Non-Autoregressive Machine Translation with Auxiliary Regularization

Yiren Wang,[1]* Fei Tian,[2] Di He,[3] Tao Qin,[2] ChengXiang Zhai,[1] Tie-Yan Liu[2]
[1]University of Illinois at Urbana-Champaign, Urbana, IL, USA
[2]Microsoft Research, Beijing, China
[3]Key Laboratory of Machine Perception, MOE, School of EECS, Peking University, Beijing, China
[1]{yiren,czhai}@illinois.edu
[2]{fetia, taoqin, tie-yan.liu}@microsoft.com
[3]di_he@pku.edu.cn

Abstract

As a new neural machine translation approach, Non-Autoregressive machine Translation (NAT) has attracted attention recently due to its high efficiency in inference. However, the high efficiency has come at the cost of not capturing the sequential dependency on the target side of translation, which causes NAT to suffer from two kinds of translation errors: 1) repeated translations (due to indistinguishable adjacent decoder hidden states), and 2) incomplete translations (due to incomplete transfer of source side information via the decoder hidden states). In this paper, we propose to address these two problems by improving the quality of decoder hidden representations via two auxiliary regularization terms in the training process of an NAT model. First, to make the hidden states more distinguishable, we regularize the similarity between consecutive hidden states based on the corresponding target tokens. Second, to force the hidden states to contain all the information in the source sentence, we leverage the dual nature of translation tasks (e.g., English to German and German to English) and minimize a backward reconstruction error to ensure that the hidden states of the NAT decoder are able to recover the source side sentence. Extensive experiments conducted on several benchmark datasets show that both regularization strategies are effective and can alleviate the issues of repeated translations and incomplete translations in NAT models. The accuracy of NAT models is therefore improved significantly over the state-of-the-art NAT models with even better efficiency for inference.

Introduction

Neural Machine Translation (NMT) based on deep neural networks has gained rapid progress over recent years (Cho et al. 2014; Bahdanau, Cho, and Bengio 2014; Wu et al. 2016; Vaswani et al. 2017; Hassan et al. 2018). NMT systems are typically implemented in an encoder-decoder framework, in which the encoder network feeds the representations of source side sentence x into the decoder network to generate the tokens in target sentence y. The decoder typically works in an auto-regressive manner: the generation of the t-th token y_t follows a conditional distribution $P(y_t|x, y_{<t})$, where $y_{<t}$ represents all the generated tokens before y_t. Therefore, during the inference process, we have to sequentially translate each target-side word one by one, which substantially hampers the inference efficiency of NMT systems.

To alleviate the latency of inference brought by autoregressive decoding, recently the community has turned to Non-Autoregressive Translation (NAT) systems (Gu et al. 2018; Lee, Mansimov, and Cho 2018; Kaiser et al. 2018). A *basic NAT model* has the same encoder-decoder architecture as the autoregressive translation (AT) model, except that the sequential dependency within the target side sentence is omitted. In this way, all the tokens can be generated in parallel, and the inference speed is thus significantly boosted. However, it comes at the cost that the translation quality is largely sacrificed since the intrinsic dependency within the natural language sentence is abandoned. To mitigate such performance degradation, previous work has tried different ways to insert intermediate discrete variables to the basic NAT model, so as to incorporate some light-weighted sequential information into the non-autoregressive decoder. The discrete variables include the autoregressively generated latent variables (Kaiser et al. 2018), and the fertility information brought by a third-party word alignment model (Gu et al. 2018). However, leveraging such discrete variables not only brings additional difficulty for optimization, but also slows down the translation by introducing extra computational cost for producing such discrete variables.

In this paper, we propose a new solution to the problem that does not rely on any discrete variables and makes little revision to the basic NAT model, thus retaining most of the benefit of an NAT model. Our approach was motivated by the following result we obtained from carefully analyzing the key issues existed in the basic NAT model. We empirically observed that the two types of translation errors frequently made by the basic NAT model are: 1) *repeated translation*, where the same token is generated repeatedly at consecutive time steps; 2) *incomplete translation*, where the semantics of several tokens/sentence pieces from the source sentence are not adequately translated. Both issues suggest that the decoder hidden states in NAT model, i.e., the hidden states output in the topmost layer of the decoder, are of low quality: the repeated translation shows that two adjacent hidden states are *indistinguishable*, leading to the same tokens decoded out, while the incomplete translation reflects that the hidden states in the decoder are *incomplete* in repre-

*The work was done when the author was an intern at Microsoft Research Asia.

senting source side information.

Such a poor quality of decoder hidden states is in fact a direct consequence of the non-autoregressive nature of the model: at each time step, the hidden states have no access to their prior decoded states, making them in a 'chaotic' state being unaware of what has and has not been translated. Therefore, it is difficult for the neural network to learn good hidden representations all by itself. Thus, in order to improve the quality of the decoder representations, we must go beyond the pure non-autoregressive models. The challenge, though, is how to improve the quality of decoder representations to address the two problems identified above while still keeping the benefit of efficient inference of the NAT models.

We propose to address this challenge by directly regularizing the learning of the decoder representations using two auxiliary regularization terms for model training. First, to overcome the problem of repeated translation, we propose to force the similarity of two neighboring hidden state vectors to be well aligned with the similarity of the embedding vectors representing the two corresponding target tokens they aim to predict. We call this regularization strategy *similarity regularization*. Second, to overcome the problem of incomplete translation, inspired by the dual nature of machine translation task (He et al. 2016), we propose to impose that the hidden representations should be able to reconstruct the source side sentence, which can be achieved by putting an auto-regressive backward translation model on top of the decoder; the backward translation model would "demand" the decoder hidden representations of NAT to contain enough information about the source. We call this regularization strategy *reconstruction regularization*. Both regularization terms only appear in the training process and have no effect during inference, thus bringing no additional computational/time cost to inference and allowing us to retain the major benefit of an NAT model. Meanwhile, the *direct* regularization on the hidden states effectively improves their representation. In contrast, the existing approaches would need either a third-party word alignment module (Gu et al. 2018) (hindering end-to-end learning), or a special component handling the discrete variables (e.g., several softmax operators in (Lee, Mansimov, and Cho 2018)) which brings additional latency for decoding.

We evaluate the proposed regularization strategies by conducting extensive experiments on several benchmark machine translation datasets. The experiment results show that both regularization strategies are effective and they can alleviate the issues of repeated translations and incomplete translations in NAT models, leading to improved NAT models that can improve accuracy substantially over the state-of-the-art NAT models without sacrificing efficiency. We set a new state-of-the-art performance of non-autoregressive translation models on the WMT14 datasets, with 24.61 BLEU in En-De and 28.90 in De-En.

Background

Neural machine translation (Bahdanau, Cho, and Bengio 2014) (NMT) has been the widely adopted machine translation approach within both academia and industry. A neural machine translation system specifies a conditional distribution $P(y|x)$ of the likelihood of translating source side sentence x into target sentence y. There are typically two parts in NMT: the encoder network and the decoder network. The encoder reads the source side sentence $x = (x_1, \cdots, x_{T_x})$ with T_x tokens, and processes it into context vectors which are then fed into the decoder network. The decoder builds the conditional distribution $P(y_t|y_{<t}, x)$ at each decoding time-step t, where $y_{<t}$ represents the set of generated tokens before time-step t. The final distribution $P(y|x)$ is then in the factorized form $P(y|x) = \prod_{t=1}^{T_y} P(y_t|y_{<t}, x)$, with y containing T_y target side tokens $y = (y_1, \cdots, y_{T_y})$.

The NMT suffers from the high inference latency, which limits its application in the real world scenarios. The main bottleneck comes from its autoregressive nature of the sequence generation: each target side token is generated one by one, which prevents parallelism during inference, and thus the computational power of GPU cannot be fully exploited. Recent research efforts have been resorted to solve the huge latency issue in the decoding process. In the domain of speech synthesis, parallel wavenet (van den Oord et al. 2018) successfully achieves the parallel sampling based on inverse autoregressive flows (Kingma et al. 2016) and knowledge distillation (Hinton, Vinyals, and Dean 2015). In NMT, the non-autoregressive neural machine translation (NAT) model has been developed recently (Gu et al. 2018; Lee, Mansimov, and Cho 2018; Kaiser et al. 2018). In NAT, all the tokens within one target side sentence are generated in parallel, without any limitation of sequential dependency. The inference speed has thus been significantly boosted (e.g., tens of times faster than the typical autoregressive NMT model) and for the sake of maintaining translation quality, there are several technical innovations in the NAT model design from previous exploration:

- *Sequence level knowledge distillation*. NAT model is typically trained with the help from an autoregressive translation (AT) model as its teacher. The knowledge of the AT model is distilled to the NAT model via the sequence level distillation technique (Kim and Rush 2016) which is essentially using the sampled translation from the teacher model, out from the source side sentences, as the bilingual training data. It has been reported previously (Gu et al. 2018), and is also consistent with our empirical studies that training NAT with such distilled data performs much better than the original ground truth data, or the mixture of ground truth and distilled translations. There has been no precise theoretical justification, but an intuitive explanation is that the NAT model suffers from the 'multimodality' problem (Gu et al. 2018) – there might be multiple valid translations for the same source word/phrase/sentence. The distilled data via the teacher model eschew such problem since the output of a neural network is less noisy and more 'deterministic'.
- *Model architecture modification*. The NAT model is similar in the framework of encoder and decoder but has several differences from the autoregressive model, including (a) the causal mask of decoder, which prevents the earlier decoding steps from accessing later information, is re-

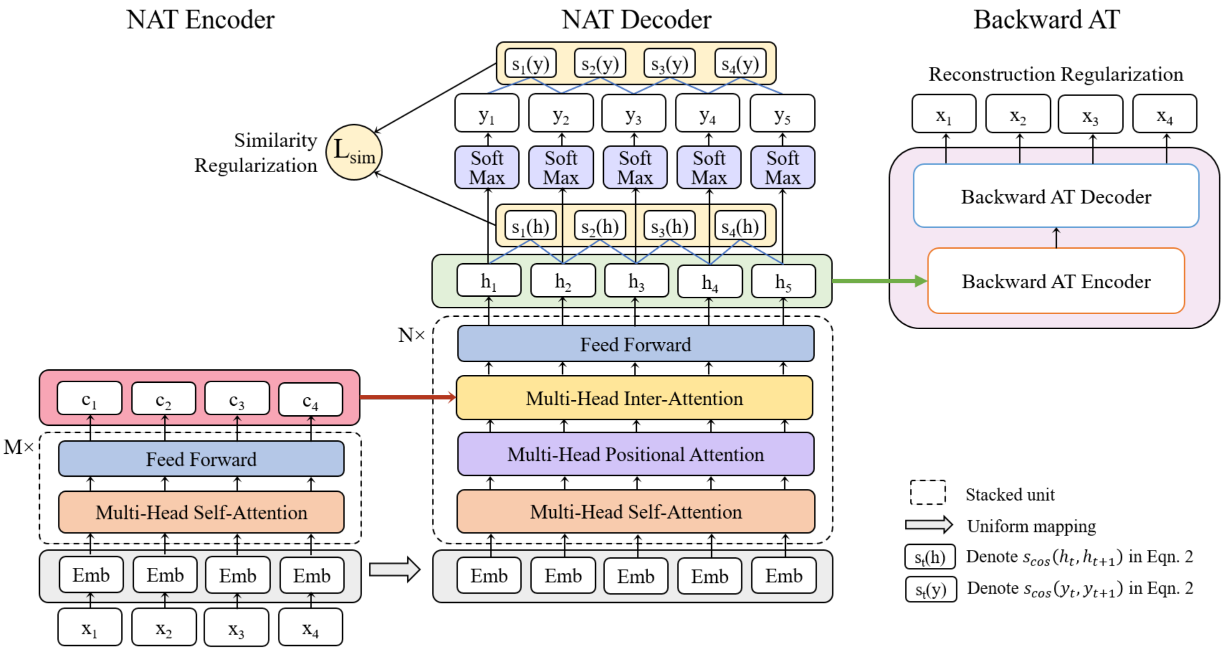

Figure 1: The overall architecture of NAT model with auxiliary regularization terms. *AT* stands for autoregressive translation. Each sublayer of the encoder/decoder contains a residual connection and layer normalization following (Vaswani et al. 2017).

moved; (b) positional attention is leveraged in the decoder to enhance position information, where the positional embeddings (Vaswani et al. 2017) are used as query and key, and hidden representations from the previous layers are used as value (Gu et al. 2018).

- *Discrete variables to aid NAT model training and inference*. There are intermediate discrete variables in previous NAT models, which aim to make up the loss of sequential information in the non-autoregressive decoding. The representative examples include the fertility value to indicate the number of copying each source side token (Gu et al. 2018), and the discrete latent variables autoregressively generated (Kaiser et al. 2018). Additional difficulty arises due to the discreteness, and thus specially designed optimization processes are necessary such as variational method (Gu et al. 2018; Lee, Mansimov, and Cho 2018) and vector quantization (Kaiser et al. 2018).

Model

We introduce the details of our proposed approaches to non-autoregressive neural machine translation in this section. The overall model architecture is depicted in Fig. 1.

We use the basic NAT model with the same encoder-decoder architecture as the AT model, and follow two of the aforementioned techniques of training NAT models. First, we train our NAT models based on the sequence-level knowledge distillation. Second, we obey the aforementioned model variants specially designed for NAT, such as the positional attention and removing causal mask in the decoder (Gu et al. 2018). What makes our solution unique is that we replace the hard-to-optimize discrete variables in prior works with two simple regularization terms. As a prerequisite, since there are no discrete variables available to indicate some sequential information, we need new mechanisms to *predict the target length during inference* and *generate inputs for the decoder*, which are necessary for the NAT model since the information fed into the decoder (i.e., the word embeddings of previous target side tokens in autoregressive model) is unknown due to parallel decoding.

Target length prediction during inference. Instead of estimating target length T_y with fertility values (Gu et al. 2018), or with a separate model $p(T_y|x)$ (Lee, Mansimov, and Cho 2018), we use a simple yet effective method as (Anonymous 2019) that leverages source length to determine target length. During inference, the target side length T_y is predicted as $T_y = T_x + \Delta T$, where T_x is the length of source sentence, ΔT is a constant bias term that can be set according to the overall length statistics of the training data.

Generate decoder input with uniform mapping. Different from other works relying on the fertility (Gu et al. 2018), we adopt a simple way of uniformly mapping the source side word embedding as the inputs to the decoder. Specifically, given source sentence $x = \{x_1, x_2, ..., x_{T_x}\}$ and target length T_y, we uniformly map the embeddings (denoted as $\mathcal{E}(\cdot)$) of the source tokens to generate decoder input with:

$$z_k = \mathcal{E}(x_i),\ i = \left[\frac{T_x}{T_y}t\right],\ t = 1, 2, ..., T_y. \quad (1)$$

We use this variant of the basic NAT model as our backbone model (denoted as "*NAT-BASE*"). We now turn to analyze the two commonly observed translation issues of NAT-BASE and introduce the corresponding solutions to tackle them.

Repeated Translation & Similarity Regularization

The issue of *repeated translation* refers to the same word token being successively generated for multiple times, e.g., the “mobility mobility , , s” in the second example of Table 2. Such an issue is quite common for NAT models. On IWSLT14 De-En dataset, we empirically find that more than half of NAT translations contain such repeated tokens. Furthermore, simply de-duplicating such repeated tokens brings limited gain (shown in Table 3). This is consistent with our observation that very likely there will be additional translation errors close to the repeated tokens, e.g., “caniichospital rates” in the same example. Apparently, it reflects that the decoder hidden states at consecutive time steps are largely similar, if the same tokens are decoded out from them.

One natural way to tackle these indistinguishable representations is to impose a similarity regularization term to force the two hidden states to be far away. Here the similarity measure to be minimize could be, for example, the cosine similarity $s_{cos}(h,h') = \frac{hh'^T}{||h||\cdot||h'||}$ and L2 similarity $s_{l2}(h,h') = -||h-h'||_2^2$. Acting in this way is simple and straightforward, but it is also apparently problematic: the degree of closeness for each pair of adjacent hidden states (h_t, h_{t+1}) varies with data. Universally imposing the same similarity regularization term for all data (x, y) and all timestep t does not accommodate such flexibility. For example, consider the target side phrase (y_t, y_{t+1}, y_{t+2})=‘hollywood movie 2019’, the hidden state h_{t+1} should be closer to h_t than with h_{t+2}, since the token ‘movie’ is semantically more related with ‘hollywood’ than ‘2019’. Then adding the same constraint on (h_t, h_{t+1}) and (h_{t+1}, h_{t+2}) is sub-optimal.

We therefore adopt a more flexible way to specify the degree of similarity regularization on two adjacent hidden states. Our intuition is consistent with the example above: for (h_t, h_{t+1}), if the target tokens (y_t, y_{t+1}) which are decoded from them are semantically far apart, then the similarity between h_t and h_{t+1} should be correspondingly smaller, and vice versa. Here the semantic similarity between two word tokens y_t and y_{t+1} is calculated with their word embeddings (Mikolov et al. 2013). The regularization term between h_t and h_{t+1} then goes as:

$$L_{sim,t} = 1 + s_{cos}(h_t, h_{t+1}) \cdot (1 - s_{cos}(y_t, y_{t+1})), \quad (2)$$

where with a little abuse of notations, y_t also represents the target side word embedding in the NMT model for word y_t. The regularization term basically says, if two words y_t and y_{t+1} are dissimilar ($s_{cos}(y_t, y_{t+1})$ is smaller), then the similarity regularization (i.e., $s_{cos}(h_t, h_{t+1})$) should have larger effect, and vice versa. Here the constant 1 is for keeping the loss values non-negative. Note here we do not incur gradients on the word embeddings y_t and y_{t+1} via Eqn. 3, for fear of bad effects brought by unqualified hidden states to the training of word embeddings. The overall loss on the sample (x, y) then is:

$$L_{sim} = \sum_{t}^{T_y} L_{sim,t}. \quad (3)$$

Incomplete Translation & Reconstruction Regularization

The *incomplete translation* means that several tokens/phrases in the source sentence are neglected by the translation model. Previously the autoregressive translation model was also observed to suffer from incomplete translation (Tu et al. 2016b). However, we empirically find that with the more powerful backbone model, such as the deep Transformer, incomplete translation is not common for autoregressive translation, as is shown by the outputs of the teacher model in Table 2 (labeled by ‘AT’). However, NAT suffers from incomplete translation significantly, such as in the same table, the source side phrase ‘und manches nicht’ in the third example is totally missed by the NAT model. We further take a close look at 50 randomly non-autoregressive translation results sampled from IWSLT14 De-En dataset, and find that more than 30% of them leave something behind on the source side.

The reconstruction regularization targets at solving the incomplete translation problem. Concretely speaking, we borrow the spirit of dual learning (He et al. 2016) by bringing in an additional backward translation model $P(x|y)$ which we further denote as $\overleftarrow{P}$ for the ease of statement. We couple the two models (i.e., the original NAT model P translating x to y and the backward translation model $\overleftarrow{P}$ translating y to x) together by feeding the hidden states output by the decoder of P (i.e., $h = \{h_1, \cdots, h_{T_y}\}$) to $\overleftarrow{P}$, and asking $\overleftarrow{P}$ to recover the source sentence from h. Here h simply acts as the input embeddings for the encoder of $\overleftarrow{P}$. The reconstruction loss for a training data (x, y) then goes as:

$$L_{rec} = -\sum_{t=1}^{T_x} \log \overleftarrow{P}(x_t|h, x_{<t}). \quad (4)$$

We set the backward translation model $\overleftarrow{P}$ as the autoregressive model since autoregressive model provides more accurate information. $\overleftarrow{P}$ will not be used in the inference process of P. One can easily see $\overleftarrow{P}$ forces h to be more comprehensive in storing the source side information by directly providing x as training objective.

Here another intuitive view to understand the effect of $\overleftarrow{P}$ is that it acts in a similar way as the discriminator of generative adversarial networks (Goodfellow et al. 2014): $\overleftarrow{P}$ decides whether h is good enough by testing whether x can be perfectly recovered from h. Then the NAT model P plays the effect of the generator with the hope of reconstructing x, and thus fooling $\overleftarrow{P}$. Setting $\overleftarrow{P}$ to be autoregressive makes the ‘discriminator’ not easily fooled, urging the output of ‘generator’, i.e., the hidden state h, to be as complete as possible to deliver the information from source side.

There are prior works for enhancing the adequacy of translation results, but devoted to autoregressive NMT models (Tu et al. 2016a; 2016b; 2017). Among all these prior works, our approach is most technically similar with that in (Tu et al. 2017), with the difference that for reconstructing from h to x, we leverage a complete backward translation model based on encoder-decoder framework, while they adopt a simple decoder without encoder. We empirically find that such an encoder cannot be dropped in the scenario of helping NAT models, which we conjecture is due to the better expressiveness brought by the encoder. Another difference is that they furthermore leverage the reconstruction score to assist the inference process via re-ranking different candidates, while we do not perform such a step since it hurts the efficiency of inference.

Joint Training

In the training process, all the two regularization terms are combined with the commonly used cross-entropy loss L_{ce}. For a training data (x, y) in which y is the sampled translation of x from the teacher model, the total loss is:

$$L = L_{ce} + \alpha L_{sim} + \beta L_{rec} \tag{5}$$

$$L_{ce} = \sum_{t=1}^{T_y} \log P(y_t|x) \tag{6}$$

where α and β are the trade-off parameters, L_{sim} and L_{rec} respectively denotes the similarity regularization in Eqn.3 and the reconstruction regularization in Eqn. 4. We denote our model with the two regularization terms as *"NAT-REG"*

Evaluation

We use multiple benchmark datasets to evaluate the effectiveness of the proposed approach. We compare the proposed approach with multiple baseline approaches that represent the state-of-the-art in terms of both translation accuracy and efficiency, and analyze the effectiveness of the two regularization strategies. As we will show, the strategies are effective and can lead to substantial improvements of translation accuracy with even better efficiency.

Experiment Design

Datasets We use several widely adopted benchmark datasets to evaluate the effectiveness of our proposed method: the IWSLT14 German to English translation (IWSLT14 De-En), the IWSLT16 English to German (IWSLT16 En-De), and the WMT14 English to German (WMT14 En-De) and German to English (WMT14 De-En) which share the same dataset. For IWSLT16 En-De, we follow the dataset construction in (Gu et al. 2018; Lee, Mansimov, and Cho 2018) with roughly $195k/1k/1k$ parallel sentences as training/dev/test sets respectively. We use the fairseq recipes[1] for preprocessing data and creating dataset split for the rest datasets. The IWSLT14 De-En contains roughly $153k/7k/7k$ parallel sentences as training/dev/test sets. The WMT14 En-De/De-En dataset is much larger with $4.5M$ parallel training pairs. We use Newstest2014 as test set and Newstest2013 as dev set. We use byte pair encoding (BPE) (Sennrich, Haddow, and Birch 2016) to segment word tokens into subword units, forming a $32k$-subword vocabulary shared by source and target languages for each dataset.

Baselines We include three latest NAT works as our baselines, the NAT with fertility (NAT-FT) (Gu et al. 2018), the NAT with iterative refinement (NAT-IR) (Lee, Mansimov, and Cho 2018) and the NAT with discrete latent variables (LT) (Kaiser et al. 2018). For all our four tasks, we obtain the baseline performance by either directly using the performance figures reported in the previous works if they are available or producing them by using the open source implementation of baseline algorithms on our datasets.

Model Settings For the teacher model, we follow the same setup with previous NAT works to adopt the Transformer model (Vaswani et al. 2017) as teacher model for both sequence-level knowledge distillation and inference rescoring. We use the same model size and hyperparameters for each NAT model and its respective AT teacher. Here we note that several baseline results are fairly weak probably due to the weak teacher models. For example, (Gu et al. 2018) reports an AT teacher with 23.45 BLEU for WMT14 En-De task, while the official number of Transformer is 27.3 (Vaswani et al. 2017). For fair comparison, we bring in a weakened AT teacher model with the same model architecture, yet sub-optimal performances similar to the teacher models in previous works. We rerun our algorithms with such weakened teacher models.

For the NAT model, we similarly adopt the Transformer architecture. For WMT datasets, we use the hyperparameter settings of `base` Transformer model in (Vaswani et al. 2017). For IWSLT14 DE-EN, we use the `small` Transformer setting with a 5-layer encoder and 5-layer decoder (size of hidden states and embeddings is 256, and the number of attention heads is 4). For IWSLT16 EN-DE, we use a slightly different version of small settings with 5 layers from (Gu et al. 2018), where size of hidden states and embeddings are 278, number of attention heads is 2. For our models with reconstruction regularization, the backward AT models share word embeddings and the same model size with the NAT model. Our models are implemented based on the official TensorFlow implementation of Transformer[2].

For sequence-level distillation, we set beam size to be 4. For our model NAT-REG, we determine the trade-off parameters, i.e, α and β in Eqn. 5 by the BLEU on the IWSLT14 De-En dev set, and use the same values for all other datasets. The optimal values are $\alpha = 2$ and $\beta = 0.5$.

Training and Inference

We train all models using Adam following the optimizer settings and learning rate schedule in Transformer(Vaswani et al. 2017). We run the training procedure on 8/1 Nvidia M40 GPUs for WMT and IWSLT datasets respectively. Distillation and inference are run on 1 GPU.

For inference, we follow the common practice of noisy parallel decoding (Gu et al. 2018), which generates a number of decoding candidates in parallel and selects the best translation via re-scoring using AT teacher. In our scenario, we generate multiple translation candidates by predicting different target lengths $T_y \in [T_x + \Delta T - B, T_x + \Delta T + B]$, which

[1] https://github.com/pytorch/fairseq

[2] https://github.com/tensorflow/tensor2tensor.

Models/Datasets	WMT14 En-De	WMT14 De-En	IWSLT14 De-En	IWSLT16 En-De	Latency	Speedup
Autoregressive Models (AT Teachers)						
Transformer (NAT-FT)	23.45	27.02	31.47†	29.70	–	–
Transformer (NAT-IR)	24.57	28.47	30.90†	28.98	–	–
Transformer (LT)	27.3	/	/	/	–	–
Transformer (NAT-REG)	27.3	31.29	33.52	28.35	607 ms	1.00×
Transformer (NAT-REG, Weak Teacher)	24.50	28.76	/	/	–	–
Non-Autoregressive Models						
NAT-FT (no NPD)	17.69	21.47	20.32†	26.52	39 ms	15.6×
NAT-FT (NPD rescoring 10)	18.66	22.41	21.39†	27.44	79 ms	7.68×
NAT-FT (NPD rescoring 100)	19.17	23.20	24.21†	28.16	257 ms	2.36×
NAT-IR (1 refinement)	13.91	16.77	21.86†	22.20	68† ms	8.9×
NAT-IR (10 refinements)	21.61	25.48	23.94†	27.11	404† ms	1.5×
NAT-IR (adaptive refinements)	21.54	25.43	24.63†	27.01	320† ms	1.9×
LT (no rescoring)	19.8	/	/	/	105 ms	5.78×
LT (rescoring 10)	21.0	/	/	/	/	/
LT (rescoring 100)	22.5	/	/	/	/	/
NAT-REG (no rescoring)	20.65	24.77	23.89	23.14	22 ms	27.6×
NAT-REG (rescoring 9**)**	24.61	28.90	28.04	27.02	40 ms	15.1×
NAT-REG (WT, no rescoring)	19.15	23.20	/	/	–	–
NAT-REG (WT, rescoring 9**)**	22.80	27.12	/	/	–	–

Table 1: The test set performances of AT and NAT models in BLEU score. *NAT-FT*, *NAT-IR* and *LT* denotes the baseline method in (Gu et al. 2018), (Lee, Mansimov, and Cho 2018) and (Kaiser et al. 2018) respectively. *NAT-REG* is our proposed NAT with two regularization terms and *Transformer (NAT-REG)* is correspondingly our AT teacher model. 'Weak Teacher' or 'WT' refers to the NAT trained with weakened teacher comparable to prior works. All AT models are decoded with a beam size of 4. '†' denotes baselines from our reproduction. '–' denotes same numbers as above/below.

results in $2B+1$ candidates. In our experiments, we set ΔT to 2, -2, -1, 1 for WMT14 EN-De, De-En and IWSLT14 De-En, IWSLT16 En-De respectively according to the average length distribution in the training set.

We evaluate the model performances with tokenized case-sensitive BLEU[3] (Papineni et al. 2002) for WMT14 datasets and tokenized case-insensitive BLEU for IWSLT14 datasets. Latency is computed as average per sentence decoding time (ms) on the full test set of IWSLT14 DE-EN without mini-batching. We test latency on 1 NVIDIA Tesla P100 to keep in line with previous works (Gu et al. 2018).

Results

We report all the results in Table 1, from which we can make the following conclusions:

1. **Our model improves NAT translation quality with a large margin**. On all the benchmark datasets except for IWSLT16 En-De, our NAT-REG achieves the best translation quality. On the WMT datasets, NAT-REG with strong teacher model has achieved new state-of-the-art performances for NAT, with 24.61 in En-De and 28.90 in De-En, which even outperform AT teachers in previous works. On the small dataset of IWSLT16 En-De, due to the inferior of our teacher model (28.35 vs. 29.70 of NAT-FT), our performance is slightly weaker than the previous works.

2. **The improvements over baseline model are not merely brought by the stronger teacher models.** In both WMT tasks, our model achieves better performances with weaker AT teacher model that is on par with the teacher model in previous works (e.g. 27.12 of NAT-REG (weak teacher) vs. 25.48 of NAT-IR and 22.41 of NAT-FT on WMT14 De-En). Therefore, it is clear that our proposal to replace hard-to-optimize discrete variables with the simple regularization terms, can indeed help obtain better NAT models.

3. **Our NAT model achieves significantly better latency without the intermediate discrete variables hampering inference efficiency**. For example, the speedup compared with AT model is 15.1× when re-scoring 9 candidates, which is comparable to NAT-FT decoding without re-scoring (15.6×). This further verifies that by removing the intermediate discrete variables, simply decoding via the neural networks is beneficial to improve the inference speed.

Analysis

Case Study We present several translation examples sampled from the IWSLT14 De-En dataset in Table 2, including the source sentence, the target reference (i.e., the ground-truth translation), the translation given by the teacher model (*AT*), by the basic NAT with sequence distillation (*NAT-BASE*), and by our NAT with the two regularization terms (*NAT-REG*). As can be seen, *NAT-BASE* suffers severely from the issue of *repeated translation* (e.g., the 'climate climate change change' in the first example) and *incomplete*

[3] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

Source:	bei der coalergy sehen wir klimaveränderung als eine ernste gefahr für unser geschäft .
Reference:	at coalergy we view climate change as a very serious threat to our business .
AT:	in coalergy , we see climate change as a serious threat to our business .
NAT-BASE:	in the coalergy , we &apos;ll see **climate climate change change** as a most serious danger for our business .
NAT-REG:	at coalergy , we &apos;re seeing climate change as a serious threat to our business .
Source:	dies ist die großartigste zeit , die es je auf diesem planeten gab , egal , welchen maßstab sie anlegen :gesundheit , reichtum , mobilität , gelegenheiten , sinkende krankheitsraten .
Reference:	this is the greatest time there &apos;s ever been on this planet by any measure that you wish to choose : health , wealth , mobility , opportunity , *declining rates of disease* .
AT:	this is the greatest time you &apos;ve ever had on this planet , no matter what scale you &apos;re putting : health , wealth , mobility , opportunities , declining disease rates .
NAT-BASE:	this is the most greatest time that ever existed on this planet no matter what scale they &apos;re **imsi : : , , mobility mobility , , scaniichospital rates** .
NAT-REG:	this is the greatest time that we &apos;ve ever been on this planet no matter what scale they &apos;re **ianition** : health , wealth , mobility , opportunities , *declining disease rates* .
Source:	und manches davon hat funktioniert und manches nicht .
Reference:	and some of it worked , *and some of it didn &apos;t* .
AT:	and some of it worked *and some of it didn &apos;t work* .
NAT-BASE:	and some of it worked .
NAT-REG:	and some of it worked *and some not* .

Table 2: Translation examples from IWSLT14 De-En task. The AT result is decoded with a beam size of 4 and NAT results are generated by re-scoring 9 candidates. We use the italic fonts to indicate the translation pieces where the NAT has the issue of incomplete translation, and bond fonts to indicate the issue of repeated translation.

Model variants	BLEU
NAT-BASE	28.73
NAT-BASE + de-duplication	29.45
NAT-BASE + universally penalize similarity	28.32
NAT-BASE + similarity regularization	30.02
NAT-BASE + reconstruction regularization	30.21
NAT-BASE + both regularizations	30.84

Table 3: Ablation study on IWLST14 De-En dev set. Results are BLEU scores with teacher rescoring 9 candidates.

translation (e.g., incomplete translation of 'and some of it didn &apos;t' in the third example), while with the two auxiliary regularization terms brought in, the two issues are largely alleviated.

Ablation Study To further study the effects brought by different techniques, we show in Table 3 the translation performance of different NAT model variants for the IWSLT14 De-En translation task. We see that the BLEU of the basic NAT model could be enhanced via either of the two regularization terms by about 1 point. As a comparison, simply de-duplicating the repeated tokens brings certain level of gain, and with the universal regularization that simply penalizes the cosine similarity of all (h_t, h_{t+1}), the performance even drops (from 28.73 to 28.32). When combining both regularization strategies, the BLEU score goes higher, which shows that the two regularization strategies are somewhat complementary to each other. A noticeable fact is that the gain of combining both regularization strategies (about 2.1) is lower than the sum of each individual gain (about 2.8). One possible explanation may be that the two types of errors (*repeated translation* and *incomplete translation*) can be correlated to some extent and tackling one might help the alleviation of the other. For example, intuitively, given the fixed target side length T_y, if repeated tokens are removed, more 'valid' tokens will occupy the position, consequently reducing the possibility of incomplete translation.

Furthermore, we evaluate the effectiveness of alleviating repeated translations with our proposed approach. We count the number of de-duplication (de-dup) operations (for example, there are 2 de-dup operations for "we &apos;ll see climate ~~climate~~ change ~~change~~"). The per-sentence de-dup operations in IWSLT14 De-En dev set are 2.3 with NAT-BASE, which has dropped to 0.9 with the introduction of similarity regularization, clearly indicating the effectiveness of the similarity regularization.

Conclusions

In this paper, we proposed two simple regularization strategies to improve the performance of non-autoregressive machine translation (NAT) models, the similarity regularization and reconstruction regularization, which have been shown to be effective for addressing two major problems of NAT models, i.e., the repeated translation and incomplete translation, respectively, consequently leading to quite strong performance with fast decoding speed.

While the two regularization strategies were proposed to improve NAT models, they may be generally applicable to other sequence generation models. Exploring such potential would be an interesting direction for future research. For example, we can apply our methods to other sequence generation tasks such as image caption and text summarization, with the hope of successfully deploying the non-autoregressive models into various real-world applications. We also plan to break the upper bound of the autoregressive teacher model and obtain better performance than the autoregressive NMT model, which is possible since there is no gap between training and inference (i.e., the *exposure bias* problem for autoregressive sequence generation (Ranzato et al. 2015)) in NAT models.

References

Anonymous. 2019. Hint-based training for non-autoregressive translation. In *Submitted to International Conference on Learning Representations*. under review.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Cho, K.; van Merrienboer, 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 (EMNLP)*, 1724–1734. Doha, Qatar: Association for Computational Linguistics.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

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

Hassan, H.; Aue, A.; Chen, C.; Chowdhary, V.; Clark, J.; Federmann, C.; Huang, X.; Junczys-Dowmunt, M.; Lewis, W.; Li, M.; et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

He, D.; Xia, Y.; Qin, T.; Wang, L.; Yu, N.; Liu, T.; and Ma, W.-Y. 2016. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, 820–828.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

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, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2395–2404.

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.

Kingma, D. P.; Salimans, T.; Jozefowicz, R.; Chen, X.; Sutskever, I.; and Welling, M. 2016. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, 4743–4751.

Lee, J.; Mansimov, E.; and Cho, K. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. *arXiv preprint arXiv:1802.06901*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

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 on association for computational linguistics*, 311–318. Association for Computational Linguistics.

Ranzato, M.; Chopra, S.; Auli, M.; and Zaremba, W. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 1715–1725.

Tu, Z.; Liu, Y.; Lu, Z.; Liu, X.; and Li, H. 2016a. Context gates for neural machine translation. *arXiv preprint arXiv:1608.06043*.

Tu, Z.; Lu, Z.; Liu, Y.; Liu, X.; and Li, H. 2016b. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 76–85.

Tu, Z.; Liu, Y.; Shang, L.; Liu, X.; and Li, H. 2017. Neural machine translation with reconstruction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 3097–3103.

van den Oord, A.; Li, Y.; Babuschkin, I.; Simonyan, K.; Vinyals, O.; Kavukcuoglu, K.; van den Driessche, G.; Lockhart, E.; Cobo, L.; Stimberg, F.; Casagrande, N.; Grewe, D.; Noury, S.; Dieleman, S.; Elsen, E.; Kalchbrenner, N.; Zen, H.; Graves, A.; King, H.; Walters, T.; Belov, D.; and Hassabis, D. 2018. Parallel WaveNet: Fast high-fidelity speech synthesis. In Dy, J., and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 3918–3926. Stockholmsmssan, Stockholm Sweden: PMLR.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; Klingner, J.; Shah, A.; Johnson, M.; Liu, X.; Kaiser, Ł.; Gouws, S.; Kato, Y.; Kudo, T.; Kazawa, H.; Stevens, K.; Kurian, G.; Patil, N.; Wang, W.; Young, C.; Smith, J.; Riesa, J.; Rudnick, A.; Vinyals, O.; Corrado, G.; Hughes, M.; and Dean, J. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *ArXiv e-prints*.