Non-autoregressive Machine Translation by Modeling Syntactic Dependency Interrelation

Anonymous ACL submission

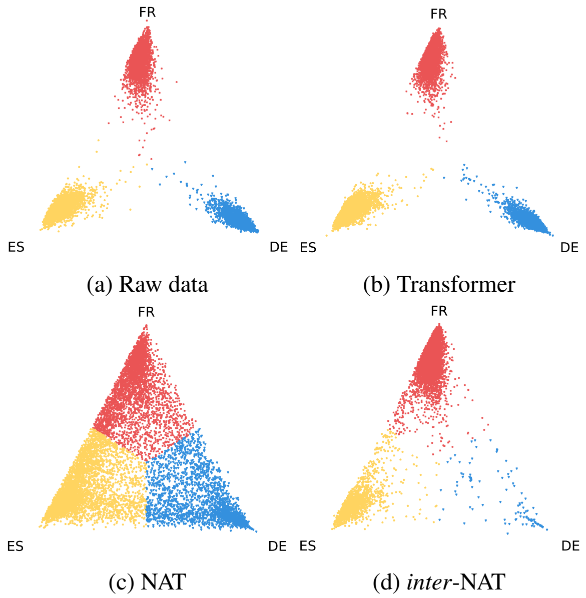
Abstract

Non-autoregressive Transformer (NAT) significantly improves translation efficiency by parallel decoding. However, the poor modeling of word inter-dependencies in NAT models prevents them from organizing consistent modes while learning the one-to-many multi-modality phenomenon. In this paper, we propose *inter*-NAT, which explicitly models the target-side word inter-dependencies for NAT models. We introduce the word inter-dependencies according to the syntactic dependency tree, which presents explicit modification relationships between the words. These dependencies could coordinate the translation of the target sentence and alleviate the multi-modality issue. Experiments results on the WMT14 and WMT16 tasks show that with only one-pass decoding *inter*-NAT achieves comparable or better performance than strong iterative NAT baselines while keeping a competitive efficiency.

1 Introduction

Non-autoregressive Transformer (NAT, Gu et al., 2018) introduces a new text generation paradigm, which generates the tokens of a sentence in parallel. It differs from the autoregressive models (Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017) in assuming that the target tokens in sentences are generated conditional independent of each other, supporting parallel decoding during inference. In practice, a vanilla NAT model (Gu et al., 2018) can achieve over 15 times speedup compared to an autoregressive Transformer (AT, Vaswani et al., 2017) in neural machine translation (NMT) tasks. However, existing NAT models (Libovický and Helcl, 2018; Ghazvininejad et al., 2020b; Saharia et al., 2020; Sun and Yang, 2020) still underperform the AT models in terms of the BLEU score (Papineni et al., 2002).

A well-recognized problem of NAT is the *multi-modality problem* (Gu et al., 2018), i.e., a source



(a) Raw data (b) Transformer (c) NAT (d) *inter*-NAT

Figure 1: Posterior distribution of language IDs (ES, DE, FR) for the outputs from different settings. Each ID represents a mode of the datasets. The more modes there are in the output, the more diverse the distribution is. More details in § 4.3.

sentence can have many valid translations, which usually leads to inconsistent translations when generated in parallel. We follow Zhou et al. (2020) to visualize the mode distribution from different models in Fig. 1. It is shown in Fig. 1b that the sequential decoding of AT is able to organize a stronger connection among the outputs and obtain consistent modes. As seen, NAT's points are scattered broadly inside the simplex, indicating that it tends to mix different modes in the outputs.

Therefore, a series of researches (Lee et al., 2018; Ghazvininejad et al., 2019; Qian et al., 2021) are devoted to enhancing the dependencies modeling to alleviate the multi-modality issue. These methods build the dependencies on complex target tokens by partially exposing target tokens as inputs (Qian et al., 2021) during training or employing iterative refinements (Lee et al., 2018) similar to autoregressive models. Although the introduced word dependencies modeling improves the model's performance, we notice that these models heavily rely on an AT model as a teacher to filter complex modes of target tokens.

Another series of researches (Kaiser et al., 2018; Ma et al., 2019; Ran et al., 2019; Shu et al., 2020; Lee et al., 2020; Bao et al., 2021) alleviate the multi-modality issue by introducing latent variables. They aim to extract informative latent variables from the target sentence and take it as a springboard to predict the sentence. These latent variables somewhat determine the target's mode, which helps to reduce the modes. However, the interpretability of latent variables is also limited, while learning the latent variables usually involves a complex network (Ma et al., 2019) or deep transformations (Lee et al., 2020).

In this paper, we propose *inter*-NAT, to introduce explicit word inter-dependencies. More specifically, we extract word inter-relationships (denoted as *interrelation*) from the syntactic dependency tree, which presents clear modification relationships between words and is shown helpful to machine translation tasks (Wang et al., 2019a; Bugliarello and Okazaki, 2020; Li et al., 2017). To our best knowledge, *inter*-NAT is the first work to define clear word inter-dependencies for non-autoregressive decoding.

To acquire the interrelation during inference, we train a non-strict biaffine dependency parser (Dozat and Manning, 2017) as the interrelation predictor. As the interrelation is defined on the words, we further adopt the progressively learning strategy (Qian et al., 2021) by gradually exposing target words to train our interrelation predictor. To incorporate the extracted interrelations into non-autoregressive decoding, we reform a self-attention encoding sublayer.

Experiment results on several machine translation benchmarks show that *inter*-NAT achieves the new state-of-the-art performances, especially in the condition that directly trains NAT models without AT teacher. It further achieves competitive quality while keeping a competitive decoding efficiency by the knowledge distillation (Kim and Rush, 2016) and reranking (Wei et al., 2019).

2 Non-autoregressive Translation

A neural machine translation (NMT) system formulates the translation task as a conditional probability model $p(\boldsymbol{y}|\boldsymbol{x})$, which defines the process of translating the source sentence $\boldsymbol{x} = (x_1, x_2, \cdots, x_m)$ into the target sentence $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)$.

Gu et al. (2018) propose Non-Autoregressive Transformer (NAT), which factorizes the $p(\boldsymbol{y}|\boldsymbol{x})$ by assuming conditional independence among the output tokens:

$$p_{\text{NAT}}(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{n} p_\theta(y_i|\boldsymbol{x}), \tag{1}$$

where θ is the parameters for translation.

To support parallel decoding during inference, NAT models usually parameterize θ with a high-parallelism encoder and decoder implemented with a multi-head attention mechanism (Vaswani et al., 2017; Shaw et al., 2018). Since no previous outputs as decoder inputs, NAT models introduce a series of intuitive mechanisms to determine it, such as copying (Gu et al., 2018; Wei et al., 2019; Bao et al., 2021) or connectionist temporal classification (CTC, Graves, 2012). The most common practice introduces a length predictor (Lee et al., 2018) and `Softcopy` mechanism (Wei et al., 2019).

Length Predictor. Given the contextual representation $E = e_{1:m}$ of $x_{1:m}$ encoded by the NAT encoder, the length predictor models the target sequence length n as:

$$\begin{aligned} p_\phi(n|\boldsymbol{x}) &= p_\phi(\Delta L|\boldsymbol{x}) \\ &= \text{MLP}(\text{mean-pooling}(\boldsymbol{E})), \\ \Delta L &= \text{CLIP}(n-m), \end{aligned} \tag{2}$$

where $\text{MLP}(\cdot)$ is a multi-layer perceptron, $\text{CLIP}(\cdot)$ is used to restrict the difference in $-128 \sim 127$.

Softcopy Inputs. Given the target length n and the source representation $\boldsymbol{E}$, we can initialize the decoder inputs $\boldsymbol{D} = d_{1:n}$ with:

$$\begin{aligned} d_j &= \sum_i^m w_{ij} \cdot e_i, \\ w_{ij} &= \frac{\exp\left[-(i - j \cdot \frac{m}{n})^2\right]}{\sum_{i'}^m \exp\left[-(i' - j \cdot \frac{m}{n})^2\right]}. \end{aligned} \tag{3}$$

Finally, the NAT decoder simultaneously generates target sentence $\boldsymbol{y}$ with the computed $\boldsymbol{D}$ and $\boldsymbol{E}$.

Though existing NAT models remarkably improve inference efficiency, they largely sacrifice translation quality. Zhou et al. (2020) study that implicit dependencies modeling in NAT models are not strong enough and makes them hardly learn the

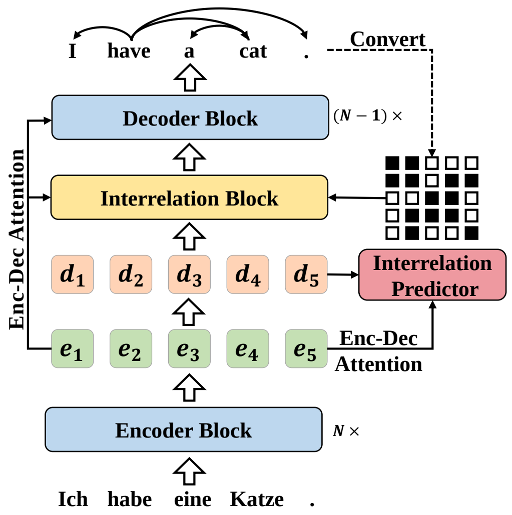


Figure 2: The overall architecture of *inter*-NAT model.

multi-modality phenomenon in datasets directly. It suffers from this issue and generates inferior outputs mixing the multiple modes. In contrast, an AT model can fight this problem by sequential (left to right) modeling, which explicitly models word inter-dependencies by exposing the history output tokens. Inspired by this, we introduce target-side word inter-dependencies for NAT models.

3 Approach

In this section, we propose *inter*-NAT, a non-autoregressive Transformer with target-side word inter-dependencies modeling. More specifically, we introduce the word inter-dependencies (denoted as interrelation $\boldsymbol{M}$) according to syntactic dependency tree to factorize the probability $p(\boldsymbol{y}|\boldsymbol{x})$ as:

$$p(\boldsymbol{y}|\boldsymbol{x}) = p_\gamma(\boldsymbol{M}|\boldsymbol{x}) \prod_{i=1}^{n} p_\theta(y_i|\boldsymbol{x}, \boldsymbol{M}), \quad (4)$$

where γ and θ are the model's parameters, $\boldsymbol{M}$ is extracted from the syntactic dependency tree of the target $\boldsymbol{y}$. It can alleviate the multi-modality problem in two aspects:

(1) Each sentence corresponds to a unique syntactic dependency tree. Providing a target dependency tree essentially reduces one-to-many phenomenon in modeling the target sentence.

(2) The syntactic dependency tree presents the clear modifier-head relations among the words, enhancing word inter-dependency modeling in non-autoregressive decoding.

3.1 Model Overview

Before detailing our proposed method, we first overview the *inter*-NAT. Fig. 2 shows the overall architecture of *inter*-NAT, which works following the NAT fashion:

(i) The encoder encodes source sentence $x_{1:m}$ into the contextual representation $\boldsymbol{E} = e_{1:m}$.

(ii) Given the source representation $\boldsymbol{E}$, length predictor computes the target length n and forms the decoder inputs $\boldsymbol{D} = d_{1:n}$ (§2).

(iii) The target-side interrelation $\boldsymbol{M}$ is predicted by the interrelation predictor or converted from the syntactic dependency tree (§3.2).

(iv) Given the decoder input $\boldsymbol{D}$ and the target-side interrelation $\boldsymbol{M}$, the decoder simultaneously generates all tokens with the help of an interrelation decoder block (§3.3).

3.2 Syntactic Dependencies as Interrelation

Our key insight is to extract the word interrelation using the dependency tree. Li et al. (2017); Zhang et al. (2019); Bugliarello and Okazaki (2020) show the syntactic dependency tree helps to the autoregressive neural machine translation.

Extracting Syntactic Interrelation. Given the syntactic dependency tree $\boldsymbol{t} = (t_1, \cdots, t_n)$ of sentence $\boldsymbol{y} = (y_1, \cdots, y_n)$, we extract the interrelation $\boldsymbol{M} \in \{0, 1\}^{n \times n}$ as follows:

$$\boldsymbol{M}_{ij} = \begin{cases} 1 & \text{if } t_i = j \text{ or } t_j = i \\ 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}, \quad (5)$$

where $t_i = j$ denotes y_j is the head-word of y_i (y_i modified y_j) and $\boldsymbol{M}_{ij}$ denotes the interrelation type between y_i and y_j. We intuitively assume that each token should be interrelated to itself. Fig. 3 shows the dependency tree of the sentence "*I have a cat.*" and its converted interrelation matrix.

Predicting Syntactic Interrelation. To acquire the target-side interrelation matrix during inference, we train a non-strict *Biaffine-Parser* (Dozat and Manning, 2017) as our interrelation predictor[1]:

$$p_\gamma(\boldsymbol{M}|\boldsymbol{x}) = p_\gamma(\boldsymbol{t}|\boldsymbol{x}) = \prod_{i=1}^{n} p_\gamma(t_i|\boldsymbol{x}), \quad (6)$$

where we employs a stacked non-autoregressive decoder block (NA-Dec) and a biaffine neural network (BiaffineNet) to parameterize the γ.

[1]The architecture is shown in Appendix

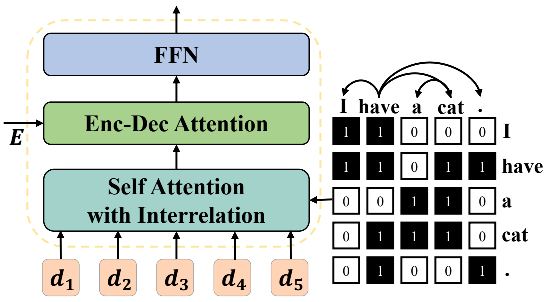


Figure 3: The interrelation block.

Given the source representation $\boldsymbol{E}$ and the decoder inputs $\boldsymbol{D}$, we first compute the score $\boldsymbol{s} \in \mathbb{R}^{n \times n}$ as:

$$\boldsymbol{s} = \text{BiaffineNet}(\text{NA-Dec}([\boldsymbol{D}; \boldsymbol{E}])). \tag{7}$$

Then, we compute the probability $p_\gamma(t_i|\boldsymbol{x})$ with softmax operation:

$$p_\gamma(t_i = j|\boldsymbol{x}) = \frac{\exp{(s_{ij})}}{\sum_{j'=1}^{n} \exp{(s_{ij'})}}. \tag{8}$$

During inference, we can obtain the head index of each decoder input d_i by $\arg\max(\boldsymbol{s})$ operation:

$$\hat{t}_i = \arg\max_{j \in [1,\cdots,n]} p_\gamma(t_i = j|\boldsymbol{x}). \tag{9}$$

Notice that we do not use the MST algorithm (Prim, 1957) due to its run-time cost.

Adaptive Training for Predictor. We experimentally find that directly learning to predict the dependency tree without any target tokens is somewhat tricky. Therefore, we adaptively expose some target tokens as inputs for training predictor.

Inspired by Qian et al. (2021), we determine the number N_{obs} of exposed tokens $\widetilde{\boldsymbol{y}}$ by the interrelation prediction quality:

$$N_{\text{obs}} = f_{\text{ratio}} \cdot \text{dist}(\boldsymbol{t}^*, \boldsymbol{t}) \tag{10}$$

$$\boldsymbol{t}^* = \arg\max(\boldsymbol{s}), \tag{11}$$

where $\text{dist}(\cdot)$ is the distance function, f_{ratio} is the sampling ratio[2]. Then, we randomly sample N_{obs} target tokens as $\widetilde{\boldsymbol{y}}$ and position-wise replace the original decoder inputs $\boldsymbol{D}$ with $\widetilde{\boldsymbol{y}}$.

Finally, we compute the interrelation prediction loss as:

$$\mathcal{L}_{\text{inter}} = -\sum_{i}^{n} \log p_\gamma(t_i|\boldsymbol{x}, \widetilde{\boldsymbol{y}}) \cdot \mathbb{1}[y_i \notin \widetilde{\boldsymbol{y}}], \tag{12}$$

where $\mathbb{1}[\cdot]$ is the indicator function.

[2]Suggested by Qian et al. (2021), we use hamming distance (Hamming, 1950) and linear decrease the ratio from 0.5 to 0.3 in training steps.

3.3 Decoding with Target-side Interrelation

To incorporate target-side interrelation information for modeling $p_\theta(y_i|\boldsymbol{x}, \boldsymbol{M})$, we introduce a specific interrelation block as the first layer of the decoder (Fig. 3).

Inspired by Transformer (Vaswani et al., 2017) or its variant (Shaw et al., 2018) that employs positional encoding as extra inputs to self-attention, we inject the interrelation $\boldsymbol{M}$ into the self-attention sublayer. Given inputs $(r_1, \cdots, r_n)$ and their interrelation matrix $\boldsymbol{M} \in \{0,1\}^{n \times n}$, we compute the self-attention sublayer outputs $(h_1, \cdots, h_n)$ as:

$$\begin{aligned} h_i &= \sum_{j=1}^{n} \alpha_{ij}(r_j W^V + \text{repr}(\boldsymbol{M})_{ij}^V) \\ \alpha_{ij} &= \text{softmax}(w_{ij}) \\ w_{ij} &= \frac{r_i W^Q (r_j W^K + \text{repr}(\boldsymbol{M})_{ij}^K)^T}{\sqrt{d_{\text{model}}}}, \end{aligned} \tag{13}$$

where W^Q, W^K, and W^V are the parameters, $\text{repr}(\boldsymbol{M})_{ij}^V$, $\text{repr}(\boldsymbol{M})_{ij}^K \in \mathbb{R}^{d_{\text{head}}}$ are the trainable representations of interrelation $\boldsymbol{M}_{ij}$. The rest layers of interrelation block keep the same as the original Transformer (Vaswani et al., 2017) decoder block, i.e., followed by an encoder-decoder attention sublayer and a feed-forward sublayer.

After the interrelation block, the remaining $N-1$ decoder blocks stay the same with the relative-position-based Transformer (Shaw et al., 2018).

3.4 Learning

Given the dependency tree $\boldsymbol{t} = \{t_1, t_2, \cdots, t_n\}$ and the target sentence $\boldsymbol{y} = \{y_1, y_2, \cdots, y_n\}$, we first extract interrelation $\boldsymbol{M}$ from $\boldsymbol{t}$ according to Eqn. (5), then compute the translation loss with:

$$\mathcal{L}_{\text{nat}} = -\sum_{i}^{n} \log p_\theta(y_i|\boldsymbol{x}, \boldsymbol{M}). \tag{14}$$

The length predictor is trained by:

$$\mathcal{L}_{\text{len}} = -\log p_\phi(\Delta L|\boldsymbol{x}), \tag{15}$$

where $\Delta L = \text{CLIP}(n - m)$.

Overall Training Objective. Combining the interrelation prediction loss, translation loss, and length prediction loss, the full-fledged training loss is:

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{nat}} + \lambda \mathcal{L}_{\text{inter}} + \alpha \mathcal{L}_{\text{len}}, \tag{16}$$

where λ and α are the hyperparameters used to adjust the importance of each training loss. We follow the previous works and set α to 0.1.

4 Experiments

4.1 Experimental Setup

Dataset. We conduct the experiments on three machine translation datasets: WMT14 English-German task (WMT14 EN-DE, 4.5M sentence pairs), WMT16 English-Romanian task (WMT16 EN-RO, 610K sentence pairs), and IWSLT16 German-English task (IWSLT16 DE-EN, 196K sentence pairs). The datasets are obtained from previous open-source work, such as `fairseq`[3] for WMT14 EN-DE, and Lee et al. (2018) for WMT16 EN-RO and IWSLT16 DE-EN[4]. Following previous practices (Vaswani et al., 2017; Lee et al., 2018), all of the datasets are tokenized with `Moses`[5] and segmented into subword units using BPE encodings (Sennrich et al., 2016). We share the subword embeddings of the source language and target language in each dataset.

The dependency tree of datasets is obtained by the `Stanza` toolkit[6]. Since our method works with the sub-word units, we adapt the words' dependency tree to the sub-word level by an intuitive transformation: the first sub-word unit of a word inherits the head of the word, the head of the remaining units is the first sub-word unit.

Model Settings. In the case of WMT tasks, we follow the base setting ($d_{\text{model}} = 512$, $d_{\text{hidden}} = 2048$, $p_{\text{dropout}} = 0.3$, $n_{\text{head}} = 8$, $d_{\text{head}} = 64$, and $n_{\text{layer}} = 6$) of Vaswani et al. (2017). We use a smaller model setting ($d_{\text{model}} = 256$, $d_{\text{hidden}} = 512$, $p_{\text{dropout}} = 0.3$, $n_{\text{head}} = 4$, $d_{\text{head}} = 64$, and $n_{\text{layer}} = 5$) for IWSLT16.

The parameter is trained with Adam (Kingma and Ba, 2015) optimizer and set $\beta = (0.9, 0.99)$. We use the invert square root learning rate schedule (Vaswani et al., 2017) for WMT tasks and the linear annealing schedule (Lee et al., 2018) from 3.0×10^{-4} to 1.0×10^{-5} for the IWSLT task. The hyperparameters λ in Eqn. (16) to adjust the importance of the interrelation prediction loss are set to 3.0 and 2.0 for WMT and IWSLT tasks, respectively.

[3] `https://github.com/pytorch/fairseq/tree/main/examples/translation`

[4] `https://github.com/nyu-dl/dl4mt-nonauto`

[5] `https://github.com/moses-smt/mosesdecoder`

[6] `https://stanfordnlp.github.io/stanza/depparse.html`. Unlabeled attach scores (%): 86.22 for English, 85.39 for German, and 90.66 for Romanian.

Model	WMT14 EN-DE	WMT14 DE-EN	WMT16 EN-RO	Speedup
CMLM	10.88	/	/	/
SynST	20.74	25.50	/	4.9 ×
Flowseq	20.85	25.40	29.86	1.1 ×
AXE	20.40	24.90	30.47	/
CNAT	21.30	25.73	/	10.4 ×
Transformer †	27.25	31.53	33.97	1.0 ×
NAT †	11.60	16.15	21.40	15.3 ×
GLAT †	16.71	24.78	/	**15.3** ×
inter-NAT †	**21.79**	**27.02**	**30.79**	15.1 ×

Table 1: Performance of the controlled experiments on the test set of WMT tasks. † indicates the results that come from our implementation.

Baselines. Except for the vanilla NAT, we also include several representative NAT as baselines:

- Non-iterative NAT: ENAT (Guo et al., 2019), NAT-REG (Wang et al., 2019b), imitation-NAT (Wei et al., 2019), NAT-DCRF (Sun et al., 2019), AXE (Ghazvininejad et al., 2020a), and GLAT (Qian et al., 2021).
- Latent variable-based NAT: NAT-FT (Gu et al., 2018), LT (Kaiser et al., 2018), Flowseq (Ma et al., 2019), SynST (Akoury et al., 2019), and CNAT (Bao et al., 2021).
- Iterative NAT: CMLM (Ghazvininejad et al., 2019).

Metrics. We compare our model with baselines in terms of translation quality and decoding efficiency. As for translation quality, we evaluate the tokenized and cased BLEU score (Papineni et al., 2002) with `fairseq-score`[7]. As for decoding efficiency, we first measure the decoding latency sentence-by-sentence, then report the relative speedups by comparing it with an autoregressive Transformer model. We obtain the performance of baselines directly using reported in the previous works if available or reproducing them on our datasets using the open-source implementation. We highlight the **best NAT result** in each table.

4.2 Main Results

First, we validate our proposed method under a strict experimental condition, in which all of the NAT models are trained on the raw dataset. The results are listed in Tab. 1.

We can see that *inter*-NAT achieves significant improvements (more than 10 BLEU in most tasks)

[7] `https://github.com/pytorch/fairseq/blob/main/fairseq_cli/score.py`

Model	WMT14 EN-DE	WMT14 DE-EN	WMT16 EN-RO	Speedup
NAT-FT	17.69	21.47	27.29	15.6 ×
LT	19.80	/	/	3.9 ×
Flowseq	23.72	28.39	29.73	1.1 ×
CNAT	25.56	29.36	/	10.4 ×
ENAT	20.65	23.03	30.08	25.3 ×
NAT-REG	20.65	24.77	/	**27.6** ×
imitate-NAT	22.44	25.67	28.61	18.6 ×
NAT-DCRF	23.44	27.22	/	10.4 ×
GLAT	25.21	29.84	31.19	15.3 ×
Transformer †	27.25	31.53	33.97	1.0 ×
inter-NAT †	**26.00**	**30.29**	**32.10**	15.1 ×

Table 2: Performance on the test set of WMT tasks trained with knowledge distillation.

Model	N_c	WMT14 EN-DE	WMT14 DE-EN	WMT16 EN-RO	Speedup
NAT-FT	100	19.17	23.20	29.79	2.4 ×
LT	10	21.00	/	/	/
Flowseq	30	25.31	30.68	32.20	/
CNAT	9	26.60	30.75	/	5.6 ×
ENAT	9	24.28	26.10	34.51	12.4 ×
NAT-REG	9	24.61	28.90	/	**15.1** ×
imitate-NAT	7	24.15	27.28	31.45	9.7 ×
NAT-DCRF	19	26.80	30.04	/	6.1 ×
GLAT	7	26.55	31.02	32.87	7.9 ×
Transformer †	-	27.25	31.53	33.97	1.0 ×
inter-NAT †	7	**27.03**	**31.46**	**33.75**	9.2 ×

Table 3: Performance on the test set of WMT tasks. The results come from length parallel reranking with NAT models trained with knowledge distillation. N_c denotes the number of re-ranked candidates.

over the vanilla NAT, indicating that target-side interrelation modeling can significantly improve the capacity to overcome multi-modality issues. Furthermore, our *inter*-NAT achieves the best results in this setting, demonstrating that decomposing the syntactic dependency information is more helpful to non-autoregressive decoding than chunks (Akoury et al., 2019), latent variables (Ma et al., 2019; Bao et al., 2021), and monotonic alignment assumption (Ghazvininejad et al., 2020a).

With Distillation. The sequence-level knowledge distillation (Kim and Rush, 2016) can directly filter the multi-modality phenomenon that exists in datasets, which becomes common practice in non-autoregressive machine translation tasks (Gu et al., 2018). Therefore, we train a Transformer model and take it as the teacher to distill the training data.

As shown in Tab. 2 , all of NAT models improve the performance with a large margin by employing a Transformer as the distilled teacher. Moreover, we can see that *inter*-NAT outperforms all the NAT models in all tasks, indicating the benefits of interrelation modeling.

With Reranking. To further improve translation quality, we also introduce the length parallel decoding (Wei et al., 2019) for *inter*-NAT. During inference, *inter*-NAT first simultaneously generates N_c candidates with different lengths, then selects the best output via re-scoring using the teacher.

We can see from Tab. 3 that *inter*-NAT achieves the best translation quality by equipping with the length parallel reranking, narrowing the performance gap between the non-autoregressive decoding and autoregressive decoding.

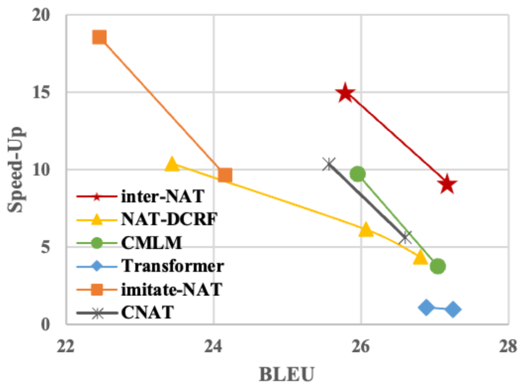


Figure 4: BLEU and decoding speed-up of NAT models on WMT14 EN-DE test set. Each point represents the decoding method run with its corresponding setting in Tab. 2, Tab. 3, or iterative refinements. Notice that we only include the results evaluated with 1080TI-GPU for fair comparisons.

Efficiency. Fig. 4 displays the trend of decoding speed-up and BLEU with different models. As seen, *inter*-NAT is located on the top-right of the baselines. It is shown that *inter*-NAT achieves higher BLEU if we fixed the Speed-Up and faster Speed-Up if we fixed the BLEU, indicating that *inter*-NAT outperforms baselines. Although CMLM (Ghazvininejad et al., 2019) achieves competitive BLEU scores, they sacrificed the speed advantages. In contrast, *inter*-NAT has a better trade-off that achieves a competitive performance while maintaining remarkable speed advantages.

4.3 Analysis

Incorporating the interrelation helps for the multi-modality problem. The multi-modality

Data	Distance$^{\downarrow}$	Token-C	Sentence-C
Raw data	0.21	2.71	3.36
Transformer	0.21	2.06	2.61
NAT	0.41	2.22	2.89
inter-NAT	0.25	1.62	2.18

Table 4: Complexity C ($\uparrow$ more complex) and average euclidean distance of outputs from different settings.

Methods	**Repetition Ratio**$^{\downarrow}$(%)	
	EN-DE	**DE-EN**
Transformer	0.06	0.01
NAT	25.02	21.67
NAT w/ distillation	6.26	5.87
inter-NAT	2.85	0.82
inter-NAT w/ distillation	**0.43**	**0.47**

Table 5: Token repetition ratio (%) of outputs from different methods on WMT14 test set.

phenomenon is unavoidable in machine translation tasks, which is a challenging problem for non-autoregressive decoding. To validate that target-side interrelation is beneficial to overcome the multi-modality problem, we follow Zhou et al. (2020) and synthesize a one-to-many translation dataset to analyze this issue.

Dataset: We construct the dataset by extracting the sentences aligned in English-German, English-French, and English-Spanish corpus[8] and processing the dataset following common practices, including tokenization with `Moses`, segmentation with BPE units (Sennrich et al., 2016), and parsing with `Stanza`, etc. In such a case, each source sentence has three different languages reference in the dataset, representing three modes. During inference, we do not supply the language signal.

Results: As illustrated in Fig. 1, a vanilla NAT model can hardly coordinate the mode (language) signal itself and always hybrid the different modes in outputs. By introducing the target-side interrelation information, *inter*-NAT well organizes the non-autoregressive decoding, resulting in a more consistent mode in its outputs.

To analyze the multi-modality problem quantitatively, we compute the average Euclidean distance between the point to its nearest vertex for each model. Besides, we follow Zhou et al. (2020) and utilize corpus *complexity* C as evaluated metrics. As shown in Tab. 4, our *inter*-NAT heavily reduces the complexity of the dataset, which is consistent

[8] `https://www.statmt.org/europarl/`

Methods	**extracted** M	**predicted** M	
	BLEU$^{\uparrow}$	BLEU$^{\uparrow}$	*inter* F1$^{\uparrow}$
NAT		18.01	
inter-NAT			
w/ dependency	42.61	**29.88**	**60.35**
w/ adjacent	22.65	/	/
w/ co-occurrence	**56.21**	23.45	20.17

Table 6: Performance on IWSLT16 DE-EN valid set with different interrelations.

with our qualitative analysis.

***inter*-NAT overcomes repeat-translation problem.** Tab. 5 analyzes the repetition ratio of translation on the test set of WMT14 tasks. Suffering from the multi-modality problem, vanilla NAT has the highest repetition ratio of the outputs. Incorporating the interrelation information or training with the knowledge distillation can reduce the repetition ratio. We can also find that combining target-side interrelation modeling and distillation are "compatible" and enable the NAT model to achieve the lowest repetition rate. The above observation is also consistent with the qualitative analysis and quantitative analysis on multi-modality problems.

Syntactic dependency well balances the prediction accuracy and translation quality. To analyze which kind of interrelation is better for non-autoregressive decoding, we further compare the syntactic dependency and two intuitive relations (word-*adjacent* and word *co-occurrence* relation, we include more details in Appendix A). We can see from Tab. 6 that NAT models always benefit from the introduced interrelation and improve the translation quality, whatever the interrelation is. There is also a trade-off in the table: even though the NAT models that incorporate the extracted word co-occurrence information achieve the highest BLEU score, their predicted performance (BLEU and *inter* F1) is relatively low; the word-adjacent relation obtains the highest *inter* F1 score, it is of little help to the NAT model. In comparison, the syntactic dependency well balances the prediction accuracy and translation quality.

Module Ablation. As shown in Tab. 7, our *inter*-NAT can benefit from both the training and integrating the target-side interrelation.

- *inter*-NAT well regularizes the encoder. Extracting the target-side interrelation as the training supervision, the NAT can regularize

Methods	BLEU↑
NAT	18.01
+ $\mathcal{L}_{\text{inter}}$	23.01 (+5.00)
+ $\mathcal{L}_{\text{inter}}$ + Interrelation Block	29.88 (+11.87)

Table 7: Performance on IWSLT16 DE-EN valid set.

Pretrained Encoder	Finetuning Decoder	
	NAT	*inter*-NAT
NAT	18.69	23.94
inter-NAT	22.49 (**+3.80**)	29.76 (**+5.68**)

Table 8: The BLEU score on IWSLT16 DE-EN valid set of different decoders training with a fixed encoder from the pretrained model (NAT or *inter*-NAT).

the encoder output by the multi-task learning and improve the performance by 5.0 BLEU (23.01 vs. 18.01). The results listed in Tab. 8 further validate this observation.

- By integrating the target-side interrelation with the introduced interrelation block, the *inter*-NAT achieves over 6.50 BLEU points improvement.

We further apply *inter*-NAT to the syntax-aware machine translation task and validate the effectiveness of the interrelation block. The details are in Appendix B.

More Analysis. We also provide more details about interrelation and several translation examples in Appendix.

5 Related Work

5.1 Non-autoregressive Machine Translation

Gu et al. (2018) propose the non-autoregressive Transformer (NAT) in machine translation task, remarkably improving the decoding efficiency at the cost of translation quality. Then, a series of improvements has been proposed.

A line of works propose to learn from Transformer to regularize the attention matrix (Li et al., 2019) and hidden states (Wei et al., 2019) of NAT models. Some works carefully design training objectives to overcome the *multi-modality problem*, such as latent alignment for cross-entropy (Libovický and Helcl, 2018; Saharia et al., 2020), bag-of-words objectives (Shao et al., 2020) or energy-based objectives (Tu et al., 2020). Our works impose a dependency tree as the regularized objective, also improving performance.

Another series of studies propose to enhance dependencies modeling to tackle the multi-modality issues. Such as multiple iterative refinements(Lee et al., 2018; Ghazvininejad et al., 2019; Guo et al., 2020) or masked language models (Qian et al., 2021). In comparison, our method utilizes the syntactic dependency tree to define clear word inter-dependencies for non-autoregressive translation and shows its help in experiments.

Some works propose to decomposing the target-side information by the latent variables (Ma et al., 2019; Ran et al., 2019; Bao et al., 2021). Unlike them, we introduce the target-side dependency tree as the decomposed goal, which is well-define and easy to learn.

5.2 Syntax-Aware Machine Translation

Our work is also related to syntax-aware translation. Sennrich and Haddow (2016) study that integrating the syntax representation into word embedding for machine translation. Wang et al. (2019a); Bugliarello and Okazaki (2020); Chen et al. (2017) propose incorporating the syntax tree information into the encoder to model the sentence's latent structure. Unlike these researches, we incorporate the dependency information into the NAT models and works on the target side decoding.

Most close to our work is SynST (Akoury et al., 2019), which autoregressively predicts the syntactic chunk sequence and integrates it in non-autoregressive translation. In contrast, our model predicts the dependency interrelation following a non-autoregressive fashion, improving translation quality and decoding efficiency.

6 Conclusion

In this paper, we propose *inter*-NAT, which models the target-side syntactic dependency interrelation for non-autoregressive decoding. Specifically, *inter*-NAT extracts the target-side interrelation from the dependency tree (ground-truth dependency tree during training or predicted dependency graph during inference) then injects it into the self-attentive sublayer in the decoder. Experiments results show that *inter*-NAT benefits from the clear syntactic relations between words presented in the syntactic dependency tree, achieving a better NAT model. We also consider exploring more effective interrelation for NAT models in our future work.

References

Nader Akoury, Kalpesh Krishna, and Mohit Iyyer. 2019. Syntactically supervised transformers for faster neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1269–1281, Florence, Italy. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yu Bao, Shujian Huang, Tong Xiao, Dongqi Wang, Xinyu Dai, and Jiajun Chen. 2021. Non-autoregressive translation by learning target categorical codes. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5749–5759, Online. Association for Computational Linguistics.

Emanuele Bugliarello and Naoaki Okazaki. 2020. Enhancing machine translation with dependency-aware self-attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1618–1627, Online. Association for Computational Linguistics.

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 2017. Improved neural machine translation with a syntax-aware encoder and decoder. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1936–1945, Vancouver, Canada. Association for Computational Linguistics.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. Controllable paraphrase generation with a syntactic exemplar. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5972–5984, Florence, Italy. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.

Marjan Ghazvininejad, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy. 2020a. Aligned cross entropy for non-autoregressive machine translation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3515–3523. PMLR.

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 6112–6121, Hong Kong, China. Association for Computational Linguistics.

Marjan Ghazvininejad, Omer Levy, and Luke Zettlemoyer. 2020b. Semi-autoregressive training improves mask-predict decoding. *ArXiv preprint*, abs/2001.08785.

Alex Graves. 2012. Connectionist temporal classification. In *Supervised Sequence Labelling with Recurrent Neural Networks*, pages 61–93. Springer.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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 *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3723–3730. AAAI Press.

Junliang Guo, Linli Xu, and Enhong Chen. 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*, pages 376–385, Online. Association for Computational Linguistics.

Richard W Hamming. 1950. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160.

Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. 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*, volume 80 of *Proceedings of Machine Learning Research*, pages 2395–2404. PMLR.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the*

2016 Conference on Empirical Methods in Natural Language Processing, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.

Jason Lee, Raphael Shu, and Kyunghyun Cho. 2020. Iterative refinement in the continuous space for non-autoregressive neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1006–1015, Online. Association for Computational Linguistics.

Junhui Li, Deyi Xiong, Zhaopeng Tu, Muhua Zhu, Min Zhang, and Guodong Zhou. 2017. Modeling source syntax for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 688–697, Vancouver, Canada. Association for Computational Linguistics.

Zhuohan Li, Zi Lin, Di He, Fei Tian, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Hint-based training for non-autoregressive machine translation. 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 5708–5713, Hong Kong, China. Association for Computational Linguistics.

Jindřich Libovický and Jindřich Helcl. 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*, pages 3016–3021, Brussels, Belgium. Association for Computational Linguistics.

Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. 2019. FlowSeq: Non-autoregressive conditional sequence generation with generative flow. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4282–4292, Hong Kong, China. Association for Computational Linguistics.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Robert Clay Prim. 1957. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2021. Glancing transformer for non-autoregressive neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1993–2003, Online. Association for Computational Linguistics.

Qiu Ran, Yankai Lin, Peng Li, and Jie Zhou. 2019. Guiding non-autoregressive neural machine translation decoding with reordering information. *ArXiv preprint*, abs/1911.02215.

Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. 2020. Non-autoregressive machine translation with latent alignments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1098–1108, Online. Association for Computational Linguistics.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 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)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Chenze Shao, Jinchao Zhang, Yang Feng, Fandong Meng, and Jie Zhou. 2020. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 198–205.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Raphael Shu, Jason Lee, Hideki Nakayama, and Kyunghyun Cho. 2020. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhiqing Sun, Zhuohan Li, Haoqing Wang, Di He, Zi Lin, and Zhi-Hong Deng. 2019. Fast structured decoding for sequence models. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3011–3020.

Zhiqing Sun and Yiming Yang. 2020. An EM approach to non-autoregressive conditional sequence generation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9249–9258. PMLR.

Lifu Tu, Richard Yuanzhe Pang, Sam Wiseman, and Kevin Gimpel. 2020. ENGINE: Energy-based inference networks for non-autoregressive machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2819–2826, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. 2019a. Self-attention with structural position representations. 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 1403–1409, Hong Kong, China. Association for Computational Linguistics.

Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019b. Non-autoregressive machine translation with auxiliary regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5377–5384.

Bingzhen Wei, Mingxuan Wang, Hao Zhou, Junyang Lin, and Xu Sun. 2019. Imitation learning for non-autoregressive neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1304–1312, Florence, Italy. Association for Computational Linguistics.

Meishan Zhang, Zhenghua Li, Guohong Fu, and Min Zhang. 2019. Syntax-enhanced neural machine translation with syntax-aware word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1151–1161, Minneapolis, Minnesota. Association for Computational Linguistics.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020. Understanding knowledge distillation in non-autoregressive machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A Interrelation Details

Interrelation architecture. Fig. 1 shows the module details of our interrelation predictor.

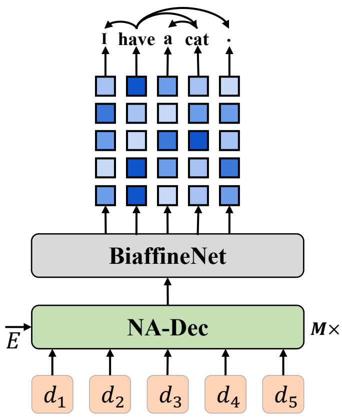


Figure 1: The module details of interrelation predictor.

Adjacent Interrelation. As shown in Fig. 2, we intuitively take the adjacent relation as an interrelation among the target outputs.

	I	have	a	cat	.
I	1	1	0	0	0
have	1	1	1	0	0
a	0	1	1	1	0
cat	0	0	1	1	1
.	0	0	0	1	1

Figure 2: The adjacent interrelation matrix.

Co-occurrence Interrelation. We also propose to extract the interrelation from the co-occurrence information. It represents the frequency of token pairs that appeared together in a sentence, which may help model the target sentence. We compute the co-occurrence matrix on the training dataset and decide the interrelated relation with a controlled ratio. In our experiments, we set it to 0.1. Notice that the co-occurrence relation is not symmetrical, as we choose its most frequent token in the sentence for each token. The example is shown in Fig. 3.

Interrelation-related Metrics. To further analyze the interrelation's influence, we compute each model's interrelation metric. Specifically, extracting the interrelation reference from the target, we compute the recall ratio of the models' output.

We can see from Tab. 1 that the interrelation recall is high related to the models' performance.

Methods	Interrelation Recall (%)			BLEU
	Adjacent	Co-occurrence	Dependency	
AT	43.29	28.25	38.36	27.25
NAT	24.03	19.15	20.88	11.60
inter-NAT	37.40	25.11	32.65	21.78

Table 1: Interrelation recall of different models on WMT14 EN-DE test set.

Methods	**BLEU**
NAT	18.01
w/ CTC	29.87
inter-NAT	29.88
w/ CTC	31.02

Table 2: Performance on IWSLT16 DE-EN valid set..

With the interrelation recall improving, the model achieves a better BLEU score.

Compatibility with CTC. We integrate the CTC loss into *inter*-NAT to verify the compatible, and the result is shown in 2. With the help of CTC loss, *inter*-NAT achieves over 1 BLEU points improvement.

B Syntax-Controllable Translation

Since our model explicitly incorporates the syntax-based interrelation into non-autoregressive decoding, we can apply it to a syntax-guided translation task (Tab. 3). Inspired by previous text transfer studies (Chen et al., 2019), we utilize a multi-reference translation dataset to avoid the mismatch between source semantics and target syntax[1]. While decoding, we feed the source inputs and its target-side interrelation extracted from the syntax reference (denote as Ref_{dep}) to the decoder and generates the outputs. We compute the BLEU score by comparing the outputs to its syntax references.

Dataset. We apply the *inter*-NAT to the syntax guided translation tasks on the LDC Chinese-English[2] (denote as LDC ZH-EN, 1.3M sentence pairs) and NIST ZH-EN dataset (MT03 as dev the set, MT05 as the test set). Each sentence in the NIST test set has multiple references. We can evaluate the model's controllable translation by given different references as the syntax providers. We use `NLPIRICTCLAS`[3] and Moses tokenizer for

[1]NIST MT05 is a test set with multiple references for each sentence to evaluate the LDC Chinese-English translation task.

[2]LDC2002E18, LDC2003E14, LDC004T08, and LDC2005T06

[3]http://ictclas.nlpir.org/

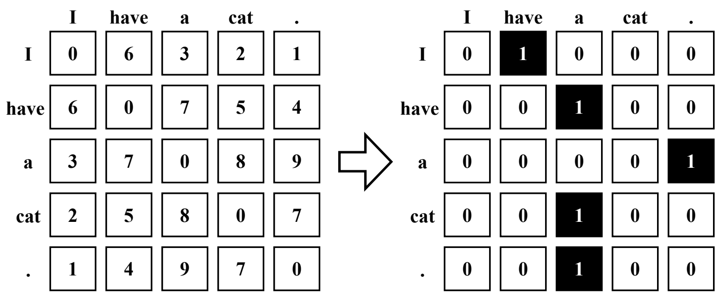


Figure 3: The co-occurrence interrelation matrix. We extract the interrelation according to the co-occurrence number of token pairs in datasets.

Chinese and English tokenization, respectively.

Results. We see in Tab. 3 that the highest BLEU scores are all located in the diagonal of table, which indicates that *inter*-NAT can generate syntactically controllable translation results. We also provide few examples of syntax-guided translation in Tab. 5.

BLEU / Ref_{dep}	$\text{BLEU}_{\text{Ref-0}}$	$\text{BLEU}_{\text{Ref-1}}$	$\text{BLEU}_{\text{Ref-2}}$	$\text{BLEU}_{\text{Ref-3}}$
Ref-0	**33.37**	23.04	22.01	23.40
Ref-1	22.04	**35.98**	23.60	23.27
Ref-2	21.64	24.80	**34.90**	22.68
Ref-3	22.95	24.00	22.60	**34.66**

Table 3: Performance of syntax-controllable translation on the NIST MT05 test set.

Source	Gutach: Noch mehr Sicherheit für Fußgänger
Reference	Gutach: Increased safety for pedestrians
NAT	Gutach: More More safety for pedestrians
***inter*-NAT**	Gutach : More Security for pedestrians
Source	Jazz und Klassik gehören gerade am Jazzstandort Stuttgart zusammen.
Reference	Jazz and classical music belong together at the jazz location of Stuttgart.
NAT	Jazz and classical cs are right together at the Stuttgart of Stuttgart.
***inter*-NAT**	Jazz and classical music belong together at the jazz location of Stuttgart.
Source	Das wäre in Amerika als Medizinstudent im zweiten Jahr niemals möglich.
Reference	That's not something you'd ever get to do in America as a second-year medical student.
NAT	This would never be America America America America a a student in the second year.
***inter*-NAT**	That would never be possible in America as a medical student in the second year.

Table 4: Examples for our *inter*-NAT model trained on the WMT14 DE-EN raw dataset.

Source	阿尔泰共和国位于西伯利亚边缘的多山地带。
Reference1	the altai republic is located in a mountainous region on the southern fringes of siberia.
Output1	the altay republic is situated in a mountain area on the cre, close of siberia.
Reference2	the altai republic is located in the mountainous region on the southern border of siberia.
Output2	the altay republic is situated in the multi-anshan area on the fringe of siberia.
Reference3	altai republic is located in the mountainous region on the fringes of siberia.
Output3	altay republic is situated in the multianshan fringe of kilometers west of siberia.
Reference4	the altai republic is located in a mountainous region on the fringes of siberia.
Output4	the altay republic is situated in the multi-anshan region at crelines of siberia.
Source	乌克兰全国有超过三万三千个投开票所。
Reference1	there are more than 33,000 polling stations in ukraine.
Output1	there were more than 33,000 opening votes in ukraine.
Reference2	there are more than 33,000 polls throughout ukraine.
Output2	there were more than 33,000 tickets in ukraine.
Reference3	there were over 33,000 polling precincts in ukraine.
Output3	there were over 33,000 opening tickets in ukraine.
Reference4	there are over 33,000 polling stations throughout ukraine.
Output4	there were over 33,000 opening tickets in ukraine.

Table 5: Examples of syntax-guided translation. We show all words in lower case.