step. The experimental results are shown in Table 2. ‘‘From scratch’’ denotes LA-NAT is directly trained using a speech dataset without extra text corpus, pre-training, and text input. In other words, only the CTC loss  $\mathcal{L}_{CTC}$  and cross-entropy loss for ASR  $\mathcal{L}_{CE}^O$  are used to update the model parameters. It is thus a naïve baseline of LA-NAT. From the results, it is easy to conclude that linguistic knowledge can indeed help the ASR model. Furthermore, the well-designed training procedure can improve the effectiveness and efficiency of LA-NAT in learning and leveraging linguistic knowledge. The best setting (i.e., ‘‘Steps 1, 2 and 3’’) can deliver up to 8.6% relative CER reduction on the test set than the naïve setting (i.e., ‘‘From scratch’’). In sum, the set of experiments stresses the utilities of the lexical information, the potential of the proposed LA-NAT, and the importance of the presented training procedure.

## 5. Conclusions

In this paper, we propose a novel lexical-aware non-autoregressive Transformer-based ASR framework, which indicates a potential way to enhance the performance of NAR ASR. The experimental results show that LA-NAT achieves competitive or SOTA results compared to other NAR models and is about 58 times faster than the classic AR model. In the future, we will continue improving the model architectures and exploring different training objectives and effective methods for modeling lexical-aware non-autoregressive ASR.

## 6. Acknowledgement

We thank the National Center for High-performance Computing of the National Applied Research Laboratories (NARLabs) in Taiwan for providing computational and storage resources.

## 7. References

- [1] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.
- [2] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.
- [3] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," vol. 2006, 01 2006, pp. 369–376.
- [4] B. Naowarat, T. Kongthaworn, K. Karunratanakul, S. H. Wu, and E. Chuangsuwanich, "Reducing spelling inconsistencies in code-switching asr using contextualized ctc loss," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6239–6243.
- [5] K.-H. Lu and K.-Y. Chen, "A context-aware knowledge transferring strategy for ctc-based asr," *arXiv preprint arXiv:2210.06244*, 2022.
- [6] K. Deng, S. Cao, Y. Zhang, L. Ma, G. Cheng, J. Xu, and P. Zhang, "Improving ctc-based speech recognition via knowledge transferring from pre-trained language models," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8517–8521.
- [7] F.-H. Yu, K.-Y. Chen, and K.-H. Lu, "Non-autoregressive asr modeling using pre-trained language models for chinese speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1474–1482, 2022.
- [8] K. Deng, Z. Yang, S. Watanabe, Y. Higuchi, G. Cheng, and P. Zhang, "Improving non-autoregressive end-to-end speech recognition with pre-trained acoustic and language models," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8522–8526.
- [9] 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.
- [10] K. Maekawa, "Corpus of spontaneous japanese: Its design and evaluation," in *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.
- [11] A. Rousseau, P. Deléglise, Y. Esteve *et al.*, "Enhancing the tedlium corpus with selected data for language modeling and more ted talks," in *LREC*, 2014, pp. 3935–3939.
- [12] S. Bengio, O. Vinyals, N. Jaitly, and N. M. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," *ArXiv*, vol. abs/1506.03099, 2015.
- [13] Z. Tian, J. Yi, J. Tao, Y. Bai, and Z. Wen, "Self-attention transducers for end-to-end speech recognition," in *Interspeech*, 2019.
- [14] L. Dong, S. Xu, and B. Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.
- [15] Z. Gao, S. Zhang, M. Lei, and I. McLoughlin, "San-m: Memory equipped self-attention for end-to-end speech recognition," in *Interspeech*, 2020.
- [16] N. Chen, S. Watanabe, J. Villalba, and N. Dehak, "Listen and fill in the missing letters: Non-autoregressive transformer for speech recognition," *ArXiv*, vol. abs/1911.04908, 2019.
- [17] 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.
- [18] Y. Bai, J. Yi, J. Tao, Z. Tian, Z. Wen, and S. Zhang, "Fast end-to-end speech recognition via non-autoregressive models and cross-modal knowledge transferring from bert," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1897–1911, 2021.
- [19] Y. Higuchi, H. Inaguma, S. Watanabe, T. Ogawa, and T. Kobayashi, "Improved mask-ctc for non-autoregressive end-to-end asr," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8363–8367, 2020.
- [20] X. Song, Z. Wu, Y. Huang, C. Weng, D. Su, and H. M. Meng, "Non-autoregressive transformer asr with ctc-enhanced decoder input," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5894–5898, 2020.
- [21] Y. Higuchi, N. Chen, Y. Fujita, H. Inaguma, T. Komatsu, J. Lee, J. Nozaki, T. Wang, and S. Watanabe, "A comparative study on non-autoregressive modelings for speech-to-text generation," *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 47–54, 2021.
- [22] E. G. Ng, C.-C. Chiu, Y. Zhang, and W. Chan, "Pushing the limits of non-autoregressive speech recognition," in *Interspeech*, 2021.
- [23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [25] Q. Wu, Z. Lan, K. Qian, J. Gu, A. Geramifard, and Z. Yu, "Memformer: A memory-augmented transformer for sequence modeling," *arXiv preprint arXiv:2010.06891v2*, 2022.
- [26] R. Fan, W. Chu, P. Chang, and J. Xiao, "Cass-nat: Ctc alignment-based single step non-autoregressive transformer for speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5889–5893.
- [27] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Yalta, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," in *Interspeech*, 2018.
- [28] K. An, H. Xiang, and Z. Ou, "Cat: A ctc-crf based asr toolkit bridging the hybrid and the end-to-end approaches towards data efficiency and low latency," in *Interspeech*, 2020.
- [29] Z. Tian, J. Yi, J. Tao, Y. Bai, S. Zhang, and Z. Wen, "Spike-triggered non-autoregressive transformer for end-to-end speech recognition," *arXiv preprint arXiv:2005.07903*, 2020.
- [30] J. Nozaki and T. Komatsu, "Relaxing the conditional independence assumption of ctc-based asr by conditioning on intermediate predictions," *arXiv preprint arXiv:2104.02724*, 2021.
- [31] X. Song, Z. Wu, Y. Huang, C. Weng, D. Su, and H. Meng, "Non-autoregressive transformer asr with ctc-enhanced decoder input," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5894–5898.
- [32] Z. Tian, J. Yi, J. Tao, S. Zhang, and Z. Wen, "Hybrid autoregressive and non-autoregressive transformer models for speech recognition," *IEEE Signal Processing Letters*, vol. 29, pp. 762–766, 2022.
- [33] R. Fan, W. Chu, P. Chang, J. Xiao, and A. Alwan, "An improved single step non-autoregressive transformer for automatic speech recognition," in *Interspeech*, 2021.
- [34] J. Du, X. Na, X. Liu, and H. Bu, "Aishell-2: Transforming mandarin asr research into industrial scale," *arXiv preprint arXiv:1808.10583*, 2018.
- [35] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.