Guiding Non-Autoregressive Neural Machine Translation Decoding with Reordering Information

Qiu Ran*, Yankai Lin*, Peng Li*, Jie Zhou
Pattern Recognition Center, WeChat AI, Tencent Inc., China
{soulcaptran,yankailin,patrickpli,withtomzhou}@tencent.com

Abstract

Non-autoregressive neural machine translation (NAT) generates each target word in parallel and has achieved promising inference acceleration. However, existing NAT models still have a big gap in translation quality compared to autoregressive neural machine translation models due to the multimodality problem: the target words may come from multiple feasible translations. To address this problem, we propose a novel NAT framework ReorderNAT which explicitly models the reordering information to guide the decoding of NAT. Specially, ReorderNAT utilizes deterministic and non-deterministic decoding strategies that leverage reordering information as a proxy for the final translation to encourage the decoder to choose words belonging to the same translation. Experimental results on various widely-used datasets show that our proposed model achieves better performance compared to most existing NAT models, and even achieves comparable translation quality as autoregressive translation models with a significant speedup.

Introduction

Neural machine translation (NMT) models with encoder-decoder framework (Sutskever, Vinyals, and Le 2014; Bahdanau, Cho, and Bengio 2014) significantly outperform conventional statistical machine translation models (Koehn, Och, and Marcu 2003; Koehn et al. 2007). Despite their success, the state-of-the-art NMT models usually suffer from the slow inference speed, which has become a bottleneck to apply NMT in real-world translation systems. The slow inference speed of NMT models is due to their autoregressive property, i.e., decoding the target sentence word-by-word according to the translation history.

Recently, Gu et al. (2018) introduced non-autoregressive NMT (NAT) which can simultaneously decode all target words to break the bottleneck of the autoregressive NMT (AT) models. To this end, NAT models (Gu et al. 2018; Wei et al. 2019; Wang et al. 2019; Guo et al. 2019a) usually directly copy the source word representations to the input of the decoder, instead of using previous predicted target word representations. Hence, the inference of different target words are independent, which enables parallel computation of the decoder in NAT models. NAT models could achieve 10-15 times speedup compared to AT models while maintaining considerable translation quality.

However, NAT models still suffer from the multimodality problem (Gu et al. 2018): it discards the dependencies among the target words, and therefore the target words may be chosen from multiple feasible translations, resulting in duplicate, missing or even wrong words. For example, the German phrase "*Vielen Dank*" can be translated as both "*thank you*" and "*many thanks*". Unfortunately, as each target word is generated independently, "*thank thanks*" and "*many you*" may also be assigned high probabilities, resulting in inferior translation quality. In this work, we argue reordering information is essential for NAT models and helpful for alleviating the multimodality problem.

To this end, we propose a novel NAT framework named ReorderNAT in this work, which explicitly models the reordering information to guide the decoding of NAT. To be specific, as shown in Figure 1, ReorderNAT first reorders the source sentence into a pseudo-translation formed by source words but in the target language word order, and then translates the source sentence conditioned on it. We further introduce two guiding decoding strategies which utilize the reordering information (i.e. pseudo-translation) to guide the word selection in decoding. The first one is *deterministic guiding decoding* which first generates a most likely pseudo-translation and then generates the target sentence based on it. The second one is *non-deterministic guiding decoding* which utilizes the conditional distribution of the pseudo-translation as a latent variable to guide the decoding of target sentences.

Ideally, the pseudo-translation can be viewed as a final translation written in source language. Guiding decoding with it could help to model the conditional dependencies of the target words and encourage the decoder to choose words belonging to the same translation, which naturally reduces the multimodality problem. Moreover, the decoding space of generating pseudo-translation is limited to the permutation of words in the source sentence, which can be well modeled by a small model. Therefore, ReorderNAT could effectively alleviate the multimodality problem by introducing the reordering information in NAT.

Experimental results on several widely-used benchmarks

*indicates equal contribution

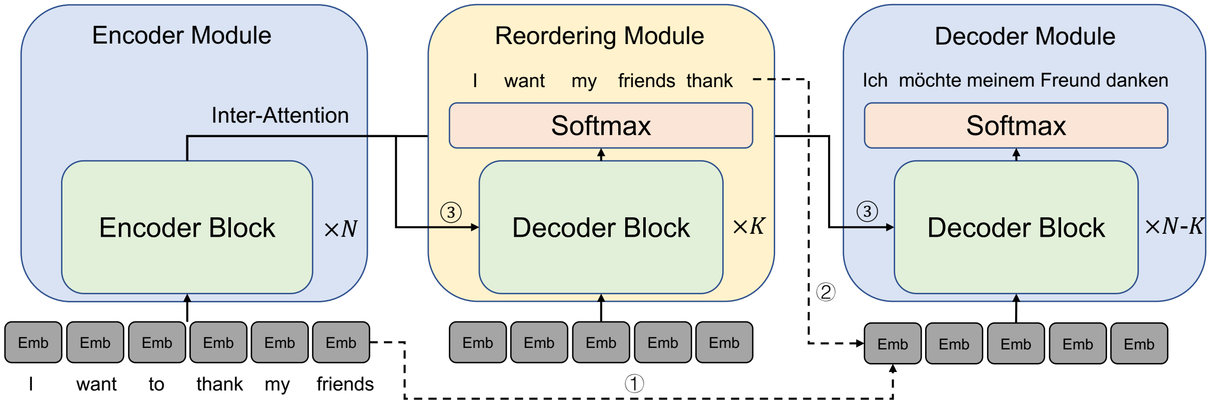

Figure 1: The architecture of our ReorderNAT model. Different from original NAT models, our model adds a reordering module between the encoder and decoder modules to explicitly model the reordering information. For original NAT models, the decoder inputs are the copied embeddings of source sentence (No.1 dashed arrow), and for our ReorderNAT model, the decoder inputs are the embeddings of pseudo-translation generated by reordering module (No. 2 dashed arrow). The encoder and decoder blocks are the same as existing NMT models (e.g., Transformer block).

show that our proposed ReorderNAT model achieves significant and consistent improvements compared to existing NAT models by explicitly modeling the reordering information to guide the decoding. Moreover, by introducing a simple but effective AT module to model reordering information, our ReorderNAT immensely narrows the translation quality gap between AT and NAT models, while maintaining considerable speedup (nearly six times faster). The source codes are available at https://github.com/ranqiu92/ReorderNAT.

Background

Non-autoregressive neural machine translation (NAT) is first proposed by Gu et al. (2018) to alleviate the slow decoding issue of autoregressive neural machine translation (AT) models, which could simultaneously generate target words by removing their dependencies. Formally, given a source sentence $\mathbf{X} = \{x_1, \cdots, x_n\}$ and a target sentence $\mathbf{Y} = \{y_1, \cdots, y_m\}$, NAT models the translation probability from $\mathbf{X}$ to $\mathbf{Y}$ as a product of conditionally independent target word probability:

$$P(\mathbf{Y}|\mathbf{X}) = \prod_{i=1}^{m} P(y_i|\mathbf{X}). \tag{1}$$

Instead of utilizing the translation history, NAT models usually copy source word representations as the input of the decoder. Hence, when translating a sentence, NAT models could predict all target words with their maximum likelihood individually by breaking the dependency among them, and therefore the decoding procedure of NAT models is in parallel and has very low translation latency.

However, since NAT models discard the sequential dependencies among words in the target sentence, they suffer from the potential performance degradation due to the multimodality problem. To be specific, a source sentence may have multiple translations. During decoding, NAT models may choose the target words from different translations, resulting in duplicate, missing or even wrong words. Consequently, NAT models cannot effectively learn the intricate translation patterns from source sentences to target sentences, leading to inferior translation quality.

Methodology

In this section, we introduce a novel NAT model named ReorderNAT, which aims to alleviate the multimodality problem in NAT models.

ReorderNAT

As shown in Figure 1, ReorderNAT employs a reordering module to explicitly model the reordering information in the decoding[1]. Formally, ReorderNAT first employs the reordering module to translate the source sentence $\mathbf{X}$ into a pseudo-translation $\mathbf{Z} = \{z_1, \cdots, z_m\}$ which reorganizes source sentence structure into the target language, and then uses the decoder module to generate target translation $\mathbf{Y}$ based on the pseudo-translation. ReorderNAT models the overall translation probability as:

$$P(\mathbf{Y}|\mathbf{X}) = \sum_{\mathbf{Z}} P(\mathbf{Y}|\mathbf{Z}, \mathbf{X}) P(\mathbf{Z}|\mathbf{X}), \tag{2}$$

where $P(\mathbf{Z}|\mathbf{X})$ is modeled by the reordering module and $P(\mathbf{Y}|\mathbf{Z}, \mathbf{X})$ is modeled by the decoder module. Next, we will introduce the reordering and decoder modules in detail[2].

Reordering Module The reordering module determines the source-side information of each target word by learning to translate the source sentence into a pseudo-translation.

[1]We do not employ positional attention (Gu et al. 2018) as the mechanism may be misguided by target supervision due to the indirect optimization and lead to inferior translation.

[2]The encoder module of ReorderNAT is a multi-layer Transformer (Vaswani et al. 2017), which is the same as original NAT models.

We propose two feasible implementations of the reordering module:

(1) **NAT Reordering Module**: Intuitively, the pseudo-translation probability can also be modeled as NAT:

$$P(\mathbf{Z}|\mathbf{X}) = \prod_{i=1}^{m} P(z_i|\mathbf{X}), \tag{3}$$

where $P(z_i|\mathbf{X})$ is calculated by a single-layer Transformer. During inference, the NAT reordering module needs to determine the length of the pseudo-translation. To this end, we use a length predictor and copy the embeddings of the source sentence as the input of the reordering module similar to existing NAT models.

(2) **AT Reordering Module**: We find that AT models are more suitable for modeling the reordering information compared to NAT models, and even a light AT model with similar decoding speed to a large NAT model could achieve better performance in modeling reordering information. Hence, we also introduce a light AT model to model the pseudo-translation probability as:

$$P(\mathbf{Z}|\mathbf{X}) = \prod_{i=1}^{m} P(z_i|\mathbf{z}_{<i}, \mathbf{X}), \tag{4}$$

where $\mathbf{z}_{<i} = \{z_1, \cdots, z_{i-1}\}$ indicates the pseudo-translation history, and $P(z_i|\mathbf{z}_{<i}, \mathbf{X})$ is calculated by a single-layer recurrent neural network.

Decoder Module The decoder module translates the source sentence into the target translation with the guiding of pseudo-translation, which regards the translation of each word as NAT:

$$P(\mathbf{Y}|\mathbf{Z}, \mathbf{X}) = \prod_{i=1}^{m} P(y_i|\mathbf{Z}, \mathbf{X}). \tag{5}$$

As shown in Figure 1, the encoder module and the decoder module can be viewed as a seq-to-seq model which translates the source sentence to target sentence. Different from original NAT, the input of our decoder module is the embeddings of pseudo-translation instead of copied embeddings of source sentence, which is used to guide the word selection. Note that the encoder outputs are also fed into the decoder attention module, which can help alleviate the reordering errors.

To make the model parameter number comparable with the baseline model, we use K and $N-K$ decoder blocks for the reordering and decoder modules respectively [3].

Guiding Decoding Strategy

ReorderNAT explicitly models reordering information of NAT and aims to utilize it to alleviate the multimodality problem. Now the remaining problem is how to perform decoding with the guide of reordering information. We propose to utilize the pseudo-translation as a bridge to guide the decoding of the target sentence, which can be formulated as:

$$\begin{aligned}\mathbf{Y}^* &= \arg\max_{\mathbf{Y}} P(\mathbf{Y}|\mathbf{X}) \\ &= \arg\max_{\mathbf{Y}} \sum_{\mathbf{Z}} P(\mathbf{Y}|\mathbf{Z}, \mathbf{X}) P(\mathbf{Z}|\mathbf{X}). \end{aligned} \tag{6}$$

It is intractable to obtain an exact solution for maximizing Eq. 6 due to the high time complexity. Inspired by the pre-ordering works in statistical machine translation, we propose a **deterministic guiding decoding (DGD)** strategy and a **non-deterministic guiding decoding (NDGD)** strategy to solve this problem.

The DGD strategy first generates the most probable pseudo-translation of the source sentence, and then translates the source sentence conditioned on it:

$$\mathbf{Z}^* = \arg\max_{\mathbf{Z}} P(\mathbf{Z}|\mathbf{X}), \tag{7}$$

$$\mathbf{Y}^* = \arg\max_{\mathbf{Y}} P(\mathbf{Y}|\mathbf{Z}^*, \mathbf{X}). \tag{8}$$

The DGD strategy is simple and effective, but the hard approximation may bring in some noises.

Different from the DGD strategy which utilizes a deterministic pseudo-translation, the NDGD strategy regards the probability distribution $\mathcal{Q}$ of words to generate the most probable pseudo-translations as a latent variable, and models the translation as generating the target sentence according to $\mathcal{Q}$, i.e., Eq. 8 is re-formulated as:

$$\mathbf{Y}^* = \arg\max_{\mathbf{Y}} P(\mathbf{Y}|\mathcal{Q}, \mathbf{X}), \tag{9}$$

where $\mathcal{Q}$ is defined as:

$$\mathcal{Q}(z_i) = P(z_i|\mathbf{z}^*_{<i}, \mathbf{X}) = \frac{\exp\big(s(z_i)/T\big)}{\sum_{z'_i} \exp\big(s(z'_i)/T\big)}, \tag{10}$$

where $s(\cdot)$ is a score function of pseudo-translation (the input of softmax layer in the decoder) and T is a temperature coefficient. Since $\mathcal{Q}$ can be viewed as a non-deterministic form of the pseudo-translation, the translation with the NDGD strategy is also guided by the pseudo-translation.

To be specific, as shown in Figure 1, the major difference between DGD and NDGD strategy is the inputs of decoder module (No. 2 dashed arrow), where the DGD strategy directly utilizes the word embeddings of generated pseudo-translation and the NDGD strategy utilizes the word embeddings weighted by the word probability of pseudo-translation.

Discussion

In ReorderNAT, the decoding space of generating pseudo-translation with reordering module is much smaller than that of the whole translation in NAT since the decoding vocabulary is limited to the words in the source sentence. The reordering module is more likely to be guided to one pseudo-translation among multiple alternatives. Therefore, ReorderNAT could easily capture the reordering information compared to the original NAT by explicitly modeling

[3] We set K to 1 for an AT module while $N-1$ for an NAT module as it is more difficult for an NAT module to model the reordering information (see Experiments).

with pseudo-translation as internal supervision. Besides, the candidates of the i-th word of the final translation can be narrowed to the translations of z_i to some extent since z_i is the i-th word in the pseudo-translation which indicates the corresponding source information of y_i. In other words, pseudo-translations could be viewed as a translation written in source language which helps the decoder to capture the dependencies among target words and choose words belonging to the same translation.

Training

In the training process, for each training sentence pair $(\mathbf{X}, \mathbf{Y}) \in D$, where D is the training set, we first generate its corresponding pseudo-translation $\hat{\mathbf{Z}}$. And then ReorderNAT is optimized by maximizing a joint loss:

$$\mathcal{L} = \mathcal{L}_R + \mathcal{L}_T, \tag{11}$$

where $\mathcal{L}_R$ and $\mathcal{L}_T$ indicate the reordering and translation losses respectively. Formally, for both DGD and NDGD, the reordering loss $\mathcal{L}_R$ is defined as[4] :

$$\mathcal{L}_R = \sum_{(\mathbf{X},\hat{\mathbf{Z}},\mathbf{Y})\in D} \log P(\hat{\mathbf{Z}}|\mathbf{X}). \tag{12}$$

For the DGD approach, the translation loss is defined as an overall maximum likelihood of translating pseudo-translation into the target sentence:

$$\mathcal{L}_T = \sum_{(\mathbf{X},\hat{\mathbf{Z}},\mathbf{Y})\in D} \log P(\mathbf{Y}|\hat{\mathbf{Z}}, \mathbf{X}), \tag{13}$$

For the NDGD approach, the translation loss is defined as an overall maximum likelihood of decoding the target sentence from the conditional probability of pseudo-translation:

$$\mathcal{L}_T = \sum_{(\mathbf{X},\hat{\mathbf{Z}},\mathbf{Y})\in D} \log P(\mathbf{Y}|\mathcal{Q}, \mathbf{X}). \tag{14}$$

In particular, we use the trained model for the DGD approach to initialize the model for the NDGD approach since if $\mathcal{Q}$ is not well trained, $\mathcal{L}_T$ will converge very slowly.

Experiments

Datasets

The main experiments are conducted on three widely-used machine translation tasks: WMT14 En-De (4.5M pairs), WMT16 En-Ro (610k pairs) and IWSLT16 En-De (196k pairs)[5]. For WMT14 En-De task, we take newstest-2013 and newstest-2014 as validation and test sets respectively. For WMT16 En-Ro task, we employ newsdev-2016 and newstest-2016 as validation and test sets respectively. For IWSLT16 En-De task, we use test2013 for validation.

We also conduct our experiments on Chinese-English translation which differs more in word order. The training set consists of 1.25M sentence pairs extracted from the LDC corpora. We use NIST 2002 (MT02) as validation set, and NIST 2003 (MT03), 2004 (MT04), 2005 (MT05) as test sets.

[4]Note that since $\mathcal{Q}(\mathbf{Z}) = P(\mathbf{Z}|\mathbf{X})$, the reordering loss could also learn $\mathcal{Q}$ for the NDGD approach.

[5]We use the prepossessed corpus provided by Lee, Mansimov, and Cho (2018) at https://github.com/nyu-dl/dl4mt-nonauto/tree/multigpu.

Experimental Settings

We use the fast_align tool[6] to generate the pseudo-translation in our experiments. We follow most of the model hyperparameter settings in (Gu et al. 2018; Lee, Mansimov, and Cho 2018; Wei et al. 2019) for fair comparison. For IWSLT16 En-De, we use a 5-layer Transformer model ($d_{model} = 278$, $d_{hidden} = 507$, $n_{head} = 2$, $p_{dropout} = 0.1$) and anneal the learning rate linearly (from 3×10^{-4} to 10^{-5}) as in (Lee, Mansimov, and Cho 2018). For WMT14 En-De, WMT16 En-Ro and Chinese-English translation, we use a 6-layer Transformer model ($d_{model} = 512$, $d_{hidden} = 512$, $n_{head} = 8$, $p_{dropout} = 0.1$) and adopt the warm-up learning rate schedule (Vaswani et al. 2017) with $t_{warmup} = 4000$. For the GRU reordering module, we set it to have the same hidden size with the Transformer model in each dataset. We employ label smoothing of value $\epsilon_{ls} = 0.15$ and utilize the sequence-level knowledge distillation (Kim and Rush 2016). For each dataset, we select the optimal guiding decoding strategy according to the model performance on validation sets. We also set T in Eq. 10 to 0.2 according to a grid search on the validation set.We measure the model inference speedup on the validation set of IWSLT16 En-De task with a NVIDIA P40 GPU and set batch size to 1.

Baselines

In the experiments, we compare ReorderNAT (NAT) and ReorderNAT (AT) which utilize an NAT and an AT reordering modules respectively with several baselines.

We select three models as our autoregressive baselines: (1) **Transformer**$_{full}$ (Vaswani et al. 2017), which is the teacher model used in the knowledge distillation and of which the hyperparameters are described in experimental settings. (2) **Transformer**$_{one}$, a lighter version of Transformer, of which the decoder layer number is 1. (3) **Transformer**$_{gru}$, which replaces the decoder of Transformer$_{full}$ with a single-layer GRU (Cho et al. 2014). And we set the beam size to 4 in the experiments.

Besides, we compare with several typical NAT models, which also have the ability to alleviate the multimodality problem and are highly relevant to our work: (1) **NAT-IR** (Lee, Mansimov, and Cho 2018), which adopts an iterative refinement mechanism enabling the model to perform inference based on surrounding words in the translation; (2) **NAT-FS** (Shao et al. 2019a), which introduces the autoregressive property to the top decoder layer of NAT; (3) **FlowSeq-base** (Ma et al. 2019), which uses generative flow to help model dependencies within target sentences. For fair comparison, We use the "base" version as it has comparable model size with our model; (4) **imitate-NAT** (Wei et al. 2019), which imitates the behavior of an AT model. (5) **CMLM-small** (Ghazvininejad et al. 2019), which is built on a conditional masked language model and also iteratively refines the translation. We use the "small" version for fair comparison; (6) **NART-DCRF** (Sun et al. 2019), which uses CRF to capture the word dependencies; (7) **LevT** (Gu, Wang, and Zhao 2019), which models the sequence generation as multi-step insertion and deletion operations.

[6]https://github.com/clab/fast_align

Model	Multi-Step	WMT14 En→De	WMT14 De→En	WMT16 En→Ro	WMT16 Ro→En	IWSLT16 En→De	Speedup
Autoregressive Models							
Transformer$_{full}$	-	27.17	31.95	32.86	32.60	31.18	1.00×
Transformer$_{one}$	-	25.52	29.31	30.61	31.23	29.52	2.42×
Transformer$_{gru}$	-	26.27	30.62	30.41	31.23	29.26	3.10×
Non-Autoregressive Models							
NAT-IR (iter=1)	-	13.91	16.77	24.45	25.73	22.20	8.98×
NAT-IR (iter=10)	√	21.61	25.48	29.32	30.19	27.11	1.55×
NAT-FS	-	22.27	27.25	30.57	30.83	27.78	3.38×
FlowSeq-base	-	21.45	26.16	29.34	30.44	-	<1.5×
FlowSeq-base+NPD (s=30)	-	23.48	28.40	<u>31.75</u>	<u>32.49</u>	-	<1.5×
imitate-NAT	-	22.44	25.67	28.61	28.90	28.41	18.6×
imitate-NAT+LPD (s=7)	-	24.15	27.28	31.45	31.81	**30.68**	9.70×
CMLM-small (iter=10)	√	25.51	29.47	31.65	<u>32.27</u>	-	<1.5×
NART-DCRF	-	23.44	27.22	-	-	-	10.4×
NART-DCRF+LPD (s=19)	-	<u>26.80</u>	30.04	-	-	-	4.39×
LevT	√	<u>27.27</u>	-	-	<u>33.26</u>	-	4.01×
Our Models							
ReorderNAT (NAT)	-	22.79	27.28	29.30	29.50	25.29	16.11×
ReorderNAT (NAT)+LPD (s=7)	-	24.74	29.11	31.16	31.44	27.40	7.40×
ReorderNAT (AT)	-	**26.49**	**31.13**	**31.70**	**31.99**	30.26	5.96×

Table 1: Overall results of AT and NAT models in BLEU score on the test sets of WMT14 and WMT16, and validation set of IWSLT16. "DCRF" denotes a CRF layer with dynamic transition (Sun et al. 2019). "NPD" denotes noisy parallel decoding (Gu et al. 2018), "LPD" denotes length parallel decoding (Wei et al. 2019), and "s" denotes sample size. "iter" denotes translation refinement iterations. Better BLEU scores with *low speedup* are underlined.

Overall Results

We compare ReorderNAT (NAT) and ReorderNAT (AT) that utilize an NAT reordering module and an AT reordering module respectively with all baseline models. All the results are shown in Table 1. From the table, we can find that:

(1) Excluding six better BLEU scores with significant low speedup, ReorderNAT (AT) achieves the best performance on most of the benchmark datasets, which is even close to the AT model with smaller than 1 BLEU gap (26.49 vs. 27.17 on WMT14 En→De task, 31.99 vs. 32.60 on WMT16 Ro→En task, 30.26 vs. 31.18 on IWSLT16 En→De task). It is also worth mentioning that although ReorderNAT utilizes a small AT model to better capture reordering information, it could still maintain low translation latency (about 16× speedup for ReorderNAT (NAT) and 6× speedup for ReorderNAT (AT)). Compared to Transformer$_{one}$ and Transformer$_{gru}$, ReorderNAT (AT) uses a much smaller vocabulary in the AT reordering module, which is limited to the words in the source sentence and makes it faster. Besides, unlike NAT-IR, CMLM-small and LevT, our model can decode all target words in parallel without multiple iterations, which helps maintain the efficiency.

(2) ReorderNAT (NAT) and ReorderNAT (NAT)+LPD also gain significant improvements compared to most existing NAT models. It verifies the reordering information modeled by ReorderNAT could effectively guide its word selection.

(3) A small AT model with close latency to large NAT models could perform much better in modeling reordering information[7]. On all benchmark datasets, ReorderNAT (AT) with small AT GRU reordering module achieves much better translation quality than that with large NAT model (2-5 BLEU scores). Moreover, we find that the AT model Transformer$_{one}$ and Transformer$_{gru}$ with a single-layer AT Transformer or GRU for decoding could also outperform most existing NAT models while maintaining acceptable latency (2.42× and 3.10× speedup respectively). The reason is that a major potential performance degradation of NAT models compared to AT models comes from the difficulty of modeling the word order difference between source and target language, i.e., reordering information, which is neglected by most existing NAT models but can be well modeled by the small AT module[8].

Results on Chinese-English Translation

To show the effectiveness of modeling reordering information in NAT, we compare ReorderNAT with baselines on Chinese-English translation since the word order between Chinese and English is more different than that between German and English (En-De). From Table 2, we can find that

[7]The decoding speed of ReorderNAT (AT) is limited by the autoregressive property of the reordering module, which is the main drawback of our model. How to further improve its speed is a future direction we would like to pursue.

[8]We conduct experiments and find that our model outperforms the AT model by a big margin when replacing the predicted pseudo-translation with the ground-truth ones. This also indicates the main multimodality problem on NAT comes from the difficulty of modeling the reordering information.

Model	MT02*	MT03	MT04	MT05
Autoregressive Models				
Transformer$_{full}$	46.11	43.74	45.59	44.11
Transformer$_{one}$	43.60	41.24	43.39	41.62
Transformer$_{gru}$	43.68	40.55	43.02	40.73
Non-Autoregressive Models				
imitate-NAT	33.77	32.29	34.83	31.96
ReorderNAT (NAT)	37.99	36.03	38.17	36.07
ReorderNAT (AT)	**45.22**	**43.20**	**44.89**	**43.45**

Table 2: BLEU scores on Chinese-English translation. * indicates the validation set.

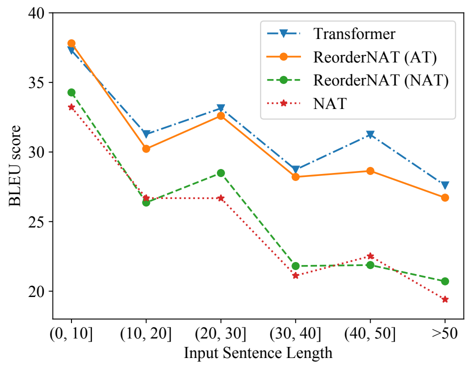

Figure 2: Translation quality on the IWSLT16 validation set over various input sentence lengths.

in Chinese-English translation, ReorderNAT (AT) achieves much more improvements (6-7 BLEU points) compared to ReorderNAT (NAT) and imitate-NAT. The reason is that more different word order in Chinese-English translation makes the decoding search space more complicated, which could be effectively alleviated by ReorderNAT.

Translation Quality v.s. Sentence Lengths

Figure 2 shows the BLEU scores of translations generated by the Transformer (AT model), the NAT model (ReorderNAT without the reordering module), ReorderNAT (NAT) and ReorderNAT (AT) on the IWSLT16 validation set with respect to input sentence lengths. We can observe that:

(1) ReorderNAT (NAT) and ReorderNAT (AT) achieve significant improvement compared to the NAT model for most lengths and ReorderNAT (AT) achieves nearly comparable performance to Transformer. It verifies the reordering information modeled by ReorderNAT could effectively help word selection and improve the translation quality.

(2) ReorderNAT (AT) achieves much better translation performance than the NAT model for sentences longer than 20 words, of which word order tends to be more different. Together with the results on Chinese-English translation (Table 2), we can conclude that NAT is weak on word reordering and our model is more effective especially when word order is more different.

Model	BLEU	RIBES	Dup	Mis
Transformer$_{full}$	31.18	83.74	-	-
NAT	24.57	82.21	50.09	9.09
ReorderNAT (NAT)	25.29	82.35	37.52	7.35
ReorderNAT (NAT)+LPD	27.04	83.21	24.31	5.59
ReorderNAT (AT)	30.26	83.55	2.84	0.52

Table 3: Relative increment of duplicate ("Dup") and missing ("Mis") token ratios on the IWSLT16 validation set. Smaller is better.

Multimodality Related Error Reduction

In this section, we investigate how our reordering module reduces the multimodality errors in NAT. Specially, we evaluate the RIBES (Isozaki et al. 2010) score and the reduction of duplicate and missing words (two most typical multimodality related errors). The results are shown in Table 3, where "Dup" and "Mis" denote the relative increment of duplicate and missing token ratios compared with the Transformer$_{full}$ model respectively[9] , and NAT is ReorderNAT without the reordering module. From the table, we can observe that:

(1) Our three ReorderNAT models achieve higher RIBES scores than the NAT model, validating our reordering module can help capture the word order difference between source and target languages. Moreover, the ReorderNAT (AT) model performs the best in RIBES, indicating the AT reordering module can model reordering information more effectively than the NAT reordering module.

(2) Compared with the NAT model, both Dup and Mis are significantly better for the three ReorderNAT models, indicating ReorderNAT is effective for alleviating the multimodality problem.

Case Study

Table 4 shows example translations of the NAT model, ReorderNAT (NAT) and ReorderNAT (AT). We find the problem of missing and duplicate words are severe in the translation (both 5 occurrences) of the NAT model, while this problem is effectively alleviated by ReorderNAT. Moreover, we find that most of the missing, duplicate or wrong words in the translation of our two ReorderNAT models come from the errors in the pseudo-translation, demonstrating that NAT models could well translate the pseudo-translation which is in the the target language word order, and the remaining problem of NAT lies on modeling reordering information.

Related Work

Non-Autoregressive Neural Machine Translation

Gu et al. (2018) first proposed the non-autoregressive neural machine translation (NAT), which enables parallel decoding for neural machine translation (NMT) and significantly accelerates the inference of NMT. However, its performance

[9]The formal definition of metrics "Dup/Mis" can be found in Ran et al. (2020).

Source		eventually , after a period of six months of brutal war and a toll rate of almost 50,000 dead , we managed to liber_ate our country and to t_opp_le the ty_rant
Reference		schließlich , nach einem Zeitraum von sechs Monaten bru_talen Krieges und fast 50.000 Toten , gelang es uns , unser Land zu befreien und den Tyran_nen zu stürzen .
NAT	Translation	schließlich , nach einer [ ] von sechs Monaten bru_bru_[ ] Krieg und einer Z_rate fast 50.000 50.000 [ ] , schafften wir es geschafft , unser Land [ ] befreien befreien und den Ty_r_ann [ ] ann entgegen_entgegen_deln .
ReorderNAT (NAT)	Pseudo-Translation	eventually , after a period of six [ ] brutal brutal war and a toll of almost 50,000 dead , managed we managed managed, [ ] country country to liber_[ ] and the ty_ty_rant rant opp_opp_opp_.
	Translation	schließlich , nach einer Zeit von sechs [ ] bru_talen Krieges und einer Z_von fast 50.000 Toten , schafften wir es geschafft , unser Land zu befreien und den Ty_r_r_ten zu_zu_ieren .
ReorderNAT (AT)	Pseudo-Translation	eventually , after a period of six months brutal brutal war and a toll toll rate of almost 50,000 dead , managed we managed , our country to liber_[ ] and the ty_ty_rant to liber_.
	Translation	schließlich , nach einer Zeitraum von sechs Monaten bru_talen Krieg und einer Z_oll_rate von fast 50.000 Toten , schafften wir es , unser Land zu befreien und den Ty_r_ann zu reparieren .

Table 4: Translation examples of NAT baseline and ReorderNAT. We use "[ ]" to denote missing words and underline the wrong words. We use _ to concatenate sub-words.

degrades greatly since it discards the sequential dependencies among target words. Recently, a variety of works have been investigated to improve its performance including (Guo et al. 2019a; Bao et al. 2019) which enhance the representation of decoder with source information; (Libovický and Helcl 2018; Shao et al. 2019a,b; Ghazvininejad et al. 2020) which optimize models with respect to sequence-level loss functions; (Wang et al. 2019; Li et al. 2020) which attempt to solve the multimodality problem using regularization or speical decoding strategies; (Li et al. 2019; Wei et al. 2019; Guo et al. 2019b; Liu et al. 2020; Sun and Yang 2020) which use AT models to guide the learning of NAT models; (Kaiser et al. 2018; Akoury, Krishna, and Iyyer 2019; Lee, Shu, and Cho 2020) which introduce latent variables to guide the decoding process of NAT models; and (Wang, Zhang, and Chen 2018; Lee, Mansimov, and Cho 2018; Gu, Wang, and Zhao 2019; Stern et al. 2019; Ghazvininejad et al. 2019; Ghazvininejad, Levy, and Zettlemoyer 2020; Kasai et al. 2020; Tu et al. 2020; Guo, Xu, and Chen 2020; Ran et al. 2020) which extend one-step NAT to multi-step NAT and generate translations iteratively. Different from existing works, we propose to explicitly model reordering information in NAT models, which serves as a proxy in capturing target word dependencies and encourages the decoder to choose words belonging to the same translation to alleviate the multimodality problem. This work intends to enhance the translation quality of one-step NAT models and has the potential to improve the performance of each iteration of multi-step NAT methods without loss of efficiency.

Modelling Reordering Information in Machine Translation

Re-ordering model is a key component in statistical machine translation (SMT), which handles word order differences between source and target languages. There has been a number of works focusing on word reordering in SMT, including deterministic reordering methods (Xia and McCord 2004; Collins, Koehn, and Kučerová 2005; Wang, Collins, and Koehn 2007; Li et al. 2007), which find an optimal reordering of source words; non-deterministic reordering methods (Kanthak et al. 2005; Zhang, Zens, and Ney 2007) which encode multiple alternative reorderings into a word lattice and remain the selection of best path in the decoder; and target word reordering methods (Bangalore, Haffner, and Kanthak 2007) which first select target lexicals and then reorder them to form final sentence. In neural machine translation (NMT), it has been shown the attention mechanism (Bahdanau, Cho, and Bengio 2014) could implicitly capture reordering information to some extent. Zhang et al. (2017) presented three distortion models to further incorporate reordering knowledge into attention-based NMT models. Chen et al. (2019) proposed to learn reordering embedding of a word based on its contextual information. Except for incorporating reordering knowledge in attention mechanism, researchers also proposed to learn to reorder source words according to target sentence structures with neural networks (Du and Way 2017; Kawara, Chu, and Arase 2018; Zhao, Zhang, and Zong 2018). This work empirically justifies reordering information is essential for NAT.

Conclusion and Future Work

In this work, to address the multimodality problem in NAT, we propose a novel NAT framework named ReorderNAT which explicitly models the reordering information in the decoding procedure. We further introduce deterministic and non-deterministic guiding decoding strategies to utilize the reordering information to encourage the decoder to choose words belonging to the same translation. Experimental results on several widely-used benchmarks show that our ReorderNAT model achieves better performance than most existing NAT models, and even achieves comparable translation quality as AT model with a significant speedup. We believe that to well model the reordering information is a potential way towards better NAT.

References

Akoury, N.; Krishna, K.; and Iyyer, M. 2019. Syntactically Supervised Transformers for Faster Neural Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1269–1281.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of International Conference on Learning Representations*.

Bangalore, S.; Haffner, P.; and Kanthak, S. 2007. Statistical Machine Translation through Global Lexical Selection and Sentence Reconstruction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 152–159.

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* .

Chen, K.; Wang, R.; Utiyama, M.; and Sumita, E. 2019. Neural Machine Translation with Reordering Embeddings. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, 1787–1799.

Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1724–1734.

Collins, M.; Koehn, P.; and Kučerová, I. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 531–540.

Du, J.; and Way, A. 2017. Pre-Reordering for Neural Machine Translation: Helpful or Harmful? *The Prague Bulletin of Mathematical Linguistics* 108(1): 171–182.

Ghazvininejad, M.; Karpukhin, V.; Zettlemoyer, L.; and Levy, O. 2020. Aligned Cross Entropy for Non-Autoregressive Machine Translation. In *Proceedings of the 37th International Conference on Machine Learning*.

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*, 6114–6123.

Ghazvininejad, M.; Levy, O.; and Zettlemoyer, L. 2020. Semi-Autoregressive Training Improves Mask-Predict Decoding. *arXiv preprint arXiv:2001.08785* .

Gu, J.; Bradbury, J.; Xiong, C.; Li, V. O.; and Socher, R. 2018. Non-autoregressive neural machine translation. In *Proceedings of International Conference on Learning Representations*.

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

Guo, J.; Tan, X.; He, D.; Qin, T.; Xu, L.; and Liu, T.-Y. 2019a. Non-autoregressive neural machine translation with enhanced decoder input. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, volume 33, 3723–3730.

Guo, J.; Tan, X.; Xu, L.; Qin, T.; Chen, E.; and Liu, T.-Y. 2019b. Fine-Tuning by Curriculum Learning for Non-Autoregressive Neural Machine Translation. *arXiv preprint arXiv:1911.08717* .

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.

Isozaki, H.; Hirao, T.; Duh, K.; Sudoh, K.; and Tsukada, H. 2010. Automatic Evaluation of Translation Quality for Distant Language Pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 944–952.

Kaiser, L.; Bengio, S.; Roy, A.; Vaswani, A.; Parmar, N.; Uszkoreit, J.; and Shazeer, N. 2018. Fast Decoding in Sequence Models Using Discrete Latent Variables. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, 2390–2399.

Kanthak, S.; Vilar, D.; Matusov, E.; Zens, R.; and Ney, H. 2005. Novel reordering approaches in phrase-based statistical machine translation. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, 167–174.

Kasai, J.; Cross, J.; Ghazvininejad, M.; and Gu, J. 2020. Non-autoregressive Machine Translation with Disentangled Context Transformer. In *Proceedings of the 37th International Conference on Machine Learning*.

Kawara, Y.; Chu, C.; and Arase, Y. 2018. Recursive Neural Network Based Preordering for English-to-Japanese Machine Translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Student Research Workshop*, 21–27.

Kim, Y.; and Rush, A. M. 2016. Sequence-Level Knowledge Distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1317–1327.

Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics on Interactive Poster and Demonstration Sessions*, 177–180.

Koehn, P.; Och, F. J.; and Marcu, D. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 48–54.

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.

Lee, J.; Shu, R.; and Cho, K. 2020. Iterative Refinement in the Continuous Space for Non-Autoregressive Neural Machine Translation. *arXiv preprint arXiv:2009.07177* .

Li, C.-H.; Li, M.; Zhang, D.; Li, M.; Zhou, M.; and Guan, Y. 2007. A probabilistic approach to syntax-based reordering for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 720–727.

Li, X.; Meng, Y.; Yuan, A.; Wu, F.; and Li, J. 2020. LAVA NAT: A Non-Autoregressive Translation Model with Look-Around Decoding and Vocabulary Attention. *arXiv preprint arXiv:2002.03084* .

Li, Z.; Lin, Z.; He, D.; Tian, F.; Qin, T.; Wang, L.; and Liu, T.-Y. 2019. Hint-based Training for Non-Autoregressive Machine Translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 5712–5717.

Libovický, J.; and Helcl, J. 2018. End-to-End Non-Autoregressive Neural Machine Translation with Connectionist Temporal Classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3016–3021.

Liu, J.; Ren, Y.; Tan, X.; Zhang, C.; Qin, T.; Zhao, Z.; and Liu, T.-Y. 2020. Task-Level Curriculum Learning for Non-Autoregressive Neural Machine Translation. *arXiv preprint arXiv:2007.08772* .

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*, 4273–4283.

Ran, Q.; Lin, Y.; Li, P.; and Zhou, J. 2020. Learning to Recover from Multi-Modality Errors for Non-Autoregressive Neural Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 3059–3069.

Shao, C.; Feng, Y.; Zhang, J.; Meng, F.; Chen, X.; and Zhou, J. 2019a. Retrieving Sequential Information for Non-Autoregressive Neural Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3013–3024.

Shao, C.; Zhang, J.; Feng, Y.; Meng, F.; and Zhou, J. 2019b. Minimizing the Bag-of-Ngrams Difference for Non-Autoregressive Neural Machine Translation. *arXiv preprint arXiv:1911.09320* .

Stern, M.; Chan, W.; Kiros, J.; and Uszkoreit, J. 2019. Insertion Transformer: Flexible Sequence Generation via Insertion Operations. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 5976–5985. Long Beach, California, USA: PMLR.

Sun, Z.; Li, Z.; Wang, H.; He, D.; Lin, Z.; and Deng, Z. 2019. Fast Structured Decoding for Sequence Models. In *Proceedings of Advances in Neural Information Processing Systems 32*, 3011–3020.

Sun, Z.; and Yang, Y. 2020. An EM Approach to Non-autoregressive Conditional Sequence Generation. In *Proceedings of the 37th International Conference on Machine Learning*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, 3104–3112.

Tu, L.; Pang, R. Y.; Wiseman, S.; and Gimpel, K. 2020. ENGINE: Energy-Based Inference Networks for Non-Autoregressive Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2819–2826.

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 *Proceedings of Advances in Neural Information Processing Systems 30*, 5998–6008.

Wang, C.; Collins, M.; and Koehn, P. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing*, 737–745.

Wang, C.; Zhang, J.; and Chen, H. 2018. Semi-Autoregressive Neural Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 479–488.

Wang, Y.; Tian, F.; He, D.; Qin, T.; Zhai, C.; and Liu, T.-Y. 2019. Non-autoregressive machine translation with auxiliary regularization. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, 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.

Xia, F.; and McCord, M. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of the 20th International Conference on Computational Linguistics*, 508.

Zhang, J.; Wang, M.; Liu, Q.; and Zhou, J. 2017. Incorporating word reordering knowledge into attention-based neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 1524–1534.

Zhang, Y.; Zens, R.; and Ney, H. 2007. Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. In *Proceedings of the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation*, 1–8.

Zhao, Y.; Zhang, J.; and Zong, C. 2018. Exploiting Pre-Ordering for Neural Machine Translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*.