RenewNAT: Renewing Potential Translation for Non-autoregressive Transformer

Pei Guo*, **Yisheng Xiao***, **Juntao Li†**, **Min Zhang**

Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University
{pguolst,ysxiaoo}@stu.suda.edu.cn, {ljt,minzhang}@suda.edu.cn

Abstract

Non-autoregressive neural machine translation (NAT) models are proposed to accelerate the inference process while maintaining relatively high performance. However, existing NAT models are difficult to achieve the desired efficiency-quality trade-off. For one thing, fully NAT models with efficient inference perform inferior to their autoregressive counterparts. For another, iterative NAT models can, though, achieve comparable performance while diminishing the advantage of speed. In this paper, we propose RenewNAT, a flexible framework with high efficiency and effectiveness, to incorporate the merits of fully and iterative NAT models. RenewNAT first generates the potential translation results and then renews them in a single pass. It can achieve significant performance improvements at the same expense as traditional NAT models (without introducing additional model parameters and decoding latency). Experimental results on various translation benchmarks (e.g., **4** WMT) show that our framework consistently improves the performance of strong fully NAT methods (e.g., GLAT and DSLP) without additional speed overhead.

Introduction

Transformer-based models have been widely applied to neural machine translation (NMT) (Dehghani et al. 2018; Wu et al. 2019; Liang et al. 2021; Takase and Kiyono 2021; Nguyen et al. 2020; Liang et al. 2022). Despite the excellent performance of these methods, they normally adopt an autoregressive (AR) decoding paradigm in which the tokens are predicted one by one in a strict left-to-right order. With such recurrent dependencies, the inference process is inevitably time-consuming, especially for long sentences. To mitigate this problem, much attention has been paid to the fully non-autoregressive (NAT) models (Xiao et al. 2022; Gu et al. 2018; Ghazvininejad et al. 2019; Qian et al. 2021), which can generate all tokens in parallel and thus accelerate the inference greatly. However, this increase in inference efficiency is at the cost of the translation quality, and there exists a considerable performance gap between NAT models and their AT counterparts. Many works have explored the potential reasons and attributed the quality decline in NAT models to *'the failure of capturing the target side dependency.'* (Gu and Kong 2021; Xiao et al. 2022). More specifically, fully NAT models learn to perform prediction without the internal dependency of the target sentence as a result of parallel decoding, unlike the AT models where the t-th token has previous t-1 contextual tokens' information to help its generation.

Significant efforts have been made to address the above-mentioned obstacles for fully NAT models from different aspects in recent years, e.g., adopting knowledge distillation (Gu et al. 2018; Zhou, Gu, and Neubig 2019), introducing latent variables (Ran et al. 2021; Liu et al. 2021), training with better criterion (Marjan et al. 2020; Du, Tu, and Jiang 2021) or learning strategies (Qian et al. 2021; Bao et al. 2022; Huang et al. 2022). Gu and Kong also combine these tricks for better performance. However, there still exists a performance gap with their strong autoregressive counterparts, indicating that the translation results are still relatively not reliable. Specifically, these models only introduce effective tricks in the training process and still lack explicit target side information to guide translation during inference. Thus, each target token is still predicted independently, and the well-known multi-modality problem (Gu et al. 2018) still hurts the translation performance seriously.

Another representative method is introducing iterative refinements for NAT models during inference (Ghazvininejad et al. 2019; Huang, Perez, and Volkovs 2022), called iterative NAT models. Iterative NAT models adopt multiple decoding steps and keep the non-autoregressive property in each step, where the tokens generated from the last step are fed into the model for refinements. Obviously, the result generated from the previous iteration can provide target-side information to help predict and do refinements in the next iteration. Compared with fully NAT models, iterative NAT models have achieved more exciting performance, even outperforming the vanilla Transformer. However, due to multiple decoding steps, the advantage of decoding speed will diminish, especially in some specific scenes.

As a result, how to effectively combine the merits of the above-mentioned two NAT paradigms needs further exploration, e.g., to refine the inferior translation results with the help of useful target-side information without decreasing the inference speed. In this work, we propose RenewNAT, a simple yet effective framework for non-autoregressive transla-

*These authors contributed equally.

†Corresponding Author

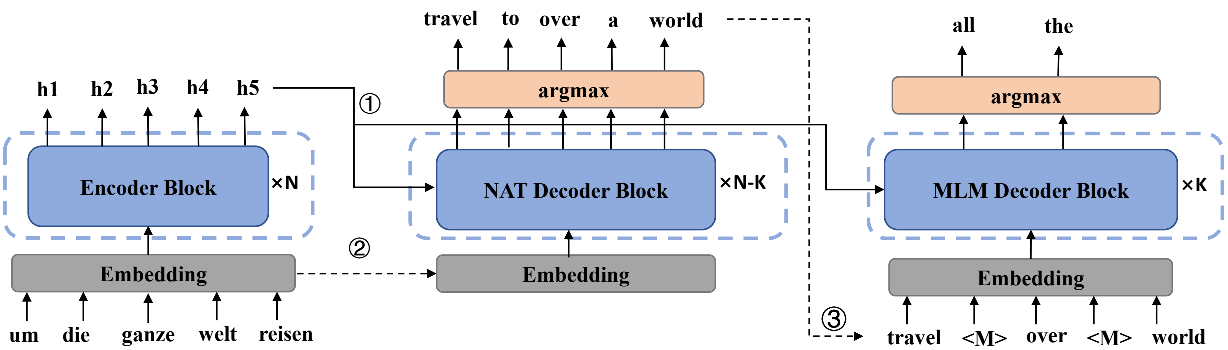

Figure 1: The architecture of our RenewNAT. Different from the traditional fully NAT models, RenewNAT splits the decoder into two sub-modules. The first one aims to predict the potential translation, which can be viewed as a noisy version of the final translation. The input to it is the embeddings of the source sentence with *soft copy* (No.2 dashed arrow). The second one tries to renew the potential translation to a credible one by masking the incorrect tokens. The input to it is the embeddings of potential translation generated by the first sub-module (No.3 dashed arrow). The No.1 solid arrow denotes the encoder-decoder attention.

tion, which first generates the potential translation and then renews them to a credible one. Concretely, as shown in Figure 1, the decoder is split into two sub-modules according to the specific function, where the first aims to generate a potential translation containing useful target side information, while the second attempts to capture the target side dependency by deriving some valuable contextual information from the potential translation. However, as the potential translation can be viewed as a noisy version of the final translation, it inevitably contains errors. These errors might be cascaded and amplified through the second sub-module, preventing it from capturing the correct target tokens dependency by directly learning from the potential translation. To best derive valuable information from it, it is critical to discard the incorrect tokens in the potential translation, Motivated by the masked language model (MLM) (Devlin et al. 2018; Ghazvininejad et al. 2019), we mask the incorrect tokens in the training of the second sub-module to keep the correctness of dependent target tokens. Compared with latent variable NAT models, which also spilt the model into two sub-modules and the first one is adopted to predict a latent variable sequence to help prediction during inference, our RenewNAT aims at providing more specific target side information to guide prediction. Moreover, latent variable NAT models need extra tools to provide latent variable information during training and this will increase the difficulty of training. In contrast, the first sub-module in our RenewNAT can be directly learned with ground truth tokens and this does not bring any additional expense.

Our RenewNAT is a generic framework in which various widely-used methods can be adopted to improve the quality of potential translation and provide more useful target side information for the second sub-module. In experiments, we evaluate our RenewNAT on IWSLT'14 English→German, WMT'14 English↔German, WMT'16 English↔Romanian and adopt a few effective and representative methods to improve the potential translation, such as GLAT (Qian et al. 2021) and DSLP (Huang et al. 2022). In a nutshell, the major contributions of our paper are as follows.

- We propose RenewNAT, which combines the merits of fully NAT and iterative NAT models. By splitting the model into two sub-modules, where the first one can provide useful target-side information to help prediction for the second one, we can alleviate the above-mentioned problem and meanwhile maintain the speed advantage.
- Experimental results on multiple machine translation benchmarks demonstrate that RenewNAT consistently improves the base model without the expensive cost of inference speed. Compared with the vanilla NAT model, RenewNAT achieves significant improvements by up to 2.0↑ of BLEU score on average. Compared with strong fully NAT models, e.g. GLAT (Qian et al. 2021) and DSLP (Huang et al. 2022), RenewNAT also shows its effectiveness by up to 0.62↑ of BLEU on GLAT and 0.3↑ of BLEU on DSLP, maintaining about 13 times speedup.

Related Work

Non-autoregressive models have attracted much attention in the field of sequence generation, such as machine translation (Gu et al. 2018), speech recognition (Higuchi et al. 2021), text summarization (Puyuan Liu and Mou 2022), grammatical error correction (Straka, Náplava, and Straková 2021), dialogue (Han et al. 2020) and etc. Compared with their auto-regressive counterparts, which generate the target tokens from left to right, non-autoregressive models predict the target sequence in parallel and accelerate the speed of inference greatly, but the generation quality declines. In this paper, we mainly focus on non-autoregressive translation (NAT) (Gu et al. 2018; Guo et al. 2019; Marjan et al. 2020; Gu and Kong 2021; Ran et al. 2021; Ding et al. 2021; Qian et al. 2021; Huang et al. 2021; Zhan et al. 2022; Bao et al. 2022), where much progress has been made to improve the performance in recent years.

The NAT model is first proposed in Gu et al. (2018), they adopt a fertility predictor to decide the number of the source token that will be aligned to and then help prediction. However, the translation performance is much lower than the vanilla Transformer because of the parallel decoding. They notice that the NAT model can not make predictions well without capturing the target side dependency, further leading to the serious multi-modality problem. Many methods have been proposed to alleviate this in recent years. Guo et al. (2019) propose phrase-table lookup and embedding mapping methods to enhance the decoder inputs with some target side information. Sun et al. (2019) utilizes conditional random fields to model the target positional contexts. Chan et al. (2020) tries to model the latent alignments to help prediction. Marjan et al. (2020) and Du, Tu, and Jiang (2021) introduce better criterion for NAT models to learn. Besides, utilizing latent variables as part of the model is also a popular method, such as the alignments between source and target tokens (Song, Kim, and Yoon 2021), positional information (Bao et al. 2019; Ran et al. 2021), syntactic labels (Akoury, Krishna, and Iyyer 2019; Liu et al. 2021). At the same time, to best find the trade-off between translation speed and quality, much progress has also been made for iterative NAT models. Instead of generating all target tokens in one pass, they learn the conditional distribution over partially observed generated tokens and adopt multiple steps to refine the previous results. There exist many specific ways for refinements, such as heuristic denoising (Lee, Mansimov, and Cho 2018; Savinov et al. 2021) insertion and deletion (Stern et al. 2019; Gu, Wang, and Zhao 2019) masking and recovering (Ghazvininejad et al. 2019; Kasai et al. 2020a) and so on. However, these methods improve translation accuracy at the expense of speedup. Recently, motivated by curriculum learning, Qian et al. (2021) introduces a glancing sampling strategy to dynamically select some positions based on the performance of the model, this significantly improves the performance of fully NAT models. Based on their simple method and excellent performance, many works further improve the performance. Gu and Kong (2021) attributes the key to fully NAT models to adopting dependency reduction in the learning space of output tokens. They combine several useful skills during training and finally achieve several SOTA results. Huang et al. (2022) introduce deep supervision and additional layer-wise prediction (DSLP) for each decoder layer and Zhan et al. (2022) propose a general approach to enhance the target dependency within the NAT decoder from decoder input and decoder self-attention, setting new SOTA results for fully NAT models. However, many methods for fully NAT models just introduce effective tricks in the training process, during inference, they are still unable to get any specific target side tokens to help do prediction, e.g., latent variables only model the latent information and this is different with the specific tokens, GLAT (Qian et al. 2021) and DSLP (Huang et al. 2022) keep the decoding process unchanged which prevents the model predicting tokens conditional on target side tokens. There is still room for improvement by modifying the traditional decoding process for fully NAT models.

Methodology

In this section, we first give a brief description of fully NAT models and iterative NAT models. Then, we present our RenewNAT framework in detail, including its training and inference process with some effective schemes.

Non-Autoregressive Transformer

Vanilla NAT models are built upon the Transformer architecture (Vaswani et al. 2017) which adopts the encoder-decoder network. Given a dataset $D = \{(X, Y)_i\}_{i=1}^{N}$, where $(X, Y)_i$ is one of the paired sentence data, and N is the size of the dataset. $X = \{x_1, x_2, ..., x_{T_X}\}$ is the source sentence to be translated from source language $\mathcal{X}$ and $Y = \{y_1, y_2, ..., y_{T_Y}\}$ is the ground-truth sentence from target language $\mathcal{Y}$, where T_X and T_Y are the length of source and target sequence. The encoder is adopted to obtain the context information in X by mapping X to a feature vector H_{enc}, and the decoder adopts H_{enc} and the decoder input $\hat{Y}$ to generate the translation results Y. The goal of NAT models is to estimate the unknown conditional distribution $P(Y|X;\theta)$ by learning a mapping function $f(\cdot)$ from the source sentence to the target sentence, where θ denotes the parameter set of a NAT model. During training, fully NAT models use the conditional independent factorization for prediction, and the objective is to maximize:

$$\mathcal{L}_{\text{NAT}} = \sum_{t=1}^{T} \log P(y_t|X;\theta), \tag{1}$$

where T is the length of the target sentence. During training, T is the same as the length of the ground-truth target sentence T_Y, while in inference, T is usually predicted by a length prediction module P_L. Compared with their AT counterparts, it is obvious that the conditional tokens $y_{<t}$ are removed for NAT models. Hence, we can translate in parallel without auto-regressive dependencies, and improve the inference speed greatly.

For iterative NAT models, they also adopt the conditional independence assumption identically, but they try to rebuild the dependency of the target tokens in an iterative fashion. During inference, they keep the non-autoregressive property in every iteration step and refine the translation results during different iteration steps. Specifically, in the first iteration, only X is fed into the model, which is the same as NAT models. After that, each iteration takes the translation generated from the last iteration as context for refinement to decode the translation. The training goal of the iterative-based NAT models is to maximize:

$$\mathcal{L}_{\text{Iter}} = \sum_{y_t \in Y_{tgt}} \log P(y_t|\hat{Y}, X;\theta), \tag{2}$$

where $\hat{Y}$ indicates the translation result of the last iteration, and Y_{tgt} is the target of current iteration. Taken CMLM (Ghazvininejad et al. 2019) as an example, it explicitly learns the conditional distribution over partially observed reference tokens, and then during inference, several tokens with low confidence will be masked and predicted again in the next iteration based on the tokens unmasked.

RenewNAT Framework

In order to effectively alleviate the problem mentioned above, we hope our proposed model is able to do prediction conditioned on specific target tokens during inference while maintaining the speed advantage without iterative operation. We propose RenewNAT, a simple yet effective framework. As shown in Figure 1, RenewNAT splits the traditional NAT decoder into two sub-modules, called NAT sub-module and MLM sub-module. The NAT sub-module aims to predict the potential translation to provide necessary target tokens while the MLM sub-module can derive some valuable contextual information and then renew the potential translation to a credible one. Next, we will introduce these two sub-modules in detail.

NAT Sub-module Formally, given the source sentence X, the encoder takes X as input and captures the contextual information by N transformer encoder layers, denoted by:

$$E_X = \mathrm{Enc}_{1:N}(X) = \mathrm{Layer}^{enc}_{1:N}(emb(X)), \tag{3}$$

where $\mathrm{Enc}_{1:N}$ denotes the encoder module, $\mathrm{Layer}^{enc}_{1:N}$ denotes the first to Nth encoder layer, $emb(\cdot)$ is the embedding function used to map the discrete words into vector representations and E_X is the encoder's hidden representation of the last layer, and the input to our NAT sub-module $H = \{h_1, h_2, ..., h_T\}$ is copied from $emb(X)$ with *uniform copy* or *soft copy*, as:

$$H = f(emb(X)), \tag{4}$$

where $f(\cdot)$ is the copy function. Then NAT sub-module will predict the potential translation with a softmax layer based on the hidden states of the last layer, denoted by:

$$E_{Y_{pot}} = \mathrm{Dec}_{1:N\text{-}K}(H, E_X) = \mathrm{Layer}^{dec}_{1:N\text{-}K}(H, E_X) \tag{5}$$

$$P(Y_{pot}|E_{Y_{pot}}) = \mathrm{softmax}(W \cdot E_{Y_{pot}}), \tag{6}$$

where $E_{Y_{pot}}$ is the hidden states of the last layer of NAT sub-module, $\mathrm{Dec}_{1:N-k}$ denotes the NAT sub-module, $\mathrm{Layer}^{dec}_{1:N-k}$ denotes the first to $N-k$th decoder layer, W is a trainable matrix, Y_{pot} is the potential translation results of our NAT sub-module. Since the potential translation can be viewed as a noisy version of the final translation, we can simply train our NAT sub-module with ground-truth tokens, so the training process is the same as fully NAT models in Equation 1, we denote the training loss as $\mathcal{L}_{\text{pot}}$ here:

$$\mathcal{L}_{\text{pot}} = \sum_{t=1}^{T} \log P(y_t|X;\theta). \tag{7}$$

MLM Sub-module After NAT sub-module predicts the potential translation T_{pot}, MLM sub-module aims to derive some valuable contextual information from it and then renew it to a credible one during inference. We do not change the internal structure of MLM sub-module layers, but to match the inference process mentioned above specifically, we change the training scheme to urge it to rebuild the dependency of the target tokens. Fortunately, we find the training scheme in CMLM (Ghazvininejad et al. 2019) with some modification can satisfy our assumption. While the tokens masked can be viewed as incorrect tokens in potential translation and the tokens unmasked can be viewed as valuable contextual information. Formally, given the target sentence Y, MLM sub-module adopts the uniform masking scheme to choose a set of target tokens and replace them with [MASK] token. As a result, Y is divided into Y_{mask} and Y_{obs}, denoted as Y', then serves as the input of MLM sub-module. MLM sub-module also utilizes the encoder output E_X into an encoder-decoder attention module to obtain source information. We can describe the modeling process as follow:

$$\begin{aligned} E_Y &= \mathrm{Dec}_{N\text{-}K+1:N}(Y', E_X) \\ &= \mathrm{Layer}^{dec}_{N\text{-}K+1:N}(emb(Y'), E_X) \end{aligned} \tag{8}$$

$$P(Y_{mlm}|E_Y) = \mathrm{softmax}(W' \cdot E_Y), \tag{9}$$

where E_Y is the hidden states of the last layer of MLM sub-module, $\mathrm{Dec}_{N\text{-}K+1:N}$ denotes MLM sub-module, $\mathrm{Layer}^{dec}_{N\text{-}K+1:N}$ denotes the $(N-K+1)$th to Nth decoder layer, W' is a trainable matrix, Y_{mlm} is the outputs of MLM module, notice that only the tokens in masked position are valuable. Such during training, we compute the loss of masked tokens only, denoted by $\mathcal{L}_{\text{mlm}}$, which can be computed as:

$$\mathcal{L}_{\text{mlm}} = -\sum_{y_t \in Y_{mask}} \log P(y_t|Y_{obs}, X; \theta). \tag{10}$$

Algorithm 1: RenewNAT Training

Input: training data $D = \{(X_i, Y_i)\}$, learning rate γ
Output: model parameters θ

```
Initialize model with parameters θ.
while not convergent do
    for (X, Y) ∈ D do
        generate Ŷ using NAT sub-module
        sample mask from Y: Y_mask ← Y
        predict Y_mask using MLM module
        compute loss L ← L_pot + L_mlm + L_len
        update model parameters θ ← θ - γ ∂/∂θ L
    end for
end while
return θ
```

Training and Inference In the previous subsection, we introduced our RenewNAT framework and two decoder sub-modules. In order to describe the training procedure for RenewNAT clearly, Algorithm 1 is given to outline. During training, NAT sub-module and MLM sub-module are trained jointly. Besides, an extra module is trained to predict the target length, which is implemented as in previous works (Qian et al. 2021; Huang et al. 2022). An additional [LENGTH] token is added to the source input, and the encoder output for the [LENGTH] token is used to predict the length. We denote the loss of the length prediction module as $\mathcal{L}_{\text{len}}$ here, then the final training loss of our RenewNAT is $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_{\text{pot}} + \mathcal{L}_{\text{mlm}} + \mathcal{L}_{\text{len}}. \tag{11}$$

During inference, RenewNAT generates the translation with only a single pass. It first predicts the target length and then utilizes the NAT decoder sub-module to generate a potential translation result, then masks the incorrect tokens in potential translation and utilizes MLM decoder sub-module to renew them. Since it is not easy to distinguish the incorrect tokens in potential translation, we try many strategies and find

Models		I_{dec}	WMT14 EN→DE	WMT14 DE→EN	WMT16 EN→RO	WMT16 RO→EN	Speedup
AT Models	Transformer (Vaswani et al. 2017)	T	27.30	/	/	/	/
	Transformer (Qian et al. 2021)	T	27.48	31.27	33.70	34.05	1.0×
	Transformer (ours) w/ KD	T	28.47	31.95	33.70	33.75	1.0×
Iterative NAT	NAT-IR (Lee, Mansimov, and Cho 2018)	10	21.61	25.48	29.32	30.19	1.5×
	LaNMT (Shu et al. 2020)	4	26.30	/	/	29.10	5.7×
	LevT (Gu, Wang, and Zhao 2019)	6+	27.27	/	/	33.26	4.0×
	Mask-Predict (Ghazvininejad et al. 2019)	10	27.03	30.53	33.08	33.31	1.7×
	JM-NAT (Guo, Xu, and Chen 2020)	10	27.69	32.24	33.52	33.72	/
	DisCO (Kasai et al. 2020a)	Adv	27.34	31.31	33.22	33.25	3.5×
	Multi-Task NAT (Hao et al. 2021)	10	27.98	31.27	33.80	33.60	1.7×
	RewriteNAT (Geng, Feng, and Qin 2021)	Adv	27.83	31.52	33.63	34.09	/
	CMLMC (Huang, Perez, and Volkovs 2022)	10	28.37	31.47	34.57	34.13	/
Fully NAT	NAT-FT (Gu et al. 2018)	1	17.69	21.47	27.29	29.06	15.6×
	Mask-Predict (Ghazvininejad et al. 2019)	1	18.05	21.83	27.32	28.20	/
	imit-NAT (Wei et al. 2019)	1	22.44	25.67	28.61	28.90	18.6×
	Flowseq (Ma et al. 2019)	1	23.72	28.39	29.73	30.72	1.1×
	CTC (Saharia et al. 2020)	1	25.70	28.10	32.20	31.60	18.6 ×
	Imputer (Saharia et al. 2020)	1	25.80	28.40	32.30	31.70	18.6 ×
	ReorderNAT (Ran et al. 2021)	1	22.79	27.28	29.30	29.50	16.1 ×
	SNAT (Liu et al. 2021)	1	24.64	28.42	32.87	32.21	22.6×
	AlignNAT (Song, Kim, and Yoon 2021)	1	26.40	30.40	32.50	33.10	11.3×
	Fully-NAT (Gu and Kong 2021)	1	26.51	30.46	**33.41**	**34.07**	16.5×
	OAXE-NAT (Du, Tu, and Jiang 2021)	1	26.10	30.20	32.40	33.30	15.3×
	DAD (Zhan et al. 2022)	1	26.43	30.42	33.07	33.82	15.1×
	†Vanilla NAT	1	19.34	24.42	29.19	29.61	13.6×
	†RenewNAT	1	22.38	27.16	31.01	32.30	13.1×
	†RenewNAT w/ NPD (m=5)	1	23.56	28.37	31.87	33.05	11.5×
	†GLAT	1	25.21	29.51	31.65	32.04	13.5×
	†RenewNAT	1	25.75	29.67	32.53	32.93	12.5×
	†RenewNAT w/ NPD (m=5)	1	**26.54**	**30.55**	32.86	33.35	10.7×
	†GLAT + DSLP	1	25.69	29.90	32.24	33.03	12.6×
	†RenewNAT	1	25.86	30.14	32.70	33.52	12.4×
	†RenewNAT w/ NPD (m=5)	1	**26.65**	**30.65**	33.02	33.74	11.2×

Table 1: Results on 4 WMT machine translation tasks. † denotes the results of our implementations.

that determining them by confidence is simple and effective. We set a threshold α to choose them, once the confidence is lower than α, we regard them as incorrect tokens and can not provide valuable information. Moreover, the best α may depend on the base model that predicts the potential translation. We will discuss more in the following experiments.

Experiment

Dataset

We evaluate our RenewNAT on five widely used machine translation benchmarks, including WMT14 EN↔DE (4.5M pairs), WMT16 EN↔RO (610K pairs) and IWSLT14 DE→EN (153K pairs). We follow the approach of Vaswani et al. (2017) to process WMT14 EN↔DE and adopt the approach to process data provided in Lee, Mansimov, and Cho (2018) for WMT16 EN↔RO. For IWSLT14 DE→EN dataset, we follow the steps in Guo et al. (2019).

Knowledge Distillation

Following the previous work (Gu et al. 2018; Wang et al. 2019; Guo et al. 2019), we adopt the sequence-level knowledge distillation for all datasets. Firstly, a vanilla Transformer (transformer-large for WMT14 En↔DE, transformer-base for WMT16 En↔RO and IWSLT14 DE→EN) is trained with raw data, and the results generated by the models on trained set are adopted as training data. Then we train RenewNAT with distilled data.

Evaluation Metrics

To evaluate our model, we report BLEU scores (Papineni et al. 2002) widely used in translation tasks to evaluate the performance. Speedup is measured by L_1^{GPU} following the previous work (Kasai et al. 2020b; Gu and Kong 2021; Helcl, Haddow, and Birch 2022), specifically, we perform generation on the test set and set the batch-size as 1, then we compare the latency of RenewNAT with Vanilla Transformer on a single Nvidia A5000 card. We use all scripts from Fairseq (Ott et al. 2019).

Experimental Settings

During training, we follow most of the hyperparameter settings in Gu and Kong (2021); Qian et al. (2021). For WMT datasets, we use base Transformer configuration (6 layers per stack, 8 attention heads per layer, 512 model dimensions, 2048 hidden dimensions) and adapt the warm-up learning rate schedule (Vaswani et al. 2017) with warming up to 5e-4 in 4k step. As IWSLT is a smaller dataset, we use a smaller

Model	K	IWSLT14 DE→EN
GLAT	-	32.49
RenewNAT (N=6)	2	33.00
	3	32.41
	4	32.03
RenewNAT (N=12)	4	32.23
	6	33.00
	8	32.90

Table 2: Performances on IWSLT14 DE→EN with different layer number of MLM module K.

Transformer model (6 layers per stack, 8 attention heads per layer, 512 model dimensions, 1024 hidden dimensions). We separately train the model with batches of 64k/8k tokens on WMT/IWSLT dataset and use Adam optimizer (Kingma and Ba 2014) β with (0.9, 0.999) and (0.9, 0.98) for GLAT and Vanilla NAT. We train all models for 300k steps and average the 5 best checkpoints chosen by validation BLEU scores as our final model for inference. It's noticed that we set K as 2 which achieves the best performance evaluated in Ablation Study. For AT models, we use a beam size of 5 for inference. For RenewNAT, we apply noisy parallel decoding denoted as NPD (Gu et al. 2018) and set the length beam as 5.

Baselines

In order to evaluate the effectiveness of our framework, we apply RenewNAT on vanilla NAT model (Gu et al. 2018) and two strong NAT models, GLAT (Qian et al. 2021) and DSLP (Huang et al. 2022), which significantly improve the performance of fully NAT models. For the auto-regressive baseline, we compare our method with the base Transformer trained with distilled data. We also report the results of the recent fully and iterative NAT models for better comparison.

Main Results

As shown in Table 1, we evaluate our RenewNAT with different base models. Compared with vanilla NAT, RenewNAT gains significant improvements on each dataset, especially on WMT14 DE→EN and WMT16 RO→EN datasets (up to +2.2 and +2.6 BLEU). Besides, upon two stronger fully NAT models, GLAT and DSLP, RenewNAT still achieves about 0.5 BLEU improvement, while keeping the inference speed and model parameters unchanged. The consistent improvements also verify the effectiveness of our approach and demonstrate that RenewNAT could be applied on various fully NAT models. Although the performance is still below strong iterative NAT models and their AT counterparts, its merit in decoding speed is significant.

Ablation Study

In this session, We mainly explore the influence of different factors for RenewNAT, including the corresponding layers to split the decoder, specific training strategy for MLM sub-module and the best decoding scheme. Besides, we also give more insights to show the effectiveness of RenewNAT compared with base models.

Model	IWSLT14 DE→EN
GLAT	32.49
RenewNAT (learning strategies)	
w/ renew [MASK]	33.00
w/ renew completely	31.14
w/ cmlmc	32.73
RenewNAT (inputs)	
w/ output tokens	32.73
w/ mixed tokens	33.22
w/ target tokens	33.00

Table 3: Performances on IWSLT14 DE→EN with different training strategies based on GLAT.

Effect of Layers of MLM Sub-module

Since RenewNAT splits the decoder into two sub-modules and adopts $N - K$ and K decoder layers for NAT decoder and MLM sub-module respectively, we wonder how the performance evolves through the number of MLM sub-module layers K. Also, we extend the decoder layers to 12. As shown in Table 2, when K is 2, RenewNAT performs best, but the performance degenerates quickly when K is 4, that's because when K is larger, the layers of NAT sub-module reduce, leading to bad quality of potential translation. Besides, it seems that adding the layers of each sub-module has no great improvements. However, with the given decoder layers N, the number of K significantly affects the performance, requiring a good balance of two sub-modules.

Effect of Training Strategy of MLM Sub-module

We also experimented with different training strategies for MLM sub-module, the results are shown in Table 3. Since MLM sub-module is designed to derive useful target-side information, First, we adopt the masking operation to realize this. We set the second module to renew each word in potential translation completely and train the second module as CMLMC (Huang, Perez, and Volkovs 2022). As a result, the performance declines seriously. Besides, from the perspective of MLM sub-module input, we propose several reasonable inputs with masking: target tokens, output tokens which are the output of NAT sub-module, and mixed tokens which mix ground-truth tokens and output tokens. Finally, taking mixed tokens as input outperforms others. In our opinion, mixed tokens can give some useful information about potential translation, which is matching to the inference process. Comparatively, output tokens may conclude many errors during training, which makes the training of MLM submodule difficult and unstable.

Effect of Distinguish Strategy

As mentioned above, there always exist some errors in potential translation, and these tokens hurt the second submodule to capture the target side dependency. We adopt the masking operation to discard these incorrect tokens, as a result, how to effectively distinguish them is critical for the performance of our RenewNAT, we try some simple methods to achieve this purpose. Firstly, we give a fixed masking ratio δ to mask the tokens with the lowest confidence,

Model	DE→EN			EN→RO		
	BLEU	Rep	Comet	BLEU	Rep	Comet
GLAT	29.51	1.05%	0.27	31.65	1.09%	0.29
RenewNAT	29.66	0.71%	0.30	32.52	0.59%	0.36
w/o MLM	29.02	1.32%	0.24	32.10	1.06%	0.31

Table 4: BLEU score and token repetition ratio on WMT14 DE→EN and WMT16 EN→RO.

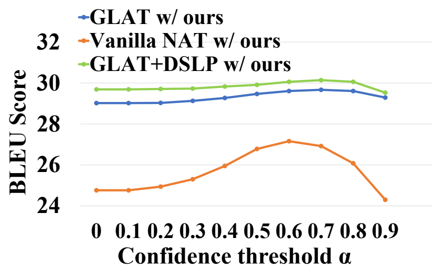

Figure 2: BLEU score with different confidence threshold on WMT14 DE→EN.

for the assumption that there is an almost fixed ratio of errors in potential translation, we find this method can obtain the high performance when δ is in [0.2, 0.5]. However, it is obvious that different potential translation results may have different incorrect ratios and the fixed ratio seems not optimal. Thus we follow another simple and effective method which utilizes a confidence threshold α to choose the incorrect tokens. Specifically, when the confidence of target tokens is lower than α, we regard them as incorrect tokens and mask them. Moreover, the best α is related to the base model, which is also easy to understand as the quality and distribution of the potential translation generated by different models are truly inconsistent. As shown in Figure 2, on WMT14 DE→EN dataset, Vanilla NAT and GLAT achieve the best performance when α is set to 0.6 and 0.7 respectively. More exploration of the reasons and the best threshold α for different settings are presented in Appendix.

Performance of MLM Sub-module

RenewNAT splits the traditional decoder module into two sub-modules, called NAT sub-module and MLM sub-module. Need to notice that NAT sub-module is similar to the traditional NAT decoder, just with fewer layers. Our assumption is that MLM sub-module can derive useful target side information from NAT sub-module. To give more insights to verify this, we wonder if MLM sub-module truly improves the performance and how this works. As shown in Table 4, we compare our RenewNAT with the corresponding base model GLAT and the one without MLM sub-module from different aspects. We can observe that RenewNAT achieves great improvements on BLEU score and Comet score of MLM sub-module (29.66 vs. 29.02, 32.52 vs. 32.10, 0.30 vs. 0.24 and 0.36 vs. 0.31), besides, MLM sub-module can effectively alleviate the repetition problem. The comparison between the base model further verifies the effectiveness of our RenewNAT, certificating that MLM sub-module can successfully capture the internal dependency of target tokens, then predicts the masked tokens effectively.

Model	EN→DE		DE→EN	
	BLEU	Comet	BLEU	Comet
GLAT	22.37	-0.19	28.55	0.08
w/ RenewNAT	22.72	-0.20	30.12	0.14

Table 5: Performances on the raw IWSLT14 EN↔DE.

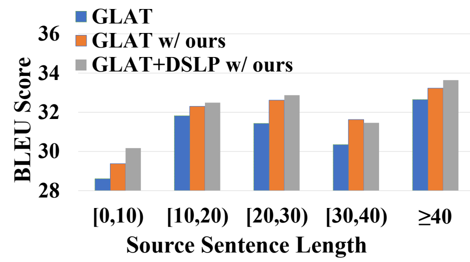

Figure 3: BLEU score with different source sentence length on WMT16 EN→RO.

Effect on Raw Data

Since we all conduct experiments on distilled data above, we wonder the effect of RenewNAT on raw data, we conduct experiments on IWSLT14 DE ↔ EN datasets without distillation. The results are shown in Table 5. As we can see, RenewNAT outperforms the corresponding base model on IWSLT DE→EN with + 1.42 BLEU score, demonstrating the robustness of our framework.

Effect of Source Input Length

We also explore the effect of sentence length for RenewNAT. Specifically, we divide the source sentences of WMT16 EN-RO test set into 5 intervals by the length after BPE operation and compute the BLEU score for each interval. Figure 3 shows the histogram results. We find that RenewNAT outperforms the correlated base model in each interval. When the source sentence length is in [20,40], RenewNAT achieves great improvement. As a result, RenewNAT can boost the performance under different length scenes.

Conclusion

In this work, we proposed RenewNAT, a simple yet effective framework with high efficiency and effectiveness. It first generates the potential translation to provide useful target side information and then renews them to capture the internal dependency of target tokens. We conduct experiments on five translation tasks with multiple representative base models (vanilla NAT and two strong fully NAT models). The results show that RenewNAT effectively improves the translation with little effect on decoding latency. In the future, we will explore more base models and whether our framework can be used in iterative NAT models to reduce decoding expense and meanwhile maintain performance advantage.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments. This work is supported by the National Science Foundation of China (NSFC No. 62206194), the Natural Science Foundation of Jiangsu Province, China (Grant No. BK20220488), and the Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions. This work is also supported by Beijing Academy of Artificial Intelligence (BAAI).

References

Akoury, N.; Krishna, K.; and Iyyer, M. 2019. Syntactically Supervised Transformers for Faster Neural Machine Translation. In *ACL*, 1269–1281.

Bao, Y.; Zhou, H.; Feng, J.; Wang, M.; Huang, S.; Chen, J.; and Li, L. 2019. Non-autoregressive transformer by position learning. *arXiv preprint arXiv:1911.10677*.

Bao, Y.; Zhou, H.; Huang, S.; Wang, D.; Qian, L.; Dai, X.; Chen, J.; and Li, L. 2022. *latent*-GLAT: Glancing at Latent Variables for Parallel Text Generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, volume 1, 8398–8409.

Chan, W.; Saharia, C.; Hinton, G.; Norouzi, M.; and Jaitly, N. 2020. Imputer: Sequence modelling via imputation and dynamic programming. In *ICML*, 1403–1413. PMLR.

Dehghani, M.; Gouws, S.; Vinyals, O.; Uszkoreit, J.; and Kaiser, L. 2018. Universal Transformers. In *International Conference on Learning Representations*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, L.; Wang, L.; Liu, X.; Wong, D. F.; Tao, D.; and Tu, Z. 2021. Progressive Multi-Granularity Training for Non-Autoregressive Translation. In *Findings of ACL-IJCNLP*, 2797–2803.

Du, C.; Tu, Z.; and Jiang, J. 2021. Order-agnostic cross entropy for non-autoregressive machine translation. In *Proceedings of the 38th International Conference on Machine Learning*, 2849–2859. PMLR.

Geng, X.; Feng, X.; and Qin, B. 2021. Learning to Rewrite for Non-Autoregressive Neural Machine Translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3297–3308.

Ghazvininejad, M.; Levy, O.; Liu, Y.; and Zettlemoyer, L. 2019. Mask-Predict: Parallel Decoding of Conditional Masked Language Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 6112–6121.

Gu, J.; Bradbury, J.; Xiong, C.; Li, V. O.; and Socher, R. 2018. Non-Autoregressive Neural Machine Translation. In *International Conference on Learning Representations*.

Gu, J.; and Kong, X. 2021. Fully Non-autoregressive Neural Machine Translation: Tricks of the Trade. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 120–133.

Gu, J.; Wang, C.; and Zhao, J. 2019. Levenshtein Transformer. In *Advances in Neural Information Processing Systems*, volume 32, 11181–11191.

Guo, J.; Tan, X.; He, D.; Qin, T.; Xu, L.; and Liu, T.-Y. 2019. Non-autoregressive neural machine translation with enhanced decoder input. In *AAAI*, volume 33, 3723–3730.

Guo, J.; Xu, L.; and Chen, E. 2020. Jointly masked sequence-to-sequence model for non-autoregressive neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 376–385.

Han, Q.; Meng, Y.; Wu, F.; and Li, J. 2020. Non-autoregressive neural dialogue generation. *arXiv preprint arXiv:2002.04250*.

Hao, Y.; He, S.; Jiao, W.; Tu, Z.; Lyu, M.; and Wang, X. 2021. Multi-Task Learning with Shared Encoder for Non-Autoregressive Machine Translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3989–3996.

Helcl, J.; Haddow, B.; and Birch, A. 2022. Non-Autoregressive Machine Translation: It's Not as Fast as it Seems. *arXiv preprint arXiv:2205.01966*.

Higuchi, Y.; Chen, N.; Fujita, Y.; Inaguma, H.; Komatsu, T.; Lee, J.; Nozaki, J.; Wang, T.; and Watanabe, S. 2021. A Comparative Study on Non-Autoregressive Modelings for Speech-to-Text Generation. *arXiv preprint arXiv:2110.05249*.

Huang, C.; Zhou, H.; Zaïane, O. R.; Mou, L.; and Li, L. 2021. Non-autoregressive translation with layer-wise prediction and deep supervision. *arXiv preprint arXiv:2110.07515*.

Huang, C.; Zhou, H.; Zaïane, O. R.; Mou, L.; and Li, L. 2022. Non-autoregressive translation with layer-wise prediction and deep supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 10776–10784.

Huang, X. S.; Perez, F.; and Volkovs, M. 2022. Improving Non-Autoregressive Translation Models Without Distillation. In *International Conference on Learning Representations*.

Kasai, J.; Cross, J.; Ghazvininejad, M.; and Gu, J. 2020a. Parallel machine translation with disentangled context transformer. *arXiv preprint arXiv:2001.05136*.

Kasai, J.; Pappas, N.; Peng, H.; Cross, J.; and Smith, N. 2020b. Deep Encoder, Shallow Decoder: Reevaluating Non-autoregressive Machine Translation. In *ICLR*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lee, J.; Mansimov, E.; and Cho, K. 2018. Deterministic Non-Autoregressive Neural Sequence Modeling by Iterative Refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1173–1182.

Liang, X.; Wu, L.; Li, J.; Qin, T.; Zhang, M.; and Liu, T.-Y. 2022. Multi-Teacher Distillation With Single Model for

Neural Machine Translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30: 992–1002.

Liang, X.; Wu, L.; Li, J.; Wang, Y.; Meng, Q.; Qin, T.; Chen, W.; Zhang, M.; Liu, T.-Y.; et al. 2021. R-drop: regularized dropout for neural networks. In *Advances in Neural Information Processing Systems*, volume 34.

Liu, Y.; Wan, Y.; Zhang, J.; Zhao, W.; and Philip, S. Y. 2021. Enriching Non-Autoregressive Transformer with Syntactic and Semantic Structures for Neural Machine Translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, 1235–1244.

Ma, X.; Zhou, C.; Li, X.; Neubig, G.; and Hovy, E. 2019. FlowSeq: Non-Autoregressive Conditional Sequence Generation with Generative Flow. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 4282–4292.

Marjan, G.; Karpukhin, V.; Zettlemoyer, L.; and Levy, O. 2020. Aligned cross entropy for non-autoregressive machine translation. In *Proceedings of the International Conference on Machine Learning*, 3515–3523. PMLR.

Nguyen, X.-P.; Joty, S.; Wu, K.; and Aw, A. T. 2020. Data diversification: A simple strategy for neural machine translation. In *Advances in Neural Information Processing Systems*, volume 33, 10018–10029.

Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; and Auli, M. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Puyuan Liu, C. H.; and Mou, L. 2022. Learning Non-Autoregressive Models from Search for Unsupervised Sentence Summarization. In *ACL*.

Qian, L.; Zhou, H.; Bao, Y.; Wang, M.; Qiu, L.; Zhang, W.; Yu, Y.; and Li, L. 2021. Glancing Transformer for Non-Autoregressive Neural Machine Translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 1993–2003.

Ran, Q.; Lin, Y.; Li, P.; and Zhou, J. 2021. Guiding Non-Autoregressive Neural Machine Translation Decoding with Reordering Information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 13727–13735.

Saharia, C.; Chan, W.; Saxena, S.; and Norouzi, M. 2020. Non-Autoregressive Machine Translation with Latent Alignments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 1098–1108.

Savinov, N.; Chung, J.; Binkowski, M.; Elsen, E.; and Oord, A. v. d. 2021. Step-unrolled denoising autoencoders for text generation. *arXiv preprint arXiv:2112.06749*.

Shu, R.; Lee, J.; Nakayama, H.; and Cho, K. 2020. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 8846–8853.

Song, J.; Kim, S.; and Yoon, S. 2021. AligNART: Non-autoregressive Neural Machine Translation by Jointly Learning to Estimate Alignment and Translate. In *EMNLP*, 1–14.

Stern, M.; Chan, W.; Kiros, J.; and Uszkoreit, J. 2019. Insertion transformer: Flexible sequence generation via insertion operations. In *ICML*, 5976–5985. PMLR.

Straka, M.; Náplava, J.; and Straková, J. 2021. Character Transformations for Non-Autoregressive GEC Tagging. In *W-NUT*, 417–422.

Sun, Z.; Li, Z.; Wang, H.; He, D.; Lin, Z.; and Deng, Z. 2019. Fast Structured Decoding for Sequence Models. *NeurIPS*, 32: 3016–3026.

Takase, S.; and Kiyono, S. 2021. Rethinking perturbations in encoder-decoders for fast training. *arXiv preprint arXiv:2104.01853*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Wang, Y.; Tian, F.; He, D.; Qin, T.; Zhai, C.; and Liu, T.-Y. 2019. Non-autoregressive machine translation with auxiliary regularization. In *AAAI*, volume 33, 5377–5384.

Wei, B.; Wang, M.; Zhou, H.; Lin, J.; and Sun, X. 2019. Imitation Learning for Non-Autoregressive Neural Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1304–1312.

Wu, L.; Wang, Y.; Xia, Y.; Tian, F.; Gao, F.; Qin, T.; Lai, J.; and Liu, T.-Y. 2019. Depth Growing for Neural Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5558–5563.

Xiao, Y.; Wu, L.; Guo, J.; Li, J.; Zhang, M.; Qin, T.; and Liu, T.-y. 2022. A Survey on Non-Autoregressive Generation for Neural Machine Translation and Beyond. *arXiv preprint arXiv:2204.09269*.

Zhan, J.; Chen, Q.; Chen, B.; Wang, W.; Bai, Y.; and Gao, Y. 2022. Non-autoregressive Translation with Dependency-Aware Decoder. *arXiv preprint arXiv:2203.16266*.

Zhou, C.; Gu, J.; and Neubig, G. 2019. Understanding Knowledge Distillation in Non-autoregressive Machine Translation. In *International Conference on Learning Representations*.