

MINIMUM WORD ERROR TRAINING FOR NON-AUTOREGRESSIVE TRANSFORMER-BASED CODE-SWITCHING ASR

Yizhou Peng^{1*}, Jicheng Zhang^{1*}, Haihua Xu^{2*}, Hao Huang^{1,4}, Eng Siong Chng³

¹School of Information Science and Engineering, Xinjiang University, China

²Bytedance AI Lab, Singapore

³School of Computer Science and Engineering, Nanyang Technological University, Singapore

⁴Xinjiang Provincial Key Laboratory of Multi-lingual Information Technology, Urumqi, China

ABSTRACT

Non-autoregressive end-to-end ASR framework might be potentially appropriate for code-switching recognition task thanks to its inherent property that present output token being independent of historical ones. However, it still under-performs the state-of-the-art autoregressive ASR frameworks. In this paper, we propose various approaches to boosting the performance of a CTC-mask-based non-autoregressive Transformer under code-switching ASR scenario. To begin with, we attempt diversified masking method that are closely related with code-switching point, yielding an improved baseline model. More importantly, we employ Minimum Word Error (MWE) criterion to train the model. One of the challenges is how to generate a diversified hypothetical space, so as to obtain the average loss for a given ground truth. To address such a challenge, we explore different approaches to yielding desired N-best-based hypothetical space. We demonstrate the efficacy of the proposed methods on SEAME corpus, a challenging English-Mandarin code-switching corpus for Southeast Asia community. Compared with the cross-entropy-trained strong baseline, the proposed MWE training method achieves consistent performance improvement on the test sets.

Index Terms— Non-autoregressive, Minimum-Word-Error, Code-switching, Transformer, ASR

1. INTRODUCTION

End-to-end (E2E) ASR has become predominant in both research and industrial areas thanks to its modeling simplicity, compactness, as well as efficacy in performance in recent years. The state-of-the-art (SOTA) E2E ASR modeling frameworks include frame-synchronous family, such as Connectionist Temporal Classification (CTC) [1, 2], Recurrent Neural Network Transducer (RNN-T) [3, 4], and lab-synchronous or attention-based encoder-decoder (AED) family, such as Transformer [5, 6, 7], and Conformer [8] etc. Except for the CTC method, all these belong to auto-regressive modeling framework, that is, the present output token of the decoder is dependent on previously predicted tokens.

The autoregressive method has one obvious advantage that it can simultaneously learn acoustic and language semantics jointly, yielding SOTA results. However, one of the limitations of the autoregressive method is that it is weak in recognition of unseen utterances. Take code-switching ASR as an example, we cannot let

an autoregressive-based ASR system that is learned with two mixed monolingual data recognize code-switching words in a given utterance [9].

More recently, non-autoregressive Transformer (NAT) modeling framework has drawn growing attention in machine translation [10, 11], as well as speech recognition [12, 13, 14, 15] societies due to its output token from the decoder can be independent to each other, resulting in faster decoding while comparable results to what is brought by the autoregressive Transformer (AT). However, the performance of the NAT method is still worse than that of its AT counterparts.

In this paper, we attempt to employ NAT for code-switching speech recognition. We are motivated by that the output token independence property of the NAT method could be potentially more appropriate for code-switching ASR given more monolingual data available. However, our first priority here is still to make the NAT performance approach that of the corresponding AT method. Consequently, we are meant to boost the SOTA non-autoregressive performance under code-switching ASR scenario.

Employing the CTC-mask-based NAT, our efforts lie in the following aspects. 1) Expedite the training process by an enforced Conditional Masked Language Model (CMLM) learning. 2) We employ different masking methods that aim at code-switching points in utterance for training, yielding improved baseline results. 3) Furthermore, we propose to employ Minimum Word Error (MWE) criterion to train our NAT model, yielding consistent performance improvement. To the best of knowledge, we are the first to introduce MWE criterion to train the NAT model for speech recognition task.

2. RELATED WORKS

Recently NAT has been widely studied in ASR area due to its much faster speed in inference while it can achieve comparable recognition results, in contrast to its AT counterpart. Though with the same NAT framework, people employ it to do the ASR work differently. One school of branch advocates using one-pass inference method, that is, once the embedding of the encoder representation for a given utterance is ready, one can let the decoder output the overall inference sequence with one-pass simultaneously [16]. Another school of branch attempts to output the inference results with iterative fashion, where the decoder input is a sequence of which some of tokens are randomly masked, and the inference process is the process of predicting those masked tokens [17, 18, 19]. In whatever approach is taken, the length of the token sequence that is fed into the decoder is essential for the inference performance. In this paper, we take the iterative inference approach, following what is proposed in [17, 20].

* Authors have equal contributions. Hao Huang is the correspondence author. This work was supported by the National Key R&D Program of China (2020AAA0107902); Opening Project of Key Laboratory of Xinjiang, China (2020D04047); NSFC (61663044, 61761041).

Our efforts are focused on extending the prior work under the code-switching speech recognition environment. More importantly, we attempt to employ MWE criterion to train the NAT network, yielding consistent improvement.

3. SYSTEM FRAMEWORK

Theoretically, given an input sequence $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ for both AT and NAT, the objective is to predict the conditional probability $P(y_i|\mathbf{X}; \theta)$ of each token y_i , where $y_i \in \mathbf{Y} = \{y_1, \dots, y_N\}$. The objective of AT and NAT ASR is the same, but it is accomplished differently.

Technically, the whole NAT network is made up of 3 sub-networks, i.e., Transformer encoder, CTC network on top of the Transformer encoder, and a self-attention based Transformer decoder. For AT model, CTC is optional though it can benefit the AT's performance [21, 22]. For NAT model, CTC is critical in both one-pass and iterative decoding fashions [16, 17, 18, 19], except that [23] directly employs a fixed length position encoded sequence as input to the decoder, producing sub-optimal results. In our iterative NAT method, CTC is employed to generate preliminary inference sequence by greedy method, besides, it is shown in [17], CTC training is essential to producing a desired performance.

For clarification to what follows, we start with revisiting the principle of working mechanism for AT, then CTC, and finally our Conditional Mask Language Model (CMLM)-based [24] NAT in this section.

3.1. Autoregressive Transformer

For Transformer, the joint probability is estimated as follows:

$$P_{\text{AT}}(\mathbf{Y}|\mathbf{X}) = \prod_{i=1}^N P(y_i|\mathbf{y}_{<i}, \mathbf{X}) \quad (1)$$

where $\mathbf{y}_{<i}$ refers to the previous tokens before i . During training, estimating $P(y_i|\mathbf{y}_{<i}, \mathbf{X}; \theta)$ can proceed with parallelism since $\mathbf{y}_{<i}$ is directly from ground truth, while the inference process has to proceeds step-by-step, as present output token y_i depends on $\mathbf{y}_{<i}$.

3.2. Connectionist Temporal Classification

For CTC, the joint probability is estimated as follows:

$$P(\mathbf{Y}'|\mathbf{X}) = \prod_{t=1}^T P(y'_t|\mathbf{X}) \quad (2)$$

where $y'_t \in \mathbf{Y}'$ is a CTC frame-synchronous token that is either a normal token as in Section 3.1, or a blank token “-”, indicating nothing being output. For training, the objective function is

$$P_{\text{CTC}}(\mathbf{Y}|\mathbf{X}) = \sum_{\mathcal{T}^{-1}(\mathbf{Y}')=\mathbf{Y}} \mathbf{P}(\mathbf{Y}'|\mathbf{X}) \quad (3)$$

where function \mathcal{T}^{-1} removes all duplicated and blank tokens. For inference, CTC can choose either beam search based forward-search algorithm, or greedy algorithm. The forward-search algorithm is governed by the following formulae:

$$\mathbf{Y}^* = \underset{\mathcal{T}^{-1}(\mathbf{Y}')=\mathbf{Y}}{\operatorname{argmax}} P(\mathbf{Y}'|\mathbf{X}) \quad (4)$$

Normally, the forward-search algorithm is time consuming. In this paper, to preserve the NAT faster inference property, we employ the greedy algorithm as follows instead:

$$\mathbf{Y}'^* = \operatorname{Concat} \underset{1 \leq t \leq T}{\operatorname{argmax}} P(y'_t|\mathbf{X}) \quad (5)$$

$$\mathbf{Y}^*_{\text{greedy}} = \mathcal{T}^{-1}(\mathbf{Y}'^*) \quad (6)$$

though the greedy algorithm yields much worse results, they will be refined by the NAT decoder iteratively.

3.3. CMLM non-autoregressive ASR

As observed from Equation (2), we know CTC belongs to non-autoregressive modeling framework by definition. If we use the greedy method (see Equations (5) and (6)) to perform inference, CTC is completely a non-autoregressive ASR method, since each output is independent to the remaining ones. However, it yields much worse ASR performance. In this paper, inspired from AT method, as well as pursuing faster inference speed from NAT method, we employ a CMLM NAT ASR framework [17] as follows:

$$P_{\text{CMLM}}(\mathbf{Y}_{\text{mask}}|\mathbf{Y}_{\text{obs}}, \mathbf{X}) = \prod_{y \in \mathbf{Y}_{\text{mask}}} P(y|\mathbf{Y}_{\text{obs}}, \mathbf{X}) \quad (7)$$

where $\mathbf{Y}_{\text{obs}} = \mathbf{Y} \setminus \mathbf{Y}_{\text{mask}}$. That is, the CMLM NAT method attempts to predict mask tokens \mathbf{Y}_{mask} , conditioning on unmasked/observed tokens \mathbf{Y}_{obs} , given an acoustic input sequence \mathbf{X} from encoder.

During training, the ground truth tokens are randomly masked, and they are replaced with a predefined mask symbol $\langle \text{MASK} \rangle$. As a result, the NAT decoder is learned as a CMLM. During inference, the output of CTC using Equation (6) takes the place of the ground truth, but some of the tokens, which have lower posteriors, are similarly masked. An iterative inference method is employed to yield the final results.

In [17], it is observed that using only criterion in Equation (7) to train the NAT model results in worse performance, and a joint CTC and CMLM criterion is employed instead as follows:

$$\mathcal{L}_{\text{NAT}} = \alpha P_{\text{CTC}}(\mathbf{Y}|\mathbf{X}) + (1 - \alpha) P_{\text{CMLM}}(\mathbf{Y}_{\text{mask}}|\mathbf{Y}_{\text{obs}}, \mathbf{X}) \quad (8)$$

where α is a hyperparameter fixed with 0.3 for all experiments in what follows.

4. PROPOSED METHODS

4.1. Enforced CMLM learning for training expedition

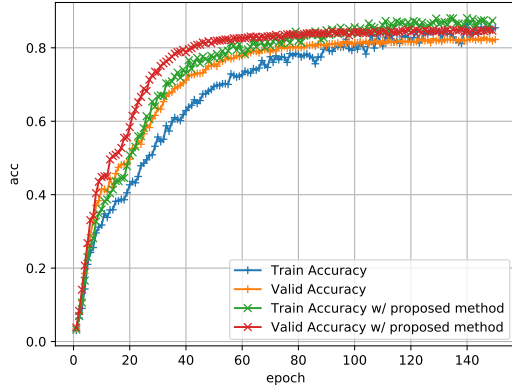
In our training practice with criterion (8), we found the training is much slower to converge compared with the normal AT training. We guess the loss from the term $P_{\text{CMLM}}(\mathbf{Y}_{\text{mask}}|\mathbf{Y}_{\text{obs}}, \mathbf{X})$ cannot make the network sufficiently trained. Therefore, we propose to add another term to make it faster to converge. Specifically, we not only employ the observed sequence to predict masked sequence, we also do the opposite prediction, that is, let the masked tokens unmasked while the observed tokens masked. Consequently, the criterion (8) is changed to

$$\mathcal{L}'_{\text{NAT}} = \alpha P_{\text{CTC}}(\mathbf{Y}|\mathbf{X}) + (1 - \alpha) [P_{\text{CMLM}}(\mathbf{Y}_{\text{mask}}|\mathbf{Y}_{\text{obs}}, \mathbf{X}) + P_{\text{CMLM}}(\mathbf{Y}_{\text{obs}}|\mathbf{Y}_{\text{mask}}, \mathbf{X})] \quad (9)$$

Using the updated objective function (9), we find that it is faster to converge for training. Figure 1 illustrates the accuracy curve on both

training and validation data set using the proposed training method in contrast to the baseline. From Figure 1, we observe faster training convergence, as well as better accuracy on both training and validation data while using the proposed training method. For convenience, we name the original mask method as R-Mask method, while the proposed one as C-Mask method.

Fig. 1. Accuracy trend on both training and validation data sets, using the proposed enforced CMLM learning criterion.



4.2. Code-switching-based masking methods

Code-switching refers to a multilingual phenomenon of which speakers spontaneously switch between two or more different languages, and such switching can occur within an utterance, denoted as CS_1 , or between consecutive utterances, denoted as CS_2 . Technically, CS_1 is much more challenging than CS_2 , since cross-lingual word switching normally has rare samples, resulting in ASR models insufficiently learned and weak in performing code-switching accurately. In what follows, we are dealing with CS_1 code-switching problem.

As the CMLM-based NAT ASR has a decoder that acts as a conditional masked language model, we can deliberately mask those words located at code-switching points. Alternatively, we can also mask one language or another language. In either way, we are meant to enforce the CMLM to sufficiently learn code-switching predictions.

Specifically, given an English-Mandarin code-switching utterance $\mathbf{Y} = \{y_1, \dots, y_N\}$, we have two kinds of code-switching pairs, that is, $\text{Pair}_1 = (y_{CN}, y_{EN})$ or $\text{Pair}_2 = (y_{EN}, y_{CN})$. We are doing the following 4 types of masking method:

- We mask the first token of those code-switching pairs, denoted as F-Mask.
- We mask the second token of the code-switching pairs, denoted as S-Mask.
- In consideration of language, we only mask the Mandarin tokens, denoted as M-Mask.
- Alternatively, we only mask the English tokens, denoted as E-Mask.

4.3. Minimum word error training

So far, CMLM network training criteria from Equations (8) or (9) belongs to the Cross Entropy (CE) category, i.e., computing the log likelihood of each ground truth posterior during training. However, ASR performance metric is defined by calculating Word Error

Rate (WER), an edit distance between the recognized sequence and ground truth, for each recognized utterance, that is, $\mathcal{D}(\mathbf{Y}, \mathbf{Y}^*)$, where \mathbf{Y}^* and \mathbf{Y} are ground truth and recognized sequences respectively. Therefore, a matched training criterion for any ASR system should be reducing the expected Word Error Rate, $\mathbb{E}[\mathcal{D}(\mathbf{Y}, \mathbf{Y}^*)]$, on training data. From this perspective, the conventional CE training yields sub-optimal performance. In this section, we are exploring sequential-based Minimum Word Error (MWE) [25, 26, 27] training under NAT framework. For simplicity, we employ N-best hypothesis list to approximate our hypothetical space. The training criterion is as follows:

$$\mathcal{L}_{\text{MWE}} = \mathbb{E}[\mathcal{D}(\mathbf{Y}, \mathbf{Y}^*)] = \sum_{i=1}^N P(\mathbf{Y}_i | \mathbf{X}) \mathcal{D}(\mathbf{Y}_i, \mathbf{Y}^*) \quad (10)$$

where N is the number of hypothesis in the N-best list, $P(\mathbf{Y}_i | \mathbf{X})$ is the utterance posterior, calculated using Equation (2) under NAT framework. In practice, to compute the MWE loss, we normalize both $P(\mathbf{Y}_i | \mathbf{X})$ and $\mathcal{D}(\mathbf{Y}_i, \mathbf{Y}^*)$ in Equation (10) as advocated in [25]. Besides, the final training criterion is an interpolation of CE and MWE as:

$$\mathcal{L}'_{\text{MWE}} = \gamma \mathcal{L}_{\text{NAT}} + (1 - \gamma) \mathcal{L}_{\text{MWE}} \quad (11)$$

where γ is a tunnable parameter and it is fixed with 0.01.

The problem of using criterion (11) for training is how to generate N-best list under our NAT framework, which is quite simple under autoregressive framework since training proceeds with token-by-token fashion, and each token can have multiple choices under predefined beams. A simplified recipe in our case is to generate Nbest restricted to \mathbf{Y}_{mask} subspace in Equation (8), denoted as Output-Nbest. Alternatively, when we get output from CTC using Equation (6), we can employ it to generate N randomly masked sequences, yielding Nbest hypothesis. To differentiate, we call such an Nbest list generation method as Input-Nbest. In this paper, we employ both methods to test the efficacy of the MWE training criterion.

5. EXPERIMENTS

5.1. Data

We conduct our experiments on SEAME data, which is a southeast English-Mandarin code-switching corpus, amounted to ~ 110 hours in total. The data is collected from both Singapore and Malaysia areas. It is spontaneous, recorded under clean conversational environment. During conversing, one speaker is an interviewer, responsible for asking questions, and the questions are covered with diversified topics. The other speaker is an interviewee to answer the questions. SEAME corpus only contains interviewee's voice, and there are 154 speakers in total. One can refer to [28] for more details.

Table 1. Main statistics for data Train, Valid, eval_{man}, and eval_{sge} respectively.

	Train	Valid	eval _{man}	eval _{sge}
Duration (Hrs)	93.6	7.0	7.5	3.9
Ave. Length (Sec)	3.9	4.0	4.1	2.7
#speaker	134	134	10	10

To evaluate ASR system's performance, we define 2 test sets, each contains the speech of 10 speakers' entire conversation. Each

speaker is either from Singapore or from Malaysia area. The 2 data sets are named as $eval_{man}$ and $eval_{sg}$ respectively. $eval_{man}$ is dominated with Mandarin, the English to Mandarin ratio is around 3:7 in terms to token counting. Here, token refers to English word and Mandarin character. By contrast, $eval_{sg}$ is dominated with English, and corresponding English to Mandarin ratio is around 7:3. Table 1 reports the overall data distribution for experiments.

Table 2. MER (%) for the proposed NAT methods in contrast with AT methods. The SA stands for Spec-Augmentation. The R-Mask is for Random Mask. (EMFS)-Masks are code-switching-based English Mask, Mandarin Mask, the First Mask and the Second Mask respectively, refer to Section 4.2 for the details. For both Input-Nbest and Output-Nbest, refer to Section 4.3 for details.

Modeling method & Setup		$eval_{man}$	$eval_{sg}$	RTF
AT+CE	Transformer	21.74	29.46	0.68
	+ SA	17.89	25.1	0.70
NAT + CE	R-Mask [17]	20.28	28.33	0.05
	C-Mask	20.33	27.87	0.05
	E-Mask	20.11	27.73	0.05
	M-Mask	20.04	28.11	0.05
	S-Mask	20.17	27.89	0.05
	F-Mask	20.07	27.56	0.05
NAT+MWE	Output-Nbest	19.97	27.34	0.05
	Input-Nbest	19.90	27.29	0.05

5.2. Modeling Setup

All experiments are conducted with Transformer on Espnet toolkit [29]. For Autoregressive Transformer (AT) models, they are configured with 12-layer encoder, and 6-layer decoder with 4-head attention. The attention dimension is 256. The AT models are also jointly trained with CTC plus attention methods, of which the weighting factor for the CTC part is 0.3. The input acoustic features are 83-dimensional including 80-dim log filter bank and 3-dim pitch features. For better performance, we adopt data augmentation methods. They are speed perturbation [30] and spec-augmentation [31] respectively. The output of the Transformer is BPE set. It consists of two parts which are 3000 BPEs for English and 3920 for single Chinese characters from the training transcript.

For CE training, all the Transformers are optimized with Noamoptimizer [7], with 0.1 drop-out configuration. All the final models are obtained by averaging the model parameters of the last 5 epochs. For MWE training, it proceeds with 5 epochs with 4-best hypothesis. For AT decoding, we adopt beam-search-based decoding method, with beam size being 10. For the CTC-mask-based NAT decoding, we follow the iterative inference fashion as in [17]. The tokens whose CTC posteriors are below 0.9 are masked. Besides, we fix the max iterations with 10 during decoding.

5.3. Performance metric

Two main metrics are employed to evaluate ASR performance. They are MER(%) and RTF respectively. MER(%) is an abbreviation for Mixed Error Rate that consists of word error rate for English and character error rate for Mandarin. The Real Time Factor (RTF) is obtained under the scenario where all test sets are decoded with single thread, by a single core of Intel(R) Xeon(R) Gold 6238R, 2.20GHz.

5.4. Results

Table 2 presents the MER (%) and RTF results of the proposed NAT methods on the two test sets in contrast with the AT method. From table 2, we observe that the proposed C-Mask method outperforms the R-Mask baseline on $eval_{sg}$ with slightly degraded MER result on $eval_{man}$. For the code-switching-based masking methods, it seems that language specific masking method favors corresponding language dominated test set, that is, the E-Mask improves on $eval_{sg}$ while the M-Mask obtains the best MER on $eval_{man}$. Among all the masking methods, F-Mask gets the best MER on $eval_{sg}$ and a second best MER on $eval_{man}$. So we continue our MWE training from F-Mask NAT model. For the MWE training, both Nbest generation methods yield consistent performance improvement over the best NAT CE training method, and the Input-Nbest slightly outperforms the Output-Nbest. Finally, though all of the proposed methods for the NAT framework still have gap with the AT one in terms of MER, we still obtain 0.38% and 1.04% absolute MER reduction respectively compared with the baseline R-Mask method. Additionally, the RTF of the proposed NAT method is over 10x times faster.

Table 3. MER(%) of words located at code-switching points, as defined in Section 4.2. Refer to Table 2 for the details of name notation.

Modeling method & Setup		$eval_{man}$	$eval_{sg}$	RTF
AT+CE	Transformer	23.1	26.8	0.68
	+ SA	19.6	23.3	0.70
NAT+CE	R-Mask [17]	23.1	26.7	0.05
	C-Mask	23.3	26.5	0.05
	E-Mask	23.5	26.1	0.05
	M-Mask	23.1	26.8	0.05
	S-Mask	23.8	27.0	0.05
	F-Mask	23.6	26.9	0.05
NAT+MWE	Output-Nbest	23.5	26.7	0.05
	Input-Nbest	23.4	26.6	0.05

Table 3 reports the MER results averaged among all code-switching points. From Table 3, we find the NAT method is much worse in MER at code-switching points than the AT method. Besides, language specific masking method seems favor corresponding recognition of the languages that are masked. This suggests NAT-based code-switching ASR has potential to improve.

6. CONCLUSION

In this paper, we proposed to employ Minimum Word Error (MWE) criterion to train non-autoregressive Transformer models under code-switching ASR scenario. To obtain a strong non-autoregressive baseline for code-switching ASR, we attempted different token masking methods at code-switching points in utterance. We found the proposed masking methods can yield improved performance. Based on the best CE trained NAT model, the MWE criterion further yields mixed error rate reduction, thanks to the proposed Nbest-list-based hypothesis generation methods. Since our Nbest hypothesis generation methods are a little bit mismatched with what we are doing for masking during decoding, the performance might be compromised. In future, we are exploring more desired Nbest list generation methods, hopefully yielding more gains in terms of MER reduction.

7. REFERENCES

- [1] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proc. ICML 2006*, pp. 369–376.
- [2] Kalpesh Krishna, Liang Lu, Kevin Gimpel, and Karen Livescu, “A study of all-convolutional encoders for connectionist temporal classification,” in *Proc. ICASSP 2018*.
- [3] A. Graves, “Sequence transduction with recurrent neural networks,” *Computer Science*, vol. 58, no. 3, pp. 235–242, 2012.
- [4] K. Rao, Haim Sak, and R. Prabhavalkar, “Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer,” in *Proc. ASRU 2017*.
- [5] Linhao Dong, Shuang Xu, and Bo Xu, “Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition,” in *Proc. ICASSP 2018*. IEEE, pp. 5884–5888.
- [6] Jicheng Zhang, Yizhou Peng, Van Tung Pham, Haihua Xu, Hao Huang, and Eng Siong Chng, “E2E-Based Multi-Task Learning Approach to Joint Speech and Accent Recognition,” in *Proc. Interspeech 2021*, 2021, pp. 1519–1523.
- [7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” 2017.
- [8] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang, “Conformer: Convolution-augmented Transformer for Speech Recognition,” in *Proc. Interspeech 2020*, 2020, pp. 5036–5040.
- [9] Yerbolat Khassanov, Haihua Xu, Van Tung Pham, Zhiping Zeng, Eng Siong Chng, Chongjia Ni, and Bin Ma, “Constrained Output Embeddings for End-to-End Code-Switching Speech Recognition with Only Monolingual Data,” in *Proc. Interspeech 2019*, 2019, pp. 2160–2164.
- [10] Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher, “Non-autoregressive neural machine translation,” 2018.
- [11] Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi, “Non-autoregressive machine translation with latent alignments,” in *Proc. EMNLP*, 2020.
- [12] Nanxin Chen, Shinji Watanabe, Jesús Villalba, and Najim Dehak, “Listen and fill in the missing letters: Non-autoregressive transformer for speech recognition,” 2020.
- [13] Edwin G. Ng, Chung-Cheng Chiu, Yu Zhang, and William Chan, “Pushing the Limits of Non-Autoregressive Speech Recognition,” in *Proc. Interspeech*, 2021, pp. 3725–3729.
- [14] Pengcheng Guo, Xuankai Chang, Shinji Watanabe, and Lei Xie, “Multi-Speaker ASR Combining Non-Autoregressive Conformer CTC and Conditional Speaker Chain,” in *Proc. Interspeech*, 2021, pp. 3720–3724.
- [15] Xingchen Song, Zhiyong Wu, Yiheng Huang, Chao Weng, Dan Su, and Helen Meng, “Non-autoregressive transformer asr with ctc-enhanced decoder input,” in *Proc. ICASSP*, 2021, pp. 5894–5898.
- [16] Zhengkun Tian, Jiangyan Yi, Jianhua Tao, Ye Bai, Shuai Zhang, and Zhengqi Wen, “Spike-triggered non-autoregressive transformer for end-to-end speech recognition,” 2020.
- [17] Yosuke Higuchi, Shinji Watanabe, Nanxin Chen, Tetsuji Ogawa, and Tetsunori Kobayashi, “Mask ctc: Non-autoregressive end-to-end asr with ctc and mask predict,” 2020.
- [18] Shun-Po Chuang, Heng-Jui Chang, Sung-Feng Huang, and Hung yi Lee, “Non-autoregressive mandarin-english code-switching speech recognition,” 2021.
- [19] Yosuke Higuchi, Hirofumi Inaguma, Shinji Watanabe, Tetsuji Ogawa, and Tetsunori Kobayashi, “Improved mask-ctc for non-autoregressive end-to-end asr,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 8363–8367.
- [20] Jason Lee, Elman Mansimov, and Kyunghyun Cho, “Deterministic non-autoregressive neural sequence modeling by iterative refinement,” in *Proc. EMNLP*, 2018, pp. 1173–1182.
- [21] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, “Joint ctc-attention based end-to-end speech recognition using multi-task learning,” in *Proc. ICASSP*, 2017, pp. 4835–4839.
- [22] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R. Hershey, and Tomoki Hayashi, “Hybrid ctc/attention architecture for end-to-end speech recognition,” 2017.
- [23] 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,” 2020.
- [24] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer, “Mask-predict: Parallel decoding of conditional masked language models,” in *Proc. EMNLP-IJCNLP*, 2019, pp. 6112–6121.
- [25] Rohit Prabhavalkar, Tara N. Sainath, Yonghui Wu, Patrick Nguyen, Zhifeng Chen, Chung-Cheng Chiu, and Anjali Kannan, “Minimum word error rate training for attention-based sequence-to-sequence models,” in *Proc. ICASSP*, 2018, pp. 4839–4843.
- [26] Matt Shannon, “Optimizing Expected Word Error Rate via Sampling for Speech Recognition,” in *Proc. Interspeech*, 2017, pp. 3537–3541.
- [27] Haşim Sak, Matt Shannon, Kanishka Rao, and Françoise Beaufays, “Recurrent Neural Aligner: An Encoder-Decoder Neural Network Model for Sequence to Sequence Mapping,” in *Proc. Interspeech*, 2017, pp. 1298–1302.
- [28] Pengcheng Guo, Haihua Xu, Lei Xie, and Eng Siong Chng, “Study of Semi-supervised Approaches to Improving English-Mandarin Code-Switching Speech Recognition,” in *Proc. Interspeech 2018*, pp. 1928–1932.
- [29] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplín, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al., “Espnet: End-to-end speech processing toolkit,” *arXiv preprint arXiv:1804.00015*, 2018.
- [30] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, “Audio augmentation for speech recognition,” in *Proc. of INTERSPEECH 2015*, 2015.
- [31] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin Dogus Cubuk, and Quoc V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Proc. Interspeech 2019*.