Jointly Masked Sequence-to-Sequence Model for Non-Autoregressive Neural Machine Translation

Junliang Guo **Linli Xu*** **Enhong Chen**

Anhui Province Key Laboratory of Big Data Analysis and Application,
School of Computer Science and Technology,
University of Science and Technology of China
guojunll@mail.ustc.edu.cn {linlixu,cheneh}@ustc.edu.cn

Abstract

The masked language model has received remarkable attention due to its effectiveness on various natural language processing tasks. However, few works have adopted this technique in the sequence-to-sequence models. In this work, we introduce a jointly masked sequence-to-sequence model and explore its application on non-autoregressive neural machine translation (NAT). Specifically, we first empirically study the functionalities of the encoder and the decoder in NAT models, and find that the encoder takes a more important role than the decoder regarding the translation quality. Therefore, we propose to train the encoder more rigorously by masking the encoder input while training. As for the decoder, we propose to train it based on the consecutive masking of the decoder input with an n-gram loss function to alleviate the problem of translating duplicate words. The two types of masks are applied to the model jointly at the training stage. We conduct experiments on five benchmark machine translation tasks, and our model can achieve 27.69/32.24 BLEU scores on WMT14 English-German/German-English tasks with 5+ times speed up compared with an autoregressive model.

1 Introduction

The encoder-decoder based sequence-to-sequence framework (Sutskever et al., 2014; Bahdanau et al., 2014) has achieved great success on the task of Neural Machine Translation (NMT) (Wu et al., 2016; Gehring et al., 2017; Vaswani et al., 2017; Hassan et al., 2018; Sheng et al., 2020). In this framework, the encoder takes the source sentence as input and extracts its hidden representation, based on which the decoder generates the target sentence word by word and from left to right, i.e., in an *autoregressive* manner, which is a natural bottleneck for the inference speed due to the sequential conditional dependence.

As the performance of NMT models have been substantially promoted, the translation efficiency is becoming a new research hotspot. Non-autoregressive neural machine translation (NAT) models are proposed to reduce the translation latency while inference, by removing the conditional dependence between target tokens and predicting all tokens in parallel (Gu et al., 2017). As the context dependency cannot be utilized while decoding, the inference speedup of NAT models comes at the cost of the degradation in performance. As studied by previous works (Guo et al., 2019; Wang et al., 2019), the inferior accuracy of NAT models mainly occurs from two aspects: 1) the source-side information is not adequately encoded which results in incomplete translation; 2) the decoder cannot handle the task well which leads to repeated translations and poor performance on long sentences.

To tackle these problems and promote the performance of NAT models, in this paper, we empirically conduct a thorough study on the functionalities of the encoder and decoder in NAT models, and conclude that the encoder has a more direct influence on the final translation performance, and is harder to train than the decoder. Therefore, we propose a jointly masked sequence-to-sequence model which is inspired by the idea of masked language modeling (Devlin et al., 2018). Specifically, for the encoder, we follow the masking strategy of BERT (Devlin et al., 2018) and randomly mask a number of tokens of the source sentence. This strategy trains the encoder more rigorously by forcing it to encode the complete information with residual input. For the decoder, we mask the consecutive fragment of the target sentence to make the decoder concentrate more on predicting adjacent tokens, and propose an n-gram based loss function to learn

*Corresponding author.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 376–385
July 5 - 10, 2020.

the consecutive tokens as a whole objective. In this way, we can alleviate the problem of repeated translations of NAT models. During inference, we adopt a mask-and-predict (Ghazvininejad et al., 2019) strategy to iteratively generate the translation result, which masks and predicts a subset of the current translation candidates in each iteration.

We verify the effectiveness of our model on five benchmark translation tasks including WMT14 English ↔ German, WMT16 English ↔ Romanian and IWSLT14 German → English. Our model outperforms all the NAT models in comparison, and can achieve comparative performance with its autoregressive counterpart while enhanced with 5+ times speedup on inference (27.69/32.24 BLEU scores and 5.73 times speedup on the WMT14 En-De/De-En tasks with an autoregressive teacher of 28.04/32.69 BLEU scores).

Our main contributions can be summarized as follows:

- While previous works only concentrate on manipulating the decoder, we illustrate and emphasize the importance of the encoder in NAT models and propose the encoder masking strategy to improve its training.
- We propose the consecutive masking strategy of the decoder input and the n-gram loss function to alleviate the problem of repetitive translations of NAT models.
- We integrate the two parts above in the jointly masked sequence-to-sequence model which shows strong performance on benchmark machine translation datasets.

2 Related Work

2.1 Non-Autoregressive Machine Translation

Neural machine translation (NMT) models have achieved great success in recent years. Traditional NMT models are based on the sequence-to-sequence framework (Bahdanau et al., 2014; Sutskever et al., 2014), taking the source sentence as input and generating the target sentence in an autoregressive manner. Specifically, given the source sentence $x = (x_1, x_2, ..., x_{T_x})$, the target sentence $y = (y_1, y_2, ..., y_{T_y})$ is generated as:

$$P(y|x) = \prod_{t=1}^{T_y} P(y_t|y_{<t}, x; \theta_{\text{enc}}, \theta_{\text{dec}}), \quad (1)$$

where $y_{<t}$ indicates the generated target tokens before timestep t, and θ_{enc} and θ_{dec} denote the parameters of the encoder and decoder respectively. For a target sentence with length n, autoregressive models have to take $O(n)$ iterations to generate it during inference. To break the sequential conditional dependency and make the generation process parallelizable, non-autoregressive machine translation (NAT) models are proposed to generate all target tokens independently (Gu et al., 2017) and reduce the time complexity from $O(n)$ to $O(k)$ where k is a constant number:

$$P(y|x) = P(T_y|x) \cdot \prod_{t=1}^{T_y} P(y_t|x; \theta_{\text{enc}}, \theta_{\text{dec}}), \quad (2)$$

where $P(T_y|x)$ is the explicit length prediction process for NAT models. Although the inference speed of NAT is significantly boosted, the translation accuracy is sacrificed due to the lack of context information at the target side. Therefore, lots of works have been conducted to promote the performance of NAT models. Specifically, Gu et al. (2017) takes a copy of the encoder input x as the decoder input and trains a fertility predictor to guide the copy procedure. Lee et al. (2018) and Ghazvininejad et al. (2019) generate the target sentence by iteratively refining the current translation. Other works enhance the performance of NAT models by utilizing auxiliary information, such as extra loss functions (Wang et al., 2019; Li et al., 2019; Sun et al., 2019; Wei et al., 2019; Shao et al., 2019), SMT components (Guo et al., 2019) and fine-tuning from an AT model (Guo et al., 2020). Recently, some works (Stern et al., 2019; Welleck et al., 2019; Gu et al., 2019) propose to change the generation order from the traditional left-to-right manner to a tree-based manner, resulting in a time complexity of $O(\log n)$. In this paper, we focus on the NAT model with $O(k)$ generation complexity.

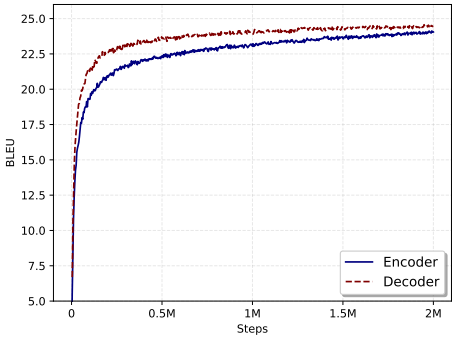
2.2 Masked Language Model

The masked language model proposed by BERT (Devlin et al., 2018) has become the essential component of the state-of-the-art pre-training methods (Song et al., 2019; Dong et al., 2019; Liu et al., 2019; Joshi et al., 2019; Lample and Conneau, 2019) in natural language understanding tasks. The standard paradigm of masked language modeling is to substitute a subset of tokens in the input sentence by a special symbol `[MASK]`, and predict the missing tokens by the residual ones. We denote the residual tokens as x^r and the masked target tokens as x^m.

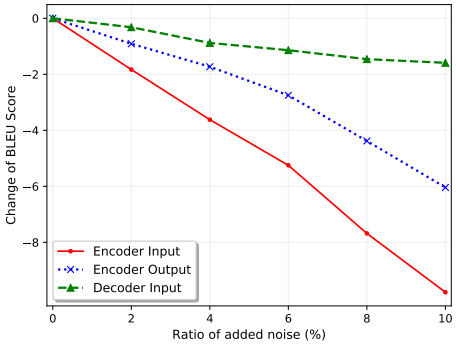
Δ Layers	+5	+10	+15
Δ Enc BLEU	+0.71	+1.05	+1.26
Δ Dec BLEU	+0.12	+0.18	+0.20

Table 1: The comparison of gains in BLEU score on the test set of the IWSLT14 German-English task when adding more layers to the encoder and decoder respectively of the NAT model.

As BERT is designed for language understanding tasks which can be handled with a single Transformer encoder, it is non-trivial to extend the paradigm into NMT tasks, where a sequence-to-sequence framework is utilized. To address that, XLM (Lample and Conneau, 2019) concatenates the source sentence and the target sentence as the encoder input to let the model learn the cross-lingual information, but still using a single Transformer encoder. MASS (Song et al., 2019) presents a sequence-to-sequence pre-training framework, which takes x^r as the encoder input and takes x^m as the decoder input as well as the target, still yielding a monolingual pre-training framework. In this paper, we propose a jointly masked language modeling method to handle the cross-lingual challenge in a unified sequence-to-sequence framework, based on which the translation accuracy of AT models and the inference speedup of NAT models can both be preserved.



(a) Convergence Speed



(b) Performance with Noisy Input

Figure 1: (a) The convergence speed of the encoder and the decoder. (b) The performance when adding noise to the encoder input, encoder output and decoder input in the inference stage of a basic NAT model.

3 Preliminary Study

To explore the functionalities of the encoder and decoder in NAT models, we conduct a thorough empirical study. We mainly follow the settings in (He et al., 2019). We train a basic NAT model proposed by Gu et al. (2017), except that we remove the fertility predictor and keep the decoder input as a hard copy of the source sentence in a similar way with (Guo et al., 2019; Wang et al., 2019). We conduct the following experiments on the IWSLT14 German to English dataset and train the model with the same number of training steps for each setting.

We study the importance of the encoder and decoder from three aspects. Firstly, we vary the number of encoder and decoder layers respectively to see which will bring more performance gain. Specifically, on a basic model with a 5-layer encoder and a 5-layer decoder, we increase the number of layers to the encoder and decoder separately. Results are illustrated in Table 1, from which we can conclude that adding the layers of the encoder can bring more performance gain than the decoder.

Secondly, we compare the convergence speed of the encoder and decoder by initializing the NAT model with a pretrained decoder/encoder and fix it during training, while randomly initialize a trainable encoder/decoder. The convergence speed is illustrated by the BLEU score along with the training steps, as shown in Figure 1(a). From the results, we can observe that the decoder converges faster than the encoder. In conclusion, we find that the encoder is dealing with a more sophisticated task than the decoder, and the encoder is not adequately trained in the initial NAT model.

Thirdly, we further conduct an investigation on the encoder input, encoder output and decoder input to evaluate their importance in the inference stage. During inference, we add random noise to the three types of inputs respectively, by randomly replacing the embeddings of some tokens with random noise. This experiment is conducted on a basic 5-layer encoder and decoder NAT model, and the

results are illustrated in Figure 1(b). Obviously, the encoder input and encoder output both largely influence the translation quality, which implies that the encoder plays an important role in the inference of NAT models, while the decoder input is the least important due to its conditional independence in nature. In a word, the performance of NAT models rely more on the encoder rather than the decoder.

4 Methodology

While most existing NAT works only focus on refining the decoder to obtain better performance, we have explored and shown the significance of the encoder in the previous section. Therefore, we propose to improve the translation performance by further manipulating the encoder, and we will introduce the proposed framework to tackle the problems discussed above in this section. We start with the problem definition.

Problem Definition Given a pair of source and target sentence $(x, y) \in (\mathcal{X}, \mathcal{Y})$ from the parallel training dataset $\mathcal{X}$ and $\mathcal{Y}$, the negative log-likelihood objective function of an NMT model can be written as:

$$L_{\text{nll}}(x, y; \theta_{\text{enc}}, \theta_{\text{dec}}) = -\log P(y|x; \theta_{\text{enc}}, \theta_{\text{dec}}), \tag{3}$$

where the conditional probability can be either Equation (1) or Equation (2) for AT or NAT models, and $\theta_{\text{enc}}, \theta_{\text{dec}}$ represent the parameters of the encoder and decoder respectively.

4.1 Encoder Masking

As studied in Section 3, the encoder needs to handle a harder task than the decoder but is not adequately trained in previous works. To maximize the functionality of the encoder, we propose to train it with masked language modeling.

The general masking strategy is as follows. Given a source sentence $x = (x_1, x_2, ..., x_{T_x})$, we randomly sample a subset from x, denoted as x^m with T_x^m tokens, and substitute them with other tokens in position. Specifically, we follow the similar substitution strategy as BERT (Devlin et al., 2018): we randomly select 10% of the tokens in x, of which 80% are substituted with a special symbol [MASK], 10% are substituted with a random token in the vocabulary, and 10% are kept unchanged. And we denote the substituted result of the source sentence as x^r. Then the loss function on the encoder of predicting the missing source tokens can be written as:

$$L_{\text{enc}}(x^m|x^r) = -\sum_{t=1}^{T_x^m} \log P(x_t^m|x^r). \tag{4}$$

4.2 Decoder Masking

For the decoder, as it is shown that the repetitive translations mainly result from the non-autoregressive nature of NAT, we alleviate this problem by applying a consecutive masking strategy and proposing a tailored n-gram based loss function. During training, given a target sentence $y = (y_1, y_2, ..., y_{T_y})$, we randomly select multiple sets of consecutive tokens and mask them in a similar strategy as masking the encoder. Each set contains n consecutive tokens, and we denote the masked target set as y^m and the substituted result as y^r, and their corresponding lengths as T_y^m and T_y. Note that in the decoder, the total number of masking tokens is uniformly sampled from 1 to T_y instead of being computed with a fixed ratio. We provide an illustration of our framework in Figure 2, where n is set to 2. The loss function of predicting the masked target tokens can be written as:

$$L_{\text{nll}}(y^m|x^r, y^r) = -\sum_{t=1}^{T_y^m} \log P(y_t^m|x^r, y^r). \tag{5}$$

We propose an n-gram based loss function, which has been applied to NMT models recently (Ma et al., 2018; Shao et al., 2018, 2019), to enhance the sentence-level information and alleviate the problem of repetitive translations of NAT models. The loss function is tied with the consecutive masking where n equals to the number of the consecutive masked tokens in each set. Specifically, given an n-gram $g = (g_1, ..., g_n)$, its occurrence count in the target sentence y can be written as: $C_y(g) = \sum_{t=0}^{T_y-n} \prod_{i=1}^{n} 1\{g_i = y_{t+i}\}$. As for the count in the masked sequence y^m, we introduce the probabilistic variant of the n-gram count to make the objective differentiable (Shao et al., 2018) by representing each token with the prediction probability:

$$\tilde{C}_{y^m}(g) = \sum_{t=0}^{T_y^m-n} \prod_{i=1}^{n} 1\{g_i = y_{t+i}^m\} \cdot p(y_{t+i}^m|x). \tag{6}$$

Considering all possible n-grams in y, the proposed

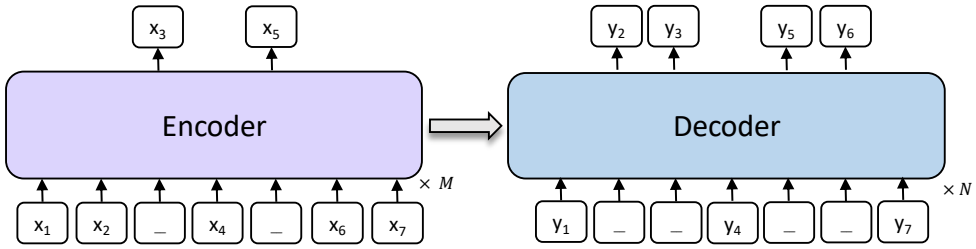


Figure 2: An illustration of the propose jointly masked sequence-to-sequence framework. "–" indicates that the token at this position is substituted by other tokens following the masking strategy. M and N indicate the number of layers of encoder and decoder respectively.

n-gram based loss function can be written as:

$$L_{\text{gram}}(y, y^m|y^r,x^r) = K - \sum_g \min(C_y(g), \tilde{C}_{y^m}(g)), \quad (7)$$

where $\min(C_y(g), \tilde{C}_{y^m}(g))$ represents the matching count between y and y^m w.r.t the n-gram g, and K is the upper bound of the total matching count which equals to the number of sets of consecutive masked tokens. The n-gram loss function will encourage the model to treat the consecutive masked tokens as a whole objective to match the sequential fragments in the target sentence, thus reducing the occurrence of repetitive translations.

4.3 Jointly Masked Model

Based on the proposed framework, the objective function of our model contains three parts: the traditional negative log-likelihood loss function to predict the missing target tokens $L_{\text{nll}}(\cdot)$, the prediction loss function on the encoder side $L_{\text{enc}}(\cdot)$, and the n-gram loss function $L_{\text{gram}}(\cdot)$. By integrating the three loss functions, given a training pair (x, y), the complete objective function of our model is:

$$\begin{aligned}\min_{\Theta} L(x, y) = & L_{\text{nll}}(y^m|x^r, y^r; \theta_{\text{enc}}, \theta_{\text{dec}}) \\ & + \alpha_1 L_{\text{enc}}(x^m|x^r; \theta_{\text{enc}}) \\ & + \alpha_2 L_{\text{gram}}(y, y^m|y^r, x^r; \theta_{\text{enc}}, \theta_{\text{dec}}),\end{aligned} \quad (8)$$

where $\Theta = (\theta_{\text{enc}}, \theta_{\text{dec}})$, α_1 and α_2 are the hyper-parameters that control the weights of different loss functions.

In the proposed training framework, the importance of the encoder has been emphasized by masking the encoder input and introducing $L_{\text{enc}}(\cdot)$. The encoder is encouraged to produce better representations of other tokens in order to predict the missing tokens. On the decoder side, the consecutive masking strategy augmented with the n-gram based loss function can help the model better capture the sentence-level information and alleviate the problem of repetitive translations.

4.4 Decoding Algorithm

For inference, we propose to iteratively refine the translation result in a mask-and-predict manner mainly following the strategy proposed in (Ghazvininejad et al., 2019), and details are introduced below.

During inference, the first step for NAT models is to determine the length of the target translation. We follow (Ghazvininejad et al., 2019) and introduce an additional prediction process to estimate the length by the source sentence. Specifically, we add a special token to the encoder and predict the target length with the output hidden vector of this token. The negative log-likelihood loss function of this token is then added to the word prediction loss in Equation (8) as the final loss. In experiments, we also consider selecting the translation with highest probability over multiple translation candidates with different target lengths to obtain better results.

Thereafter, based on the mask-and-predict paradigm, we design our decoding algorithm as follows. Given the target length T_y, we initiate the target sentence with `[MASK]` at all positions, and take it as the decoder input followed by conducting translation. Next, for each iteration, we apply consecutive masking to the translation candidates as we have done in the training stage. Specifically, we select several tokens with the lowest probabilities from the current translation candidates, and mask these tokens as well as their adjacent ones. The number of tokens to mask at each iteration follows a linear decay function utilized in (Ghazvininejad et al., 2019). As for the stop condition, the final translation is taken either when a pre-defined number of iterations is reached, or the translation candidates do not change between two iterations.

5 Experiments

5.1 Experimental Setup

5.1.1 Datasets

We evaluate our method on five widely used benchmark tasks: IWSLT14 German→English translation (IWSLT14 De-En)[1], WMT16 English↔Romanian translation (WMT16 En-Ro/Ro-En)[2], and WMT14 English↔German translation (WMT14 En-De/De-En)[3]. We strictly follow the dataset configurations of previous works. For the IWSLT14 De-En task, we train the model on its training set with 157k training samples, and evaluate on its test set. For the WMT14 En-De/De-En task, we train the model on the training set with 4.5M training samples, where `newstest2013` and `newstest2014` are used as the validation and test set respectively. As for the WMT16 En-Ro task which has 610k training pairs, we utilize `newsdev2016` and `newstest2016` as the validation and test set. For each dataset, we tokenize the sentences by Moses (Koehn et al., 2007) and segment each word into subwords using Byte-Pair Encoding (BPE) (Sennrich et al., 2015), resulting in a 32k vocabulary shared by source and target languages.

5.1.2 Model Settings

We strictly follow the previous works to set the configurations of models. Our model is based on the Transformer (Vaswani et al., 2017) architecture, with multi-head positional attention proposed in (Gu et al., 2017). We utilize the `small` Transformer ($d_{\text{model}} = d_{\text{hidden}} = 256$, $n_{\text{head}} = 4$) with 5-layer encoder and decoder for the IWSLT14 De-En task, and the `base` Transformer ($d_{\text{model}} = d_{\text{hidden}} = 512$, $n_{\text{layer}} = 6$, $n_{\text{head}} = 8$) for the WMT14 and WMT16 tasks. We set $n = 2$ for all tasks, i.e., we consider two-gram matchings when calculating L_{gram}. The hyper-parameters α_1 and α_2 are both set to 0.01 for all tasks.

5.1.3 Baselines

We consider seven recent works as our baselines, including five NAT works: NAT with fertility (NAT-FT) (Gu et al., 2017), NAT with Imitation Learning (Imitate-NAT) (Wei et al., 2019), NAT with Regularizations (NAT-Reg) (Wang et al., 2019), NAT with Curriculum Learning (FCL-NAT) (Guo et al., 2020), NAT with Dynamic Conditional Random Field (NAT-DCRF) (Sun et al., 2019); and two iterative decoding based works: NAT with Iterative Refinement (NAT-IR) (Lee et al., 2018) and Conditional Masked NAT (CM-NAT) (Ghazvininejad et al., 2019). The first five models are purely non-autoregressive, whose time complexities during inference are all $O(1)$. The other two models are based on iteratively refining the translation results by k iterations, where k is a constant number, yielding $O(k)$ complexity. In the experiments, we also compare with them in terms of the inference latency on clock.

5.1.4 Sequence-Level Knowledge Distillation

We adopt sequence-level knowledge distillation (Kim and Rush, 2016) on the training set of each task, which has been proved by previous NAT models that it can produce less noisy and more deterministic training data (Gu et al., 2017). As stated by Wang et al. (2019), the performance of the AT teacher will affect the final performance of the NAT student model. While AT teachers used in previous works have various performance, we utilize the teacher model which has similar performance with the one used in our main baseline CM-NAT (Ghazvininejad et al., 2019) to construct a fair comparison. In addition, we also provide the performance of our model trained by a weakened AT teacher (denoted as WT in Table 2) which has similar performance with the one used in (Wang et al., 2019) to compare with them.

5.1.5 Training and Inference

We train the model with 8/1 Nvidia 1080Ti GPUs on the WMT datasets and IWSLT14 dataset respectively, and we utilize the Adam optimizer while following the same settings used in the original Transformer. During inference, we generate multiple translation candidates by taking the top B length predictions into consideration, and select the translation with the highest probability as the final result. We set $B = 3$ on WMT tasks and $B = 4$ on IWSLT14 tasks. We also report the clock time of inference latency on a single Nvidia 1080Ti GPU in our experiments, where we set the batch size to 1 and calculate the average per sentence translation time on `newstest2014` for the WMT14 En-De task to keep consistence with previous works.

As for evaluation, we use BLEU scores (Papineni et al., 2002) as the evaluation metric, and

[1] https://wit3.fbk.eu/

[2] https://www.statmt.org/wmt16/translation-task

[3] https://www.statmt.org/wmt14/translation-task

Models	WMT14 En−De	WMT14 De−En	WMT16 En−Ro	WMT16 Ro−En	IWSLT14 De−En	Latency	Speedup
Transformer (Vaswani et al., 2017)	28.04*	32.69*	34.13*	34.46*	32.99*	607 ms	1.00×
Transformer (Weak Teacher)	27.40*	31.29*	/	/	/	–	–
NAT-FT (NPD 100) (Gu et al., 2017)	19.17	23.20	29.79	31.44	24.21†	257 ms	2.36×
Imitate-NAT (Wei et al., 2019)	24.15	27.28	31.45	31.81	/	/	/
NAT-Reg (NPD 9) (Wang et al., 2019)	24.61	28.90	/	/	28.04	40 ms	15.1×
FCL-NAT (NPD 9) (Guo et al., 2020)	25.75	29.50	/	/	29.91	38 ms	16.0×
NAT-DCRF (NPD 9) (Sun et al., 2019)	26.07	29.68	/	/	29.99	63 ms	9.63×
NAT-IR ($k = 5$) (Lee et al., 2018)	20.26	23.86	28.86	29.72	/	/	/
NAT-IR ($k = 10$)	21.61	25.48	29.32	30.19	23.94†	404† ms	1.50×
CM-NAT ($k = 4$) (Ghazvininejad et al., 2019)	25.94	29.90	32.53	33.23	30.42*	62* ms	9.79×
CM-NAT ($k = 10$)	27.03	30.53	33.08	33.31	31.71*	161* ms	3.77×
JM-NAT ($k = 4$)	27.05	31.51	32.97	33.21	31.27	45 ms	13.5×
JM-NAT ($k = 10$)	**27.69**	**32.24**	**33.52**	**33.72**	**32.59**	106 ms	5.73×
JM-NAT (WT) ($k = 4$)	26.82	30.59	/	/	/	–	–
JM-NAT (WT) ($k = 10$)	27.31	31.02	/	/	/	–	–

Table 2: The BLEU scores of our proposed JM-NAT and the baseline methods on the WMT14 En-De/De-En, WMT16 En-Ro/Ro-En and IWSLT14 De-En tasks. We report the best results for the baseline methods and also list the inference latency on clock as well as the speedup w.r.t autoregressive models. "†" indicates that the result is provided by (Wang et al., 2019), "$*$" indicates the results obtained by our implementation, "/" indicates the corresponding result is not reported in the original paper, and "–" indicates the same numbers as above. "Weak Teacher (WT)" indicates the NAT is trained with a weakened AT teacher through knowledge distillation. NPD stands for Noisy Parallel Decoding utilized in previous works. "k" represents the number of iterations while inference.

report the tokenized case-sensitive scores for the WMT datasets, as well as the tokenized case-insensitive scores for the IWSLT14 dataset. Our implementation is based on `fairseq` (Ott et al., 2019) and is avaliable at `https://github.com/lemmonation/jm-nat`.

5.2 Results

The main results are listed in Table 2. We denote our model as Jointly Masked NAT (JM-NAT), and show the results when the upper bound of iterations k is set to 4 and 10. As can be observed from Table 2, our model achieves comparable performance with its AT teacher on all datasets (only 0.5 BLEU score behind in average), while achieving 5+ times speedup on the inference latency. Compared with the pure NAT models with $O(1)$ time complexity, with similar inference latency by setting $k = 4$, our model outperforms all baselines with a consistent margin on different tasks. Compared with the models based on iterative refinement, JM-NAT also shows consistent superiority with the same time complexity. Our model outperforms CM-NAT (Ghazvininejad et al., 2019) with margins from 0.41 to 1.71 on different tasks, illustrating the boosted performance brought by the jointly masked model as well as the proposed loss functions. It is worth noting that CM-NAT utilizes a much stronger AT teacher on the WMT14 En-De task (using the `large` configuration of Transformer and achieving 28.65 BLEU score). Our model, even with less iterations or a weaker AT teacher, still outperforms CM-NAT in most cases, and it is straightforward to further improve our performance with a stronger teacher.

5.3 Analysis

5.3.1 Encoder Performance

As there does not exist a clear metric (such as the perplexity in language generation tasks) to evaluate the quality of the encoder in a sequence-to-sequence model, we adopt a naive version of the adversarial attack on text (Belinkov and Bisk, 2017) to the encoder input to test the robustness of the encoder. Specifically, during inference, we follow the same strategy used in Section 3 to add noise to the source sentence x. Given the noise ratio $\alpha \in (0, 1)$, we randomly select $\lfloor \alpha \cdot T_x \rfloor$ (where $\lfloor \cdot \rfloor$ stands for the rounding function) source tokens and either drop or replace them with other tokens in the vocabulary. We increase α from 0 to 10% and test the performance of each model on the validation set of the IWSLT14 De-En task, and show the results in Figure 3. We compare our model with baselines

NAT-FT	NAT-Reg	CM-NAT	JM-NAT
2.30	0.90	0.48	**0.17**

Table 3: The comparison on the average number of per-sentence repetitive tokens on the validation set of the IWSLT14 De-En task.

including NAT-FT and CM-NAT. According to the results, compared with CM-NAT, which is also an iterative decoding based method, our model shows more robust performance with regard to the noise on the encoder input, showing the efficacy of the proposed masking strategy and the better quality of our encoder.

5.3.2 Repetitive Words

As studied by Wang et al. (2019), the tendency of producing repetitive words in translation is a major drawback of NAT models. We propose to alleviate this problem by training the decoder with the consecutive masking strategy as well as the n-gram loss function. We compute the average number of consecutive repetitive tokens per sentence in the translation results on the validation set of the IWSLT14 De-En task. Results are shown in Table 3. Without introducing explicit regularizations (Wang et al., 2019), our method is still able to alleviate the problem of repetitive words. Compared with CM-NAT who also utilizes an iterative decoding method, the superiority of our method demonstrates the proposed consecutive masking strategy better solves the problem than random masking.

5.3.3 Ablation Study

We conduct the ablation study on the validation set of the IWSLT14 De-En task to illustrate the contribution of different components in our model. Results are shown in Table 4. For the encoder, both encoder masking and the objective function L_{enc} contribute to the final performance, and encoder masking provides the most prominent performance promotion. On the decoder side, both of the consecutive masking strategy and the n-gram loss function are indispensable to produce solid performance as they are tied together through the hyper-parameter n. In addition, all the proposed components are effective in alleviating the repetitive translations, and the n-gram loss function contributes the most.

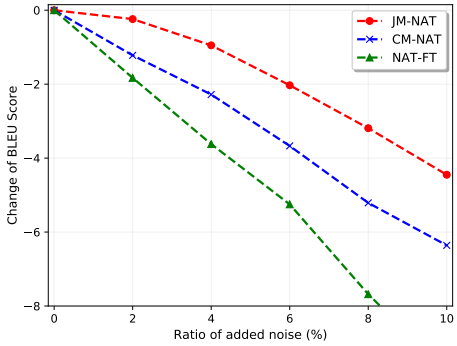


Figure 3: The performance of considered NAT models when adding noise to the encoder input. The X-axis indicates the ratio of noise, and the Y-axis indicates the ΔBLEU score compared with feeding the input without noise.

Model Variants	BLEU	ΔBLEU	Reps
JM-NAT	33.82	–	0.17
On the Encoder Side			
w/o L_{enc}	33.32	−0.50	0.21
w/o Encoder Mask & L_{enc}	32.15	−1.67	0.23
On the Decoder Side			
w/o L_{gram}	33.27	−0.55	0.30
w/o Consecutive Mask	32.97	−0.85	0.25

Table 4: The ablation study on different components of the proposed model conducted on the validation set of IWSTL14 De-En task. "Reps" indicates the average number of repetitive translations computed same as in Table 3.

5.4 Case Study

We further conduct case studies to intuitively demonstrate the performance of different models and the generation process of our model. Results are listed in Table 5. As we discussed in Section 1, repetitive translations and missing translations are two stubborn problems of NAT models. In Table 5, both NAT-FT and CM-NAT tend to generate repetitive words (such as "eliminate diabetes diabetes" and "reduce cancer risk risk") as well as incomplete translations (both of them miss the word "eliminate" in the second clause), while our model achieves better results.

6 Conclusion

In this paper, we propose a jointly masked sequence-to-sequence model for non-autoregressive neural machine translation. We first empirically investigate the functionalities of the non-autoregressive translation model, and

Source:	was wäre , wenn sie die genetischen veränderungen machen könnten , um diabetes oder alzheimer zu beseitigen oder das reduzieren des krebsrisikos oder schlaganfälle zu eliminieren ?
Target:	what if you could make the genetic changes to eliminate diabetes or alzheimer 's or reduce the risk of cancer or eliminate stroke ?
NAT-FT:	what if you could make the genetic changes in order to eliminate ***diabetes diabetes*** or alzheimer disease or ***reduce reduce*** the ***cancer of cancer*** or strostroke ?
CM-NAT:	what if you could make the genetic changes to eliminate diabetes or ***alzheimer alzheimer*** 's , or reduce cancer ***risk risk*** or ***stro stro*** dents ?
JM-NAT:	what if you could make the genetic changes to eliminate diabetes or alzheimer 's disease or the reduce cancer risk or eliminate stroke ?

Table 5: A case study on the translation results of different models on the IWSLT14 De-En task. We set $k = 10$ for our model. The bold italics represent the repetitive words in the translation results.

improve the training of the encoder by masking its input and introducing a prediction based loss function. For the decoder, we propose to utilize consecutive masking and introduce an n-gram based loss function to alleviate the problem of repetitive translations. Our model outperforms all compared NAT baselines and achieves comparable performance with autoregressive models on five benchmark tasks with $5+$ times speed up on the inference latency.

In the future, we will extend the investigation on the functionalities of the encoder and decoder to other sequence-to-sequence tasks such as text summarization and text style transfer to explore more applications of our model.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (No. 61673364, No. U1605251) and the Fundamental Research Funds for the Central Universities (WK2150110008). We would like to thank the Information Science Laboratory Center of USTC for the hardware and software services. We thank the anonymous reviewers as well as Zhirui Zhang and Tianyu He for helpful feedback on early versions of this work.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 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 (EMNLP-IJCNLP)*, pages 6114–6123.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2017. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*.

Jiatao Gu, Changhan Wang, and Jake Zhao. 2019. Levenshtein transformer. *arXiv preprint arXiv:1905.11006*.

Junliang Guo, Xu Tan, Di He, Tao Qin, Linli Xu, and Tie-Yan Liu. 2019. Non-autoregressive neural machine translation with enhanced decoder input. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Junliang Guo, Xu Tan, Linli Xu, Tao Qin, Enhong Chen, and Tie-Yan Liu. 2020. Fine-tuning by curriculum learning for non-autoregressive neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt,

William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Tianyu He, Xu Tan, and Tao Qin. 2019. Hard but robust, easy but sensitive: How encoder and decoder perform in neural machine translation. *arXiv preprint arXiv:1908.06259*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. *arXiv preprint arXiv:1802.06901*.

Zhuohan Li, Zi Lin, Di He, Fei Tian, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Hint-based training for non-autoregressive translation. *arXiv preprint arXiv:1909.06708*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Shuming Ma, Xu Sun, Yizhong Wang, and Junyang Lin. 2018. Bag-of-words as target for neural machine translation. *arXiv preprint arXiv:1805.04871*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Chenze Shao, Yang Feng, and Xilin Chen. 2018. Greedy search with probabilistic n-gram matching for neural machine translation. *arXiv preprint arXiv:1809.03132*.

Chenze Shao, Jinchao Zhang, Yang Feng, Fandong Meng, and Jie Zhou. 2019. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. *arXiv preprint arXiv:1911.09320*.

Xin Sheng, Linli Xu, Junliang Guo, Jingchang Liu, Ruoyu Zhao, and Yinlong Xu. 2020. Introvnmt: An introspective model for variational neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. *arXiv preprint arXiv:1902.03249*.

Zhiqing Sun, Zhuohan Li, Haoqing Wang, Di He, Zi Lin, and Zhihong Deng. 2019. Fast structured decoding for sequence models. In *Advances in Neural Information Processing Systems*, pages 3011–3020.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Non-autoregressive machine translation with auxiliary regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Bingzhen Wei, Mingxuan Wang, Hao Zhou, Junyang Lin, and Xu Sun. 2019. Imitation learning for non-autoregressive neural machine translation. *arXiv preprint arXiv:1906.02041*.

Sean Welleck, Kianté Brantley, Hal Daumé III, and Kyunghyun Cho. 2019. Non-monotonic sequential text generation. *arXiv preprint arXiv:1902.02192*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.