Multi-Granularity Optimization for Non-Autoregressive Translation

Yafu Li[♠♡], **Leyang Cui**[♣†], **Yongjng Yin**[♠♡], **Yue Zhang**[♡♢†]
[♠] Zhejiang University
[♡] School of Engineering, Westlake University
[♣] Tencent AI lab
[♢] Institute of Advanced Technology, Westlake Institute for Advanced Study
yafuly@gmail.com *leyangcui@tencent.com*
yinyongjing@westlake.edu.cn *yue.zhang@wias.org.cn*

Abstract

Despite low latency, non-autoregressive machine translation (NAT) suffers severe performance deterioration due to the naive independence assumption. This assumption is further strengthened by cross-entropy loss, which encourages a strict match between the hypothesis and the reference token by token. To alleviate this issue, we propose multi-granularity optimization for NAT, which collects model behaviors on translation segments of various granularities and integrates feedback for back-propagation. Experiments on four WMT benchmarks show that the proposed method significantly outperforms the baseline models trained with cross-entropy loss, and achieves the best performance on WMT'16 En⇔Ro and highly competitive results on WMT'14 En⇔De for fully non-autoregressive translation.

1 Introduction

Neural machine translation (NMT) systems have shown superior performance on various benchmark datasets (Vaswani et al., 2017; Edunov et al., 2018a). In the training stage, NMT systems minimize the token-level cross-entropy loss between the reference sequence and the model hypothesis. During inference, NMT models adopt autoregressive decoding, where the decoder generates the target sentence token by token ($O(N)$). To reduce the latency of NMT systems, Gu et al. (2018) propose non-autoregressive neural machine translation (NAT), which improves the decoding speed by generating the entire target sequence in parallel ($O(1)$).

Despite low latency, without modeling the target sequence history, NAT models tend to generate translations of low quality (Gu et al., 2018; Sun et al., 2019; Ghazvininejad et al., 2019). NAT ignores inter-token dependency and naively factorizes the sequence-level probability as a product of independent token probability. However, vanilla

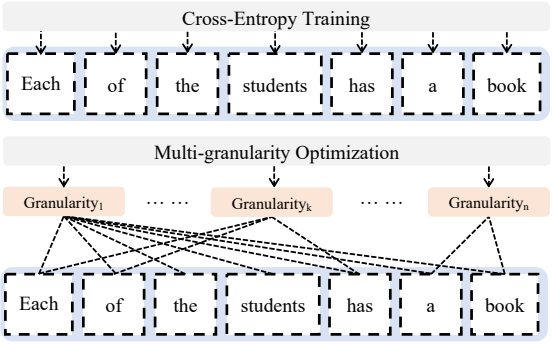

Figure 1: An illustration of modeling the multi-granularity token-dependency beyond cross-entropy.

NAT adopts the same training optimization method as autoregressive (AT) models, i.e., cross-entropy loss (XE loss), which forces the model to learn a strict position-to-position mapping, heavily penalizing hypotheses that suffer position shifts but share large similarity with the references. Given a reference "she left her keys yesterday .", an inappropriate hypothesis "she left her her her ." can yield a lower cross-entropy than one reasonable hypothesis "yesterday she left her keys .". Autoregressive models suffer less from the issue by considering previous generated tokens during inference, which is however infeasible for parallel decoding under the independence assumption. As a result, NAT models trained using cross-entropy loss are weak at handling multi-modality issues and prone to token repetition mistakes (Sun et al., 2019; Qian et al., 2021; Ghazvininejad et al., 2020).

Intuitively, generating adequate and fluent translations involves resolving dependencies of various ranges (Yule, 2006). For example, to generate a translation "Each of the students has a book", the model needs to consider the local n-gram pattern "a - book", the subject-verb agreement across the non-continuous span "each - has", and the global context. To capture the token dependency without

[†]Corresponding authors.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 5073–5084
December 7-11, 2022

the language model, feedback on model's behavior on text spans of multiple granularities can be incorporated. To this end, we propose a multi-granularity optimization method to provide NAT models with rich feedback on various text spans involving multi-level dependencies. As shown in Figure 1, instead of exerting a strict token-level supervision, we evaluate model behavior on various granularities before integrating scores of each granularity to optimize the model. In this way, for each sample we highlight different parts of the translation, e.g., "a book" or "each of the students has".

During training, instead of searching for a single output for each source sequence, we explore the search space by sampling a set of hypotheses. For each hypothesis, we jointly mask part of the tokens and those of the gold reference at the same positions. To directly evaluate each partially masked hypothesis, we adopt metric-based optimization (Ranzato et al., 2016; Shen et al., 2016) which rewards the model with a metric function measuring hypothesis-reference text similarity. Since both the hypothesis and the reference share the same masked positions, the metric score of each sample is mainly determined by those exposed segments. Finally, we weigh each sample score by the model confidence to integrate the metric feedback on segments of various granularities. An illustrative representation is shown in Figure 2, where a set of masked hypothesis-reference pairs are sampled and scored respectively before being merged by segment probabilities. In this way, the model is optimized based on its behavior on text spans of multiple granularities for each training instance within a single forward-backward pass.

We evaluate the proposed method across four machine translation benchmarks: WMT14 En⇔De and WMT16 En⇔Ro. Results show that the proposed method outperforms baseline NAT models trained with XE loss by a large margin, while maintaining the same inference latency. The proposed method achieves two best performances for fully non-autoregressive models among four benchmarks, and obtains highly competitive results compared with the AT model. To the best of our knowledge, we are the first to leverage multi-granularity metric feedback for training NAT models. Our code is released at https://github.com/yafuly/MGMO-NAT.

2 Method

We first briefly introduce some preliminaries including non-autoregressive machine translation (Section 2.1) and cross-entropy (Section 2.2), and then we elaborate our proposed method where the model learns segments of different granularities for each instance (Section 2.3).

2.1 Non-autoregressive Machine Translation (NAT)

The machine translation task can be formally defined as a sequence-to-sequence generation problem, where the model generates the target language sequence $\boldsymbol{y} = \{y_1, y_2, ..., y_T\}$ given the source language sequence $\boldsymbol{x} = \{x_1, x_2, ..., x_S\}$ based on the conditional probability $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$ ($\boldsymbol{\theta}$ denotes the model parameters). Autoregressive neural machine translation factorizes the conditional probability to: $\prod_{t=1}^{T} p(y_t|y_1, ..., y_{t-1}, \boldsymbol{x})$. In contrast, non-autoregressive machine translation (Gu et al., 2018) ignores the dependency between target tokens and factorizes the probability as $\prod_{t=1}^{T} p(y_t|\boldsymbol{x})$, where tokens at each time step are predicted independently.

2.2 Cross Entropy (XE)

Similar to AT models, vanilla NAT models are typically trained using the cross entropy loss:

$$\mathcal{L}_{XE} = -logp(\boldsymbol{y}|\boldsymbol{x}) = -\sum_{t=1}^{T} logp_{\boldsymbol{\theta}}(y_t|\boldsymbol{x}) \quad (1)$$

In addition, a loss for length prediction during inference is introduced:

$$\mathcal{L}_{length} = -logp_{\boldsymbol{\theta}}(T|\boldsymbol{x}) \quad (2)$$

Ghazvininejad et al. (2019) adopt masking scheme in masked language models and train the NAT models as a conditional masked language model (CMLM):

$$\mathcal{L}_{CMLM} = -\sum_{y_t \in \mathcal{Y}(\boldsymbol{y})} logp_{\boldsymbol{\theta}}(y_t|\Omega(\boldsymbol{y}, \mathcal{Y}(\boldsymbol{y})), \boldsymbol{x}) \quad (3)$$

where $\mathcal{Y}(\boldsymbol{y})$ is a randomly selected subset of target tokens and Ω denotes a function that masks a selected set of tokens in $\mathcal{Y}(\boldsymbol{y})$. During decoding, the CMLM models can generate target language sequences via iteratively refining translations from previous iterations.

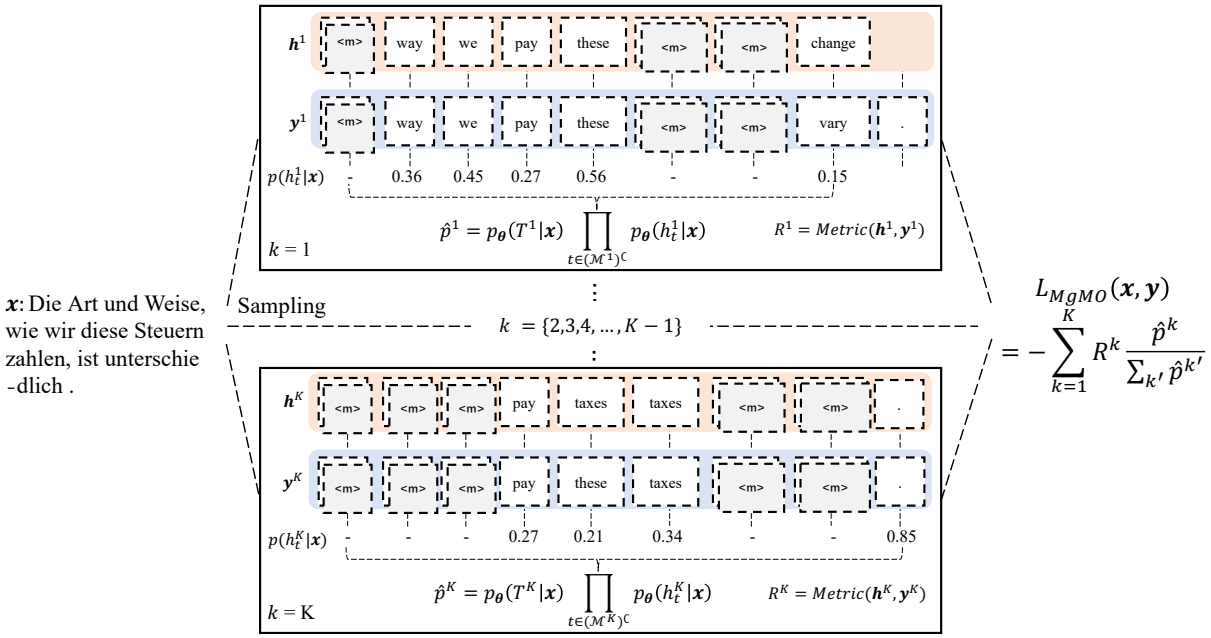

Figure 2: Method illustration of multi-granularity optimization for NAT. During training, our method (MgMO) samples K hypotheses for each source sequence, and focuses on different parts of each one by applying the random masking strategy. For example, MgMO collects model's performance on the partially exposed segments "way we pay these" and "change" for the first hypothesis ($\boldsymbol{h}^1$), while pays more attention to the phrase "pay these taxes" for the K-th one ($\boldsymbol{h}^K$).

2.3 Multi-granularity Optimization for NAT

We propose multi-granularity optimization which integrates feedback on various types of granularities for each training instance. The overall method illustration is presented in Figure 2.

Sequence Decomposition In order to obtain output spans of multiple granularities, we sample K output sequences from the model following a two-step sampling process. In particular, we first sample a hypothesis length and then sample the output token at each time step independently given the sequence length. The probability of the k-th hypothesis $\boldsymbol{h}^k$ is calculated as:

$$p_{\boldsymbol{\theta}}(\boldsymbol{h}^k|\boldsymbol{x}) = p_{\boldsymbol{\theta}}(T^k|\boldsymbol{x}) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(h_t^k|\boldsymbol{x}) \quad (4)$$

To highlight different segments of multiple granularities for each sample, we apply a masking strategy that randomly masks a subset of the tokens for both the hypothesis and the reference at the same position. We denote the masked hypothesis and reference as $\boldsymbol{h}^k = \{h_1^k, \ldots, h_{T^k}^k\}$ and $\boldsymbol{y}^k = \{y_1^k, \ldots, y_T^k\}$, respectively, and denote the set of masked positions as $\mathcal{M}^k$. Note that the reference length T may be different from the hypothesis length T^k.

For the first hypothesis output ($k = 1$) in Figure 2, given the randomly generated masked position set $\mathcal{M}^1 = \{1, 6, 7\}$, the masked hypothesis and reference are $\boldsymbol{h}^1 = \{\langle m\rangle, h_2^1, h_3^1, ..., \langle m\rangle, \langle m\rangle, h_8^1\}$ and $\boldsymbol{y}^1 = \{\langle m\rangle, y_2^1, y_3^1, ..., \langle m\rangle, \langle m\rangle, y_8^1, y_9^1\}$, respectively, where $\langle m\rangle$ represents the masked token. To determine the number of masked tokens $|\mathcal{M}^k|$ for each training instance, we first sample a threshold τ from a uniform distribution $\mathcal{U}(0, \gamma)$, and computes $|\mathcal{M}^k|$ as follows:

$$|\mathcal{M}^k| = max(\lfloor T^k - \tau * T^k \rfloor, 0) \quad (5)$$

where γ is a scaling ratio that controls the likelihood of being masked for each token. Note that the value of $|\mathcal{M}^k|$ lies within the range $[0, T^k - 1]$, meaning that at least one token is kept.

In this way, we decompose each training instance into K pairs of masked hypotheses and references with different granularities exposed. For example, in the last sample ($k = K$) in Figure 2, only the verb phrase "*pay these taxes*" and the period (".") are learned by the model, whereas the sample ($k = 1$) reveals more informative segments.

Metric-based Optimization (MO) To avoid the strict mapping of XE loss (Ghazvininejad et al., 2020; Du et al., 2021), we incorporate metric-based

optimization which is widely studied in reinforcement learning for NMT (Ranzato et al., 2016; Shen et al., 2016; Kiegeland and Kreutzer, 2021; Wu et al., 2018). Metric-based optimization allows more flexibility of token positions by rewarding the model with global scores instead of forcing it to fit a strict position-to-position mapping under XE.

The objective of metric-based optimization is to maximize the expected reward with respect to the posterior distribution given the parameters $\boldsymbol{\theta}$. The reward $\Re(\boldsymbol{\theta})$ for a training instance can be formally written as:

$$\Re(\boldsymbol{\theta}) = \sum_{\boldsymbol{h} \in \mathcal{H}(\boldsymbol{x})} p_{\boldsymbol{\theta}}(\boldsymbol{h}|\boldsymbol{x}) R(\boldsymbol{h}, \boldsymbol{y}) \quad (6)$$

where $\mathcal{H}(\boldsymbol{x})$ denotes the set of all possible candidate hypotheses for the source sequence $\boldsymbol{x}$. $R(\boldsymbol{h}, \boldsymbol{y})$ denotes the metric function that measures the similarity between the hypothesis and the reference under a specific evaluation metric, e.g., GLEU (Wu et al., 2016).

Multi-granularity Metric-based Optimization (MgMO) Despite a lower gap between training and evaluation under the metric-based optimization, it suffers relatively coarse training signals as rewards are obtained by measuring sequence-level similarity. However, due to lack of explicit token dependency, NAT requires more fine-grained feedback for capturing complex token dependencies that can spread across various lengths, e.g., continuous local dependencies and subject-verb agreement across the non-continuous spans. We enrich the metric feedback in Equation 6 by decomposing a single sequence-level reward into multi-granularity evaluation. For each training instance, the model is optimized by integrated feedback on its performance on various sequence segments of diverse granularities.

As enumerating segments of all possible granularities is intractable, we consider a finite set of sampled hypotheses to traverse as many granularities as possible. Combining with sequence decomposition, we can rewrite the Equation 6 into a form of loss function:

$$\mathcal{L}(\boldsymbol{\theta}) = - \sum_{\boldsymbol{h}^k \in \mathcal{K}(\boldsymbol{x})} p_{\boldsymbol{\theta}}(\boldsymbol{h}^k|\boldsymbol{x}) R(\boldsymbol{h}^k, \boldsymbol{y}^k) \quad (7)$$

where $\mathcal{K}(\boldsymbol{x})$ is a sampled subset consisting of K sampled hypotheses. Applying the log derivative trick, the gradient can be derived:

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = - \sum_{\boldsymbol{h}^k \in \mathcal{K}(\boldsymbol{x})} [R(\boldsymbol{h}^k, \boldsymbol{y}^k) \nabla_{\boldsymbol{\theta}} log p_{\boldsymbol{\theta}}(\boldsymbol{h}^k|\boldsymbol{x})] \quad (8)$$

which does not require differentiation of the metric function R. Since both the hypothesis and the reference are partially masked at the same positions, the exposed segments exert much larger effects on the metric function $R(\boldsymbol{h}^k, \boldsymbol{y}^k)$, i.e., the dissimilarity only origins from the unmasked tokens.

In order to make the model focus on the exposed segments, we transform the sequence probability in Equation 7 into the probability of the segment co-occurrence. Since the output distributions of tokens are independent with each other in NAT, the segment co-occurrence probability is simply the multiplication of the probability of each unmasked token:

$$\hat{p}_{\boldsymbol{\theta}}(\boldsymbol{h}^k|\boldsymbol{x}) = p_{\boldsymbol{\theta}}(T^k|\boldsymbol{x}) \prod_{t \in (\mathcal{M}^k)^{\complement}} p_{\boldsymbol{\theta}}(h_t^k|\boldsymbol{x}) \quad (9)$$

where $(\mathcal{M}^k)^{\complement}$ denotes the complementary set of the masked position set $\mathcal{M}^k$.

Within the sample space, the sequence probability can be renormalized by model confidence (Och, 2003; Shen et al., 2016):

$$q_{\boldsymbol{\theta}}(\boldsymbol{h}^k|\boldsymbol{x}; \alpha) = \frac{\hat{p}_{\boldsymbol{\theta}}(\boldsymbol{h}^k|\boldsymbol{x})^{\alpha}}{\sum_{\boldsymbol{h'} \in \mathcal{K}(\boldsymbol{x})} \hat{p}_{\boldsymbol{\theta}}(\boldsymbol{h'}|\boldsymbol{x})^{\alpha}} \quad (10)$$

where α controls the distribution sharpness.

Taking the renormalized probability into Equation 7, the loss of multi-granularity metric optimization is formally written as:

$$\mathcal{L}_{MgMO}(\boldsymbol{\theta}) = - \sum_{k=1}^{K} q_{\boldsymbol{\theta}}(\boldsymbol{h}^k|\boldsymbol{x}) R(\boldsymbol{h}^k, \boldsymbol{y}^k) \quad (11)$$

In this way, MgMO decomposes the sequence-level feedback into pieces of multi-granularity metric feedback before integrating them to optimize the model, resulting in a set of more fine-grained training signals that examines different parts of the hypothesis.

Training Following previous work (Ranzato et al., 2016; Shen et al., 2016; Shao et al., 2020; Kong et al., 2019; Du et al., 2021), we adopt a two-stage training strategy, where CMLM loss is first applied for initialization and then replaced by MgMO loss for finetuning. The length loss is maintained throughout the training process.

Model	WMT14		WMT16	
	En-De	De-En	En-Ro	Ro-En
Autoregressive Transformer	27.5	31.2	33.7	34.1
NAT w/ Fertility (Gu et al., 2018)	17.6	19.8	24.5	25.7
CMLM (Ghazvininejad et al., 2019)	18.3	22.0	27.6	28.6
Bag-of-ngrams Loss (Shao et al., 2020)	20.9	24.6	28.3	29.3
Flowseq (Ma et al., 2019)	21.5	26.2	29.3	30.4
Bigram CRF (Sun et al., 2019)	23.4	27.2	-	-
CMLM + AXE (Ghazvininejad et al., 2020)	23.5	27.9	30.8	31.5
DSLP & MT (Huang et al., 2021)	24.2	28.6	31.5	32.6
CMLM + EISL (Liu et al., 2022)	24.2	-	-	-
EM+ODD (Sun and Yang, 2020)	24.5	27.9	-	-
GLAT (Qian et al., 2021)	25.2	29.8	31.2	32.0
Imputer (Saharia et al., 2020)	25.8	28.4	32.3	31.7
CMLM + Order-Agnostic XE (Du et al., 2021)	26.1	30.2	32.4	33.3
AlignNART (Song et al., 2021)	**26.4**	**30.4**	32.5	33.1
latentGLAT (Bao et al., 2022)	**26.6**	29.9	32.5	-
CMLM_1 (our implementation)	20.2	24.5	27.5	29.0
Metric-based Optimization (MO)	24.8	29.1	31.2	32.4
Multi-granularity Metric-based Optimization (MgMO)	**26.4**	**30.3**	**32.9**	**33.6**

Table 1: Performance (test set BLEU) comparison of our proposed method (MgMO) with other fully non-autoregressive models (i.e., one-decoding pass). MgMO is significantly better than both the metric-based model (MO) and the CMLM baseline with $p < 0.01$ (Koehn, 2004). Performance of other NAT models are obtained from corresponding papers. CMLM_1 denotes one-step decoding.

3 Experiments

3.1 Settings

Data We conduct experiments on both directions of two standard machine translation datasets including WMT14 En⇔De and WMT16 En⇔Ro. Knowledge distillation is commonly used for training NAT models (Gu et al., 2018; Sun et al., 2019; Ghazvininejad et al., 2019, 2020). We use the distilled dataset released by Huang et al. (2021) to obtain comparable baseline models.

Initialization We mainly follow Huang et al. (2021) for the model configuration. We use Transformer and adopt Transformer_Base configuration for all experiments: both the encoder and decoder consist of 6 layers with 8 attention heads, and the hidden dimension and feedforward layer dimension is 512 and 2,048, respectively. We train the model using Adam (Kingma and Ba, 2015) optimizer. We set the weight decay as 0.01 and label smoothing as 0.1. The learning rate increases to $5 \cdot 10^{-4}$ in the first 10K steps and then anneals exponentially. For WMT16 En⇔Ro, we use a dropout rate of 0.3 and a batch size of 32K tokens, whereas for WMT14 En⇔De, we switch to 0.1 and 128K accordingly. Code implementation is based on Fairseq (Ott et al., 2019). We train all models for 300,000 steps and select the checkpoint with the best validation performance for MgMO finetuning.

Finetuning We finetune all models for 100,000 steps and use a dropout rate of 0.1. The batch size for WMT16 En⇔Ro and WMT14 En⇔De is 256 and 1,024, respectively. We use a fixed learning rate of $2 \cdot 10^{-6}$ during finetuning. We use GLEU (Wu et al., 2016) score as the metric function and set the value of α in Q-distribution as 0.005. Based on validation results, we use a maximum n-gram size of 6 for GLEU score, set the scaling ratio γ as 8 and set the sample space size K as 40.

Evaluation We use BLEU (Papineni et al., 2002) for all directions. Similar to autoregressive settings, we use $l = 5$ length candidates during inference (Ghazvininejad et al., 2020; Du et al., 2021). We select the best checkpoint for evaluation based on validation BLEU scores.

3.2 Main Results

We compare our method with the autoregressive Transformer and other fully NAT baselines (i.e., one decoding pass). The results are shown in Table 1. We can observe that applying metric-based optimization (MO) on the CMLM baseline brings an improvement of 4.1 BLEU scores on average. By decomposing the metric feedback into multi-granularity levels, MgMO further obtains a significant improvement (with $p < 0.01$ (Koehn, 2004)), expanding the advantage to 5.5 BLEU scores on average. Compared with other representative NAT baselines, MgMO achieves the best performance

Strategy	Decoder Input Decoder Target
N&C	<unk> <unk> <unk> <unk> <unk> I never went back .
P&C	<unk> <unk> went back <unk> I never went back .
P&P	<unk> <unk> went back <unk> I never <m> <m> .
N&P	<unk> <unk> <unk> <unk> <unk> I never <m> <m> .

Table 2: Sample decoder input-target pairs for the source German sentence "Ich kehrte nie zurück .", under different training strategies. For decoder input, "<unk>" denotes the unknown token in the dictionary and indicates null information. For decoder traget, segments that are not replaced by "<m>" are highlighted for computing metric feedback.

Dataset	Training Strategy			
	N&C	P&C	P&P	N&P
WMT16 En⇒Ro	32.0	32.7	32.7	**32.9**
WMT16 Ro⇒En	33.8	34.3	34.4	**34.5**

Table 3: Comparison of different training strategies on WMT16 En⇔Ro validation set.

on WMT16 En⇔Ro and highly competitive performance on WMT14 En⇔De. Since no modification is involved in model architecture and inference, MgMO achieves the same inference latency with the vanilla NAT model.

3.3 Training Strategies

In addition to the default setting, we consider three alternative training strategies. Generally, these strategies differ in how much information is fed into the NAT decoder and how much information left requires the decoder to fit: (1) none of the target tokens are observed and the complete set of target tokens are considered for computing metric feedback (**N&C**); (2) part of the target tokens are observed and the complete set of target tokens are considered for computing metric feedback (**P&C**); (3) part of the target tokens are observed and a partial set of the target tokens are highlighted for computing metric feedback (**P&P**); (4) none of the target tokens are observed and a partial set of the target tokens are highlighted for computing metric feedback (**N&P**, i.e., the default setting). We present an example in Table 2 as an intuitive illustration for different training strategies.

We conduct experiments on the WMT16 En⇔Ro dataset. The results are shown in Table 3. We can observe that both partial observations (P&C and P&P) and partial predictions (P&P and N&P)

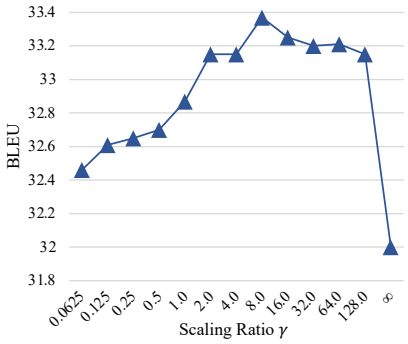

Figure 3: Effect of γ on the WMT En⇒Ro validation set. A larger γ indicates tokens in hypotheses and corresponding references are less likely to be masked.

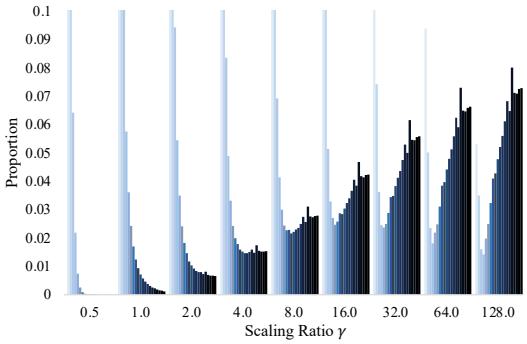

Figure 4: Proportions of granularities (1-gram to 20-gram) under different scaling ratios γ. Darker colors indicate larger granularities.

obtain consistent improvement over the N&C strategy, i.e., metric-based optimization without multi-granularity (MO in Table 1). P&C achieves similar performance to P&P, since it is easy for the model to copy the partial observations as the corresponding predictions, and thus focus on the unobserved tokens. In other words, P&C, P&P and N&P all incorporate training signals from multiple granularities, either explicitly or implicitly, and therefore obtain better performance. Due to a smaller discrepancy between training and inference, N&P obtains further performance improvement over P&C and P&P.

3.4 Ratio of Exposed Segments

The scaling ratio γ controls how likely one token is masked, i.e., a larger γ indicates a higher probability of being exposed for each token and a lower probability otherwise. In general, a proper scaling ratio yields more diverse granularities from which the model learns rich token dependencies.

As can be seen from the Figure 3, increasing the scaling ratio γ until 8.0 steadily brings performance

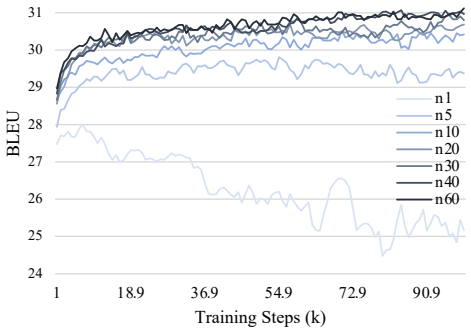

Figure 5: Effect of sample space size K on the WMT En⇒Ro validation set. Darker colors indicate more samples are explored for each training instance.

Maximum N-gram Size			
1-gram	**2-gram**	**4-gram**	**6-gram**
32.7	32.7	**32.9**	**32.9**

Table 4: Effect of maximum n-gram size of the GLEU metric on the WMT16 En⇒Ro validation set.

Metric Function for Optimization				
BLEU	**GLEU**	**chrF**	**TER**	**Rouge-2**
32.7	**32.9**	**32.9**	32.6	32.7

Table 5: Effect of different metric functions on the WMT16 En⇒Ro validation set.

improvement in terms of validation BLEU scores. This can be because a larger γ results in longer and more informative segments across different samples. Therefore, the model is encouraged to learn to handle longer and more difficult token dependency, which is common in long sequences and is a major challenge for NAT (Gu et al., 2018; Qian et al., 2021; Du et al., 2021). Analysis of performance on different sequence lengths (Section 4.1) further validate our assumption.

As the ratio increases further, model performance begins to deteriorate since overly large ratios nearly expose all tokens for each sample, resulting in coarse granularities with limited diversity. The extreme case becomes the N&C strategy ($\gamma \to \infty$), which reveals all target tokens in every sample. In this case, the complete sequence becomes the only granularity, giving a coarse and monotonous feedback that is hard to learn from.

To provide a statistical intuition of why the ratio of 8 obtains better performance, we traverse the training set (WMT'16 En⇒Ro) under different scaling ratios, and calculate the ratios of different granularities, i.e., the segments of different lengths. The results are shown in Figure 4. We can observe that masking with a larger ratio spreads more proportions on larger granularities. While maintaining a significant portion on large granularities, the ratio of 8.0 yields a relatively smooth distribution over various granularities, contributing to a progressive learning curriculum.

3.5 Size of Search Space

Intuitively, a larger sample space, i.e., a larger K, brings better granularity diversity since more sets of segments are encountered. On the other hand, enlarging the sample space increases computational complexity. We explore the effect of sample space size K, with results shown in Figure 5. We can observe that increasing K up to 40 brings a steady performance improvement on the validation set. A larger K (i.e., 60) does not lead to further improvement, indicating that the model has encountered sufficient types of granularities.

3.6 Alternative Optimization Targets

Beyond the standard GLEU metric, we also explore the effects of the maximum n-gram size and other alternative metrics for optimization.

Effects of N-gram Size We vary the maximum n-gram size in GLEU score and the results are shown in Table 4. We can observe that rewarding local matches (1-gram and 2-gram) obtains comparable performance to that of larger span matches (4-gram and 6-gram). We hypothesize that multi-granularity optimization compensates for capturing word ordering in some degree by simultaneously evaluating various granularities of the target sequence, which implicitly restricts the token locations.

Alternative Metric Functions We also explore the model performance under different metric functions, with results presented in Table 5. We can see that different metric functions achieve comparable performance as they all measure sequence-level similarities. Specifically, the metrics (GLEU and chrF (Popović, 2015)) considering both n-gram precision and recall obtain better performance compared with BLEU which only considers precision. Since Rouge-2 (Lin, 2004) considers a maximum n-gram size of 2, it achieves worse performance than GLEU. TER (Snover et al., 2006) measures the number of edit operations and obtains a slightly worse performance, since it does not reward n-gram match and suffers a discrepancy from the evaluation metric, i.e., BLEU.

Reference	Dist@@ ant plan@@ ets , gra@@ vit@@ ational waves or black holes - experts from the European Space Agency must now agree on two major projects that are to be launched in the coming years .
CMLM	Far af@@ ets , waves waves gra@@ vit@@ vit@@ black black black holes - experts the European 1.0 0.75 0.12 0.41 0.96 0.30 0.14 0.60 0.28 0.25 0.39 0.43 0.57 0.23 0.21 Space Agency Agency must agree on on two projects projects that they to launch in in next few years . 0.18 0.23 0.36 0.23 0.16 0.21 0.29 0.30 0.22 0.19 0.48 0.35 0.40 0.26 0.42 0.26 0.39 0.45 0.41
MgMO	Far plan@@ ets , gra@@ vit@@ ational waves or black holes - experts from the European 1.0 0.91 1.0 1.0 0.99 1.0 0.99 1.0 1.0 1.0 1.0 1.0 1.0 1.0 Space Agency must now agree on two major projects that they intend to launch in the coming years . 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.98 1.0 1.0 1.0 1.0 1.0

Figure 6: An example from the WMT'14 De⇒En test set, with confidence (probability of generating the token) annotated below each token. Blue color denotes token repetitions.

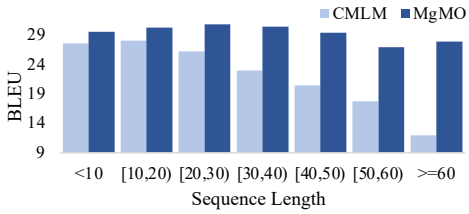

Figure 7: Comparison of BLEU scores with respect to the lengths of the reference sentences on the WMT'14 De⇒En test set.

Model	WMT'14	
	En⇒De	De⇒En
CMLM + OaXE	0.19	0.19
CMLM	0.76	0.67
MgMO	**0.10**	**0.10**

Table 6: Normalized Corpus-level multimodality (NCM) scores of the sentences on the WMT'14 En⇔De test set. The results of CMLM + OaXE are obtained from Du et al. (2021).

4 Analysis

In this section, we conduct quantitative and qualitative analysis to dig some insights into how MgMO benefits non-autoregressive translation.

4.1 Sequence Lengths

We analyze the effectiveness of our method by comparing performance on test sentences of different lengths. We use compare-mt (Neubig et al., 2019) to split the WMT'14 De⇒En test sets into several subsets based on target sequence lengths. The results are shown in Figure 7. As the sequence length grows, the baseline model trained using XE loss suffers great performance deterioration. Under MgMO, the model maintains relatively stable performance on test sentences across different lengths, proving that multi-granularity learning brings benefits for capturing non-local dependencies that can spread across long text spans.

4.2 Prediction Confidence

NAT shows weakness in handling multi-modality (Gu et al., 2018; Sun et al., 2019), which is reflected by its low confidence on locating token translations among neighboring positions (Ghazvininejad et al., 2020; Du et al., 2021). Ideally, we expect each token to have a high probability mass at the position it is predicted, but low at the neighboring positions. Following previous work (Sun and Yang, 2020; Du et al., 2021), we compute Normalized Corpus-level multimodality (NCM) on the WMT14 En⇔De test set which measures average token-level prediction confidence. The results are shown in Table 6, and lower NCM scores indicate higher confidence. We can see that applying MgMO largely increases the model prediction confidence at each step. This can be because MgMO better captures token interdependency via optimizing model predictions based on various contexts, i.e., different sets of exposed segments. We show an example in Figure 6 to provide an intuition of effects brought by higher prediction confidence.

5 Related Work

Fully Non-Autoregressive Models To bridge the performance gap between fully NAT and the autoregressive counterpart, lots of techniques have been proposed to build dependencies among the target tokens such as curriculum learning (Guo et al., 2020; Liu et al., 2020; Ding et al., 2021a), latent variable modeling (Libovický and Helcl, 2018; Kaiser et al., 2018; Ma et al., 2019; Saharia et al., 2020; Bao et al., 2022, 2021), improving distillation training (Zhou et al., 2020; Ding et al., 2021c,b), and adaptive token sampling (Qian et al., 2021). Despite their success, these methods are trained with XE loss, which forces a strict mapping. Another line of work shift to improvement of XE loss or metric-based objectives. For example, Ghazvinine-

jad et al. (2020) soften the penalty for word order errors based on a monotonic alignment assumption, and Du et al. (2021) computes XE loss based on the best possible alignment between predictions and target tokens. In contrast, we unitize hypothesis sampling and metric-based optimization, allowing the model to explore hypothesis of different lengths. Sun et al. (2019) incorporate an approximation of Conditional Random Fields (CRF) to model output dependency, while the decoding is not parallelized. Shao et al. (2019) devise customized reinforcement algorithms to optimize global metrics for NAT. Shao et al. (2020) and Liu et al. (2022) propose differentiable n-gram matching losses between the hypothesis and reference. In comparison, we propose to integrate feedback from evaluating model behavior on multi-level granularities within a single forward-backward propagation.

Metric-based Optimization for NMT Metric-based optimization has been utilized in NMT (Ranzato et al., 2016; Shen et al., 2016; Bahdanau et al., 2017; Wu et al., 2018; Edunov et al., 2018b; Kong et al., 2019; Kiegeland and Kreutzer, 2021) to alleviate the mismatch between the optimization during training and evaluation during inference. For example, Ranzato et al. (2016) train NMT using the objective gradually shifting from token-level likelihood to sentence-level BLEU score, and Shen et al. (2016) adopt minimum risk training (MRT) to minimize the task specific expected loss (i.e., induced by BLEU score). Kong et al. (2019) use translation adequacy as the metric function. Different from them, we propose to integrate metric feedback on various granularities instead of a coarse sequence-level reward.

6 Conclusion

We proposed multi-granularity optimization for NAT, which considers metric feedback on hypothesis segments of multiple granularities. Through integrating multi-granularity feedback, the model is optimized by focusing on different parts of the sequence within a single forward-backward pass, obtaining more detailed and informative training signals. Empirical results demonstrated that our method achieved highly competitive performance compared with other representative baselines for fully NAT. Analysis further showed that MgMO maintained strong performance on long sequences that vanilla NAT models suffer, and obtained high prediction confidence. Beyond non-autoregressive translation, our proposed method can be used in other text generation tasks.

Limitations

Despite competitive performance, the model still suffers common issues in NMT. Firstly, although the total GPU memory cost for our method is lower than that of XE loss (as the batch size under MgMO is much lower), MgMO requires a relatively large minimum memory capacity since lots of samples are considered for forward and backward propagation for each training instance. Secondly, like many other NAT models, our model also suffers, though smaller, performance deterioration without using data distilled from autoregressive teacher models.

Acknowledgment

We thank all reviewers for their insightful comments. This publication has emanated from research conducted with the financial support of the Pioneer and "Leading Goose" R&D Program of Zhejiang under Grant Number 2022SDXHDX0003. This work is also under a grant from Lan-bridge Information Technology Co., Ltd.

References

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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.

Yu Bao, Hao Zhou, Shujian Huang, Dongqi Wang, Lihua Qian, Xinyu Dai, Jiajun Chen, and Lei Li. 2022. GLAT: glancing at latent variables for parallel text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8398–8409. Association for Computational Linguistics.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021a. Progressive multi-granularity training for non-autoregressive

translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2797–2803, Online. Association for Computational Linguistics.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021b. Rejuvenating low-frequency words: Making the most of parallel data in non-autoregressive 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 3431–3441, Online. Association for Computational Linguistics.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021c. Understanding and improving lexical choice in non-autoregressive translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Cunxiao Du, Zhaopeng Tu, and Jing Jiang. 2021. Order-agnostic cross entropy for non-autoregressive machine translation. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 2849–2859. PMLR.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018a. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018b. Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 355–364, New Orleans, Louisiana. Association for Computational Linguistics.

Marjan Ghazvininejad, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy. 2020. 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.

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, Linli Xu, Tao Qin, Enhong Chen, and Tie-Yan Liu. 2020. Fine-tuning by curriculum learning for non-autoregressive neural machine translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7839–7846. AAAI Press.

Chenyang Huang, Hao Zhou, Osmar R. Zaïane, Lili Mou, and Lei Li. 2021. Non-autoregressive translation with layer-wise prediction and deep supervision. *ArXiv preprint*, abs/2110.07515.

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.

Samuel Kiegeland and Julia Kreutzer. 2021. Revisiting the weaknesses of reinforcement learning for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1673–1681, Online. 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*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Xiang Kong, Zhaopeng Tu, Shuming Shi, Eduard H. Hovy, and Tong Zhang. 2019. Neural machine translation with adequacy-oriented learning. 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 6618–6625. AAAI Press.

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.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Guangyi Liu, Zichao Yang, Tianhua Tao, Xiaodan Liang, Junwei Bao, Zhen Li, Xiaodong He, Shuguang Cui, and Zhiting Hu. 2022. Don't take it literally: An edit-invariant sequence loss for text generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2055–2078, Seattle, United States. Association for Computational Linguistics.

Jinglin Liu, Yi Ren, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2020. Task-level curriculum learning for non-autoregressive neural machine translation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3861–3867. ijcai.org.

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.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. compare-mt: A tool for holistic comparison of language generation systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 35–41, Minneapolis, Minnesota. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. 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.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. 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.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

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.

Chenze Shao, Yang Feng, Jinchao Zhang, Fandong Meng, Xilin Chen, and Jie Zhou. 2019. Retrieving sequential information for non-autoregressive neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3013–3024, Florence, Italy. 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 *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 198–205. AAAI Press.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In

Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Jongyoon Song, Sungwon Kim, and Sungroh Yoon. 2021. AligNART: Non-autoregressive neural machine translation by jointly learning to estimate alignment and translate. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1–14, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

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.

Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. A study of reinforcement learning for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3612–3621, Brussels, Belgium. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv preprint*, abs/1609.08144.

G. Yule. 2006. *The Study of Language*. Cambridge University Press.

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.