Order-Agnostic Cross Entropy for Non-Autoregressive Machine Translation

Cunxiao Du [1] **Zhaopeng Tu** [2] **Jing Jiang** [1]

Abstract

We propose a new training objective named *order-agnostic cross entropy* (OAXE) for fully non-autoregressive translation (NAT) models. OAXE improves the standard cross-entropy loss to ameliorate the effect of word reordering, which is a common source of the critical multimodality problem in NAT. Concretely, OAXE removes the penalty for word order errors, and computes the cross entropy loss based on the best possible alignment between model predictions and target tokens. Since the log loss is very sensitive to invalid references, we leverage cross entropy initialization and loss truncation to ensure the model focuses on a good part of the search space. Extensive experiments on major WMT benchmarks show that OAXE substantially improves translation performance, setting new state of the art for fully NAT models. Further analyses show that OAXE alleviates the multimodality problem by reducing token repetitions and increasing prediction confidence. Our code, data, and trained models are available at https://github.com/tencent-ailab/ICML21_OAXE.

1. Introduction

Non-autoregressive translation (NAT, Gu et al., 2018; Libovický & Helcl, 2018) has received increasing attention for its efficient decoding by predicting every target token in parallel. However, such advantage comes at the cost sacrificing translation quality due to the *multimodality* problem. There exist many possible translations of the same sentence, while vanilla NAT models may consider them at the same time due to the independent predictions, which leads to multi-modal outputs in the form of token repetitions (Gu et al., 2018).

A number of recent efforts have explored ways to improve the NAT models' ability to handle multimodality. One thread of research relaxes the fully non-autoregressive restriction and iteratively refine the generated outputs with K decoding passes (Lee et al., 2018; Gu et al., 2019). However, the iterative refinement method improves translation quality by sacrificing the primary benefit of NAT models – fast inference (Kasai et al., 2021).

Another thread of work focuses on improving *fully* NAT models (i.e., one decoding pass) to maintain their advantage of decoding efficiency. Along this direction, recent works have incorporated approaches to improving the standard cross-entropy loss to ameliorate the effect of multimodality. Ghazvininejad et al. (2020) state that modeling word order is difficult for NAT, since the model cannot condition on its previous predictions like its autoregressive counterpart. Starting from this intuition, they propose a new training loss that **softens** the penalty for word order errors, which assigns loss based on the best possible monotonic alignment between target tokens and model predictions. Similarly, Libovický & Helcl (2018); Saharia et al. (2020) marginalize all the monotonic alignments between target and predictions to compute loss. While both approaches improve NAT performance by relaxing the word order restriction based on the **monotonic** alignment assumption, they cannot handle **word reordering** – a common source of multimodality problem (see Figure 1). In response to this problem, we propose to **remove** the penalty for word order errors, which makes the loss more compatible with parallel decoding in NAT.

To achieve this more relaxed loss, we introduce order-agnostic cross entropy (OAXE), a new training objective function that computes the cross entropy loss based on the best possible alignment between model predictions and target tokens. Compared with previous studies (Ghazvininejad et al., 2020; Saharia et al., 2020), OAXE guides NAT models to focus on lexical matching, which is at the core of machine translation tasks. We use Hungarian algorithm to efficiently implement OAXE (e.g., 7 lines of core code, see Appendix A.1), which makes our new loss accommodate existing NAT models seamlessly. The log loss is sensitive to invalid and noisy references (Kang & Hashimoto, 2020), and there exists a large number of invalid alignments in the search space, which will hurt the performance of OAXE in the form of log loss. We tackle these challenges by removing

[1] School of Computing and Information System, Singapore Management University, Singapore. Work was done when Cunxiao Du was under the Rhino-Bird Elite Training Program of Tencent AI Lab. [2] Tencent AI Lab, China. Correspondence to: Zhaopeng Tu <zptu@tencent.com>.

Proceedings of the 38^{th} *International Conference on Machine Learning*, PMLR 139, 2021. Copyright 2021 by the author(s).

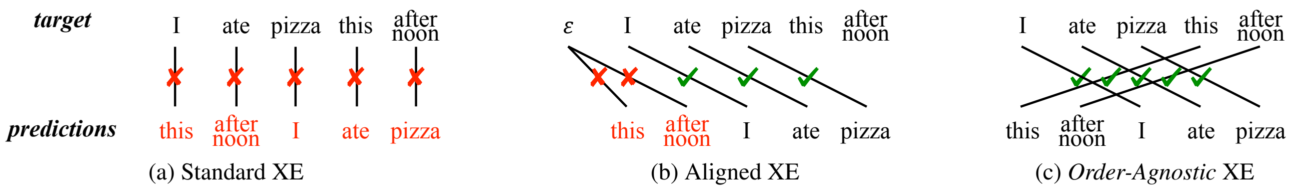


Figure 1. Illustration of the loss functions: (a) the standard XE (Gu et al., 2018; Ghazvininejad et al., 2019) where a penalty is incurred for every out-of-position prediction, (b) a relaxed AXE (Ghazvininejad et al., 2020) that assigns loss based on *the best possible monotonic alignment* between model predictions and target tokens, and (c) our proposed Order-Agnostic XE that assigns loss based on **the best possible alignment** between predictions and target. We highlight in red color the predictions that are penalized by the loss function.

invalid alignments via cross entropy initialization to ensure the model focuses on a good part of the search space. We also use loss truncation (Kang & Hashimoto, 2020) to drop invalid model predictions, which biases the model towards learning only the confident (and likely accurate) predictions.

We first design a synthetic experiment to demonstrate that OAXE better captures the multimodality of word orders. Extensive experiments on several machine translation benchmarks with different settings show that OAXE substantially improves translation performance over NAT models trained with both cross entropy (Ghazvininejad et al., 2019) and aligned cross entropy (AXE, Ghazvininejad et al., 2020). OAXE increases training time by 1.36× (on par with cross entropy), and does not affect inference speed. Our comprehensive analyses show that OAXE indeed alleviates the critical multimodality problem by reducing token repetitions and increasing prediction confidence. OAXE can also improve output fluency by benefiting from its ability to find an appropriate ordering of generated tokens.

Contributions Our main contributions are as follows:

- We demonstrate the necessity of explicitly handling word reordering – a common source of the critical multimodality problem in NAT.
- We propose a new training objective for NAT that removes the penalty for word order errors, which consistently outperforms the standard cross entropy loss on several benchmarking datasets with and without knowledge distillation.
- We achieve new state-of-the-art performances for fully NAT on six machine translation benchmarks: WMT14 En⇔De (26.1 and 30.2), WMT16 En⇔Ro (32.4 and 33.3), and WMT17 En⇔Zh (32.9 and 22.1), which closely match the performance of AT model.

2. Methodology

In this section, we first briefly introduce the NAT model and its limitation on handling word order (Section 2.1). We then describe the proposed Order-Agnostic Cross Entropy (OAXE) loss, which alleviates the limitation by assigning loss based on the best possible alignment between target tokens and model predictions (Section 2.2). To ease the training of OAXE, we propose two training strategies to remove invalid alignments in the search space (Section 2.3).

2.1. Preliminaries: Non-Autoregressive Translation

Cross Entropy (XE) Standard NAT models (Gu et al., 2018) are trained with the vanilla cross entropy loss:

$$\mathcal{L}_{XE} = -\log P(Y|X) = -\sum_{y_n} \log P(y_n|X), \qquad (1)$$

where (X, Y) with $Y = \{y_1, \dots, y_N\}$ is a bilingual training example, and $P(y_n|X)$ is calculated independently by the NAT model with parameters θ. As shown in Figure 1a, the standard XE requires a strict match of word order between target tokens and model predictions, thus will heavily penalize all predictions for "this afternoon I ate pizza", although it is semantically equivalent to the target.

Aligned Cross Entropy (AXE) Several recent efforts improve the standard cross-entropy loss to ameliorate the effect of multimodality of word order (Libovický & Helcl, 2018; Ghazvininejad et al., 2020; Saharia et al., 2020). Closely related to this work, Ghazvininejad et al. (2020) softens the penalty for word order errors based on a **monotonic alignment** assumption. They define an alignment α to be a function that maps target positions to prediction positions, and the conditional AXE is calculated as:

$$\mathcal{L}_{AXE} = -\sum_{y_n} \log P_\alpha(y_n|X) - \sum_{k \notin \alpha} P_k(\epsilon), \qquad (2)$$

where the first term is an aligned cross entropy between target and predictions, and the second term is a penalty for unaligned predictions. As shown in Figure 1b, the monotonic assumption of AXE can recover part of the mistakenly penalized predictions (e.g., "I ate pizza"), but fails to handle reordered predictions (e.g., "this afternoon").

2.2. Order-Agnostic Cross Entropy OAXE

Intuition Word reordering – one of the structural sentence characteristics, is a common source of multimodality. Two semantically equivalent sentences can vary greatly in how they arrange sentence components. For example, sentences can be in the active or passive form; a temporal adverbial can be placed either at the beginning of a sentence (e.g., "*this afternoon* I ate pizza") or at the end (e.g., "I ate pizza *this afternoon*"). Sun & Yang (2020) have shown that NAT is hard to fit all the semantically equivalent translations via cross entropy; however, NAT is powerful enough to fit one specific translation. The observation opens up the possibility that an appropriate loss can guide the model to find a possible translation that is semantically equivalent to the target while in a different word order. Figure 1c illustrates the intuition of the proposed OAXE to handle word reordering for NAT models. OAXE removes the word order restriction of XE, and assigns loss based on the best alignment between target tokens and model predictions.

Formulation and Implementation Given one training example (X, Y), each alignment between model predictions and target tokens is an ordering of the set of target tokens $(y_1, \dots, y_N)$ (e.g., $Y^i = \{y_N, y_1, \dots, y_{N-1}\}$). Without loss of generality, from here on, *we refer to each alignment as an ordering as well*. Specifically, we define the ordering space $\mathbf{O} = \{O^1, \dots, O^I\}$ for Y, and the OAXE objective is defined as finding the best ordering O^i to minimize the cross entropy loss:

$$\mathcal{L}_{\text{OAXE}} = \underset{O^i \in \mathbf{O}}{\operatorname{argmin}} \left(-\log P(O^i|X) \right), \tag{3}$$

where $-\log P(O^i|X)$ is the cross entropy loss for ordering O^i, which is calculated by Equation 1. Briefly, after we got the best ordering, we used it to reorder the original reference, which will be used to calculate the loss $-\log P(O^i|X)$ using Equation 1. For example, the original reference "I ate pizza this afternoon" will be reordered to "this afternoon I ate pizza", which is used to calculate the XE loss.

For the target Y with N tokens, there exists $N!$ orderings in the search space $\mathbf{O}$, which makes it computationally infeasible to calculate Equation 3 explicitly. In response to this problem, we cast this problem as Maximum Bipartite Matching and leverage the efficient Hungarian algorithm (Kuhn, 1955), which reduces the time complexity from $\mathcal{O}(N!)$ to $\mathcal{O}(N^3)$. Specifically, as shown in Figure 2, we construct the bipartite graph $G = (U, V, E)$ where the first part of vertices U is the set of N positions and the second part of vertices V is the set of N target tokens. Each edge in E is the prediction log probability for the token y in position n. With the efficient implementation,[1] the training time of the OAXE loss is similar to (about 1.36 times slower than) training with cross entropy loss.[2] Note that OAXE requires no changes to parallel decoding in inference.

Vocabulary	Pos:1	Pos:2	Pos:3	Pos:4	Pos:5
I	0.2	0.1	**0.4**	0.1	0.1
ate	0.1	0.1	0.1	**0.4**	0.1
pizza	0.1	0.1	0.3	0.1	**0.5**
this	**0.5**	0.1	0.1	0.3	0.1
afternoon	0.1	**0.6**	0.1	0.1	0.3

Output Probability Distribution

Figure 2. Illustration of bipartite matching to implement the proposed order-agnostic XE loss. We only show the probabilities of the target words for better illustration, and highlight in bold the prediction selected by OAXE for each position. The best ordering is "this afternoon I ate pizza".

2.3. Training

A large portion of the $N!$ possible orderings in $\mathbf{O}$ are invalid (e.g., "pizza I this ate afternoon"), and log loss requires that models assign probabilities to all potential reference sequences (i.e., orderings in this study). Kang & Hashimoto (2020) have demonstrated that log loss is not robust to noisy and invalid references, since log loss is sensitive to outliers: invalid or noisy references with small probability mass can cause large changes in model behavior. In response to this problem, we design two strategies to remove invalid references at the levels of both sentence (i.e., invalid orderings) and token (i.e., invalid predictions).

Avoiding Invalid Orderings via XE Initialization Huang et al. (2009) have shown that the marginal distribution can be used as first order approximation to the joint probability of distributions on permutations. As pointed by Kasai et al. (2020) and Kreutzer et al. (2020), the NAT model trained with vanilla cross entropy could be view as estimating the marginal distribution $P(y_n|X)$ for the n^{th} target token. Starting from this intuition, we initialize with XE loss to make sure the model can effectively deal with the large search space of orderings. Such strategy has been successfully applied in sequence-level training for autoregressive models (Ranzato et al., 2016; Shen et al., 2016; Kong et al., 2019) to deal with the large search space of the sequence-level loss. Specifically, we design two methods to initialize the OAXE-based NAT with the XE loss.

[1] We implemented Hungarian Match with a CPU-version python package *scipy*, which iteratively handles each instance in a mini-batch. This can be accelerated by a GPU-friendly version (e.g. Lopes et al., 2019), which we leave for the future work.

[2] The training process consists of both the forward pass (e.g. layer and softmax operations) and backward pass (e.g. **loss calculation** and parameter update). Accordingly, although the additional computation of the loss OAXE is $\mathcal{O}(N^3)$, the training speed is only 1.36 times slower.

For the first idea, we start with an initialization pre-trained with the XE loss and slowly deviate from it, instead of starting from a poor random initialization. This ensures that we start off with a much better initialization than random, since now the model focuses on a good part of the search space, which can avoid a large portion of invalid orderings.

The second idea is to introduce OAXE loss during training with an annealing schedule in order to gradually teach the model to remove the order constraint for loss calculation. Specifically, we follow Ma et al. (2018) to combine XE and OAXE losses:

$$\mathcal{L}_{\text{JOINT}} = T_m * \mathcal{L}_{\text{XE}} + (1 - T_m) * \mathcal{L}_{\text{OAXE}},$$

where T_m is the temperature at the m^{th} epoch of training, which progressively decreases to 0:

$$T_m = \max(0, 1 - c^{m - \lambda M}),$$

where M is the total epochs of training, and (c, λ) are pre-defined constant numbers. Accordingly, in the last $(1-\lambda)M$ epochs, only OAXE is used to train the models.

Dropping Invalid Predictions via Loss Truncation Even with XE initialization, the model may still find orderings with invalid predictions (e.g., "I *pizza ate* this afternoon"), which are back propagated to update the model parameters. To tackle this problem, we drop the invalid predictions in the searched ordering $\widehat{O} = \{\hat{y}_1, \ldots, \hat{y}_N\}$ with loss truncation (Kang & Hashimoto, 2020):

$$\mathcal{L}_{\text{OAXE-T}} = \sum_{\hat{y}_n : P(\hat{y}_n) > \pi} \left(-\log P(y_n | X) \right),$$

where π is the truncation margin, which is tuned on the validation set. We only back-propagate the tokens whose probabilities are higher than the margin, which biases the model towards learning only the confident (and likely factually accurate) predictions. We have empirically found that truncating after XE hotstarting primarily drops invalid predictions, which is consistent with the findings reported by Kang & Hashimoto (2020).

3. Synthetic Ordering Experiment

We first evaluate the effectiveness of the proposed approach to alleviate the multimodality problem in terms of word reordering, which is closely related to the proposed order-agnostic XE loss. For this, we design a synthetic dataset to simulate the word reordering of machine translation tasks, and simplify the other characteristics.

Data To isolate the effect of word reordering from machine translation, we ignore the word transformation from the source language to the target language. The *source sentence* is a sequence of random numbers, and the target is an ordering of the numbers (Table 1). To mimic the real dataset, we set the vocabulary size of numbers as 32K, and samples the sequence length from a uniform distribution from 10 to 100. The training set consists of 300K instances, in which the target is an ordering sampled from a given set of ordering modes from a categorical distribution. Both the validation and test sets consist of 3K instances and all the ordering modes serves as the references for the test sets. Suppose the training data consists of two ordering modes (e.g. direct and reverse). In testing, we will consider the output as Exact Match if the model can generate either direct or reverse orders. This setting tries to simulate the MT scenario: although only one mode is observed in the training data, the other semantically equivalent modes should also be regarded as correct translation.

Mode	Source	Target
Direct	1 3 10 7 9 2	1 3 10 7 9 2
Reverse	1 3 10 7 9 2	*2 9 7 10 3 1*
Flip	1 3 10 7 9 2	7 9 2 1 3 10
Flip-Right-Rev	1 3 10 7 9 2	7 9 2 *10 3 1*
Filp-Left-Rev	1 3 10 7 9 2	*2 9 7* 1 3 10

Table 1. Examples of the synthetic ordering modes.

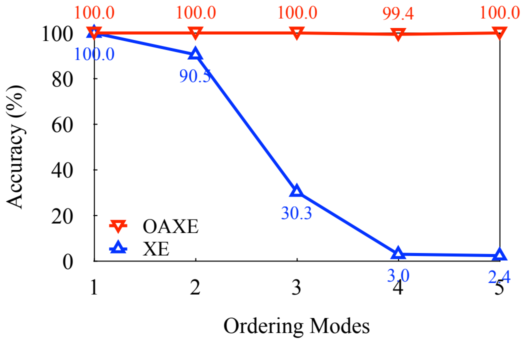


Figure 3. Results on the synthetic ordering dataset. "Ordering Modes" is the number of distinct ordering modes in the dataset.

Model The NAT models used in this experiment generally share the same hyper-parameters with that in the machine translation experiment (Section 4.1), except that we trained the models for 30K steps, measured the validation loss at the end of each epoch, and chose the best one.

Evaluation Figure 3 shows the exact match accuracy for different number of ordering modes. "Ordering Modes = 3" denotes that we sample the target ordering from the set of ("Direct", "Reverse", "Flip"). Clearly, the performance of XE drops rapidly when the number of ordering modes increases, which reconfirms that the XE loss is weak at handling multimodality problem. Our OAXE performs well for multiple ordering modes, which we attribute to its superiority on finding the best ordering in the search space.

4. Machine Translation Experiment

4.1. Experimental Setup

Data We conducted experiments on major benchmarking datasets that are widely-used in previous NAT studies (Gu et al., 2018; Shao et al., 2020; Ma et al., 2019; Saharia et al., 2020): WMT14 English⇔German (En⇔De, 4.5M sentence pairs), WMT16 English⇔Romanian (En⇔Ro, 0.6M sentence pairs). We preprocessed the datasets with a joint BPE (Sennrich et al., 2016) with 32K merge operations for En⇔De, and 40K merge operations for En⇔Ro. We reported the BLEU (Papineni et al., 2002) on both tasks.

Since our work is closely related to Ghazvininejad et al. (2019; 2020), we also validated our approach on the large-scale WMT17 English⇔Chinese (En⇔Zh, 20M sentence pairs) dataset. We learned a BPE model with 32K merge operations for the dataset. For fair comparison, we reported the Sacre BLEU (Post, 2018) on the En-Zh task, and the BLEU on the Zh-En task.

Knowledge Distillation We closely followed previous works on NAT to apply sequence-level knowledge distillation (Kim & Rush, 2016) to reduce the modes of the training data. We trained the NAT models on the translations generated by a left-to-right AT model. Consistent with CMLM (Ghazvininejad et al., 2019) and AXE (Ghazvininejad et al., 2020), we employed Transformer-BIG (Vaswani et al., 2017) for distilling the En⇔De and En⇔Zh datasets, and Transformer-BASE for distilling the En⇔Ro dataset.

NAT Models We validated OAXE on the SOTA NAT model – CMLM (Ghazvininejad et al., 2019), which uses the conditional mask LM (Devlin et al., 2019) to generate the target sequence from the masked input. The NAT model shares the same architecture as Transformer-BASE: 6 layers for both the encoder and decoder, 8 attention heads, 512 model dimensions, and 2048 hidden dimensions. We chose the CMLM models with the vanilla XE loss (Ghazvininejad et al., 2019) and the AXE loss (Ghazvininejad et al., 2020)[3] as our two main baselines. To keep consistent with main baselines, we set 5 as length candidates for all CMLM models during inference. We generally followed the standard hyperparameters used in (Ghazvininejad et al., 2019).

Loss	XE Annealing		XE Pre-Training	
	En-De	De-En	En-De	De-En
XE	18.6	23.0	18.9	23.9
AXE	23.1	28.0	23.2	28.1
OAXE	23.8	28.9	**24.1**	**29.4**

Table 2. Impact of different XE initialization strategies on the WMT14 En⇔De validation set. We list the results of fine-tuning with XE (i.e., 18.9 and 23.9) to make fair comparison. We re-implemented AXE in this experiment.

We trained batches of approximately 128K tokens using Adam (Kingma & Ba, 2015). The learning rate warmed up to 5×10^{-4} in the first 10K steps, and then decayed with the inverse square-root schedule. We trained all models for 300k steps, measured the validation BLEU at the end of each epoch, and averaged the 5 best checkpoints.

4.2. Ablation Study

In this section, we investigated the impact of different components for OAXE on the WMT14 En⇔De validation sets.

Different XE Initialization Strategies For the XE annealing experiments, we set c as 16, and λ as 0.95. For the XE pre-training experiments, we fine-tuned all models for 10 epochs (e.g., 11K steps for En⇔De) for all language pairs, and chose the best single checkpoint with highest validation BLEU. The learning rate of fine-tuning warmed up to 5×10^{-6} in the first 5K steps, and then decayed with the inverse square-root schedule.

Table 2 shows the effect of different XE initialization strategies for OAXE training. We compare with two XE variants which share the same model architecture, and the only difference is the training objective. Clearly, the proposed OAXE loss consistently outperforms the other XE losses in all cases, demonstrating its effectiveness for machine translation. The XE pre-training strategy achieves the best performance, which can obtain an averaged improvement of 6.0 BLEU over the XE baseline. In the following experiments, we use XE pre-training as the default training strategy for its efficiency and effectiveness.

However, OAXE fails to converge without XE initialization (i.e. 0 BLEU). We also conducted experiments about how XE initialization with different pre-training steps affect OAXE in Appendix A.3. Encouragingly, a pre-trained XE initialization at very early stage (9.0 BLEU at 10K steps) can train a reasonable OAXE model (18.5 BLEU).

Impact of Truncation Margin Figure 4 shows the impact of truncation margin π, which is searched from

[3] Since the code of AXE was not released, we re-implemented it on top of the released CMLM code, which shares code and distilled dataset with OAXE. Unfortunately, our re-implementation cannot re-produce the reported BLEU scores on all benchmarks (e.g. the same on WMT14 De-En, while 1.8 BLEU lower on En-De). Accordingly, we re-used the reported results when available, and only used our re-implementation in Table 2.

Model	WMT14		WMT16		WMT17	
	En-De	De-En	En-Ro	Ro-En	En-Zh	Zh-En
Autoregressive						
Transformer	27.6	31.4	34.3	34.0	34.3	23.7
+ Knowledge Distillation	27.8	31.3	-	-	34.4	24.0
Non-Autoregressive						
Bag-of-ngrams Loss (Shao et al., 2020)	20.9	24.6	28.3	29.3	-	-
Hint-based Loss (Li et al., 2019)	21.1	25.2	-	-	-	-
Flowseq (Ma et al., 2019)	21.5	26.2	29.3	30.4	-	-
Bigram CRF (Sun et al., 2019)	23.4	27.2	-	-	-	-
EM+ODD (Sun & Yang, 2020)	24.5	27.9	-	-	-	-
GLAT (Qian et al., 2020)	25.2	29.8	31.2	32.0	-	-
Imputer (Saharia et al., 2020)	25.8	28.4	32.3	31.7	-	-
CMLM (Ghazvininejad et al., 2019)	18.3	22.0	27.6	28.6	23.6	13.5
CMLM + AXE (Ghazvininejad et al., 2020)	23.5	27.9	30.8	31.5	30.9	19.8
CMLM + *Order-Agnostic XE* (Ours)	**26.1**⇑	**30.2**⇑	**32.4**⇑	**33.3**⇑	**32.9**⇑	**22.1**⇑

Table 3. The performance of CMLMs trained with OaXE, compared to other purely non-autoregressive methods. We implemented our model on top of the released CMLM models (Ghazvininejad et al., 2019). "⇑" denotes significantly better than the CMLM with $p < 0.01$. The results of CMLM are slightly better than those reported by Ghazvininejad et al. (2019), since we use de-duplication to remove repetitive tokens in the generated output. We ue the results of AXE reported by Ghazvininejad et al. (2020).

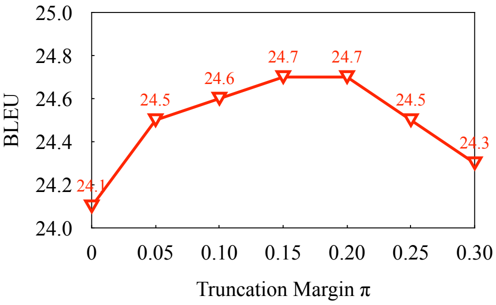


Figure 4. Impact of truncation margin on the En-De validation.

$\{0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30\}$ on the WMT14 En-De validation set. Intuitively, the higher π is, the more predictions are dropped. When π increases from 0 to 0.15, we achieved 0.6 BLEU improvement by dropping likely invalid predictions. When π further increases, the valid predictions can be mistakenly dropped, which leads to a performance drop. We used $\pi = 0.15$ for all language pairs in the following experiments.

4.3. Translation Performance

State of the Art In addition to the closely related XE variants, we also compare the performance of CMLM with OaXE against 7 strong baseline models: 1) bag-of-ngrams training (Shao et al., 2020); 2) hint-based training (Li et al., 2019); 3) Flowseq – a latent variable model based on generative flow (Ma et al., 2019); 4) the CRF-based semi-autoregressive model (Sun et al., 2019); 5) EM+ODD – an EM based method to remove the multimodality in the training dataset (Sun & Yang, 2020); 6) Glancing-based training (Qian et al., 2020); 7) Imputer – a NAT model with latent alignments (Saharia et al., 2020).

We followed (Li et al., 2019; Sun & Yang, 2020; Saharia et al., 2020) to use de-duplication trick to remove repetitive tokens in the generated output.[4] Table 3 shows that our approach achieves the highest BLEU scores in all benchmarks. Our approach achieves an averaged improvement of 2.1 BLEU over the advanced AXE on all benchmarks. Encouragingly, OaXE-trained CMLMs outperforms the best fully NAT model (Imputer) by 1.0 BLEU on average, setting a new state-of-the-art for fully non-autoregressive models on the major NAT benchmarks.[5]

Raw Data In this experiment, we evaluated the performance of fully NAT models that do not rely on any external resources (e.g., AT models for knowledge distillation). Table 4 shows the performance of NAT models that are trained on raw data without knowledge distillation. OaXE-trained CMLMs still significantly outperform the other NAT models in the raw data scenarios, demonstrating the effectiveness

[4] Without de-duplication, the averaged BLEU score of XE is 0.07 lower, while that of OaXE is the same ($\{25.8, 30.1, 32.2, 33.1, 32.9, 22.5\}$ vs $\{26.1, 30.2, 32.4, 33.3, 32.9, 22.1\}$). Our general conclusions still hold without de-duplication.

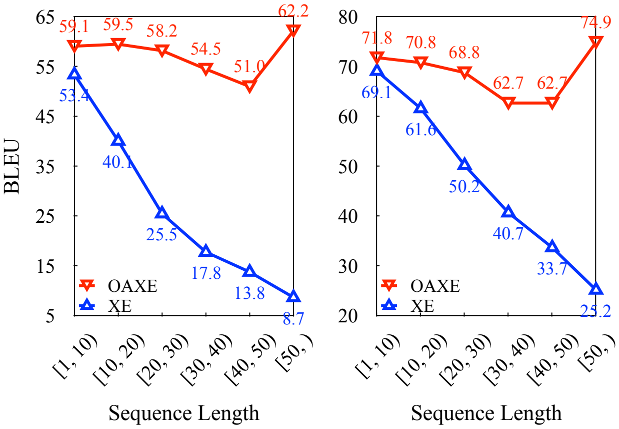
[5] When using Transformer-Base for distillation, the BLEU scores on WMT14 En-De and De-En are respectively 25.9 and 29.8, which still outperform previous work.

Model	WMT14	
	En-De	De-En
CTC Loss (Libovický & Helcl, 2018)	17.7	19.8
Flowseq (Ma et al., 2019)	18.6	23.4
Imputer (Saharia et al., 2020)	15.6	-
CMLM (Ghazvininejad et al., 2019)	10.6	-
CMLM + AXE (2020)	20.4	24.9
CMLM (Our reimplemented)	10.6	15.1
CMLM + *Order-Agnostic XE* (Ours)	**22.4**$^{\Uparrow}$	**26.8**$^{\Uparrow}$

Table 4. The performance of OAXE-based CMLM, compared to other non-autoregressive methods on raw data.

Model	Single		Multiple	
	BLEU	Δ	BLEU	Δ
Raw Data				
CMLM	11.0	-	28.1	-
CMLM + OAXE	22.7	+11.7	57.5	**+29.4**
Distillation				
CMLM	18.6	-	50.7	-
CMLM + OAXE	26.2	+7.6	68.0	**+17.3**

Table 5. BLEU scores on WMT14 En-De test sets with different number of references (e.g., "Single" and "Multiple"). "Δ" denotes the improvement of OAXE over the vanilla CMLM model.



(a) Raw Data

(b) Knowledge Distillation

Figure 5. Performance of the generated translations with respect to the lengths of the reference sentences. Results are reported on the WMT14 En-De test sets with multiple references (e.g., Table 5), which can better measure the quality of the generated outputs.

Model	Ja-En
Transformer-Base	28.5
CMLM (Our reimplemented)	20.9
CMLM + *Order-Agnostic XE* (Ours)	28.4

Table 6. Performance on WAT17 Japanese-English data.

and robustness of our approach. Encouragingly, OAXE narrows the performance gap between training on the raw data and on the distilled data, moving towards independent training of NAT models without relying on external models (e.g., AT models for knowledge distillation).

Multiple References As aforementioned, the multimodality problem that NAT models suffer is mainly due to the uncertainty nature of translation. In this experiment, we measure how well NAT models can handle the translation uncertainty using multiple semantically equivalent references. Specifically, we use the dataset released by Ott et al. (2018) for evaluating translation uncertainty, which consists of ten human translations for 500 sentences taken from the WMT14 En-De test set. Table 5 lists the results. Two observations can be made. Firstly, OAXE achieves more significant improvement over the vanilla XE when measured by multiple references. One possible reason is that OAXE produces more diverse translations with different word orders, which can be better measured by multiple references.

Secondly, when measured by multiple references, the improvement of OAXE on the raw data is even more significant than that on the distilled data. This is intuitive, since the raw data is more complex in terms of "modes" (alternative translations for an input), and knowledge distillation reduces the modes to help NAT to model the variations in the output data (Zhou et al., 2020; Ding et al., 2021a;b). Accordingly, knowledge distillation is essential for the XE-based NAT, which is weak at modeling multimodality. In contrast, OAXE reduces the modes by relaxing the order constraint, thus releases the reliance on the knowledge distillation.

Sequence Lengths We also investigate the model performance for different sequence lengths. We split the test sets into different buckets based on the reference sentence length, indicating whether a system does better or worse at shorter or longer sentences. Figure 5 shows that results on the sampled WMT14 En-De test set with multiple references (i.e., Table 5), which can better measure the translation quality. As seen, the performance of XE drops rapidly when the sequence length increases. Our OAXE significantly improves performance, especially on longer sentences by further relaxing the strict order constraint, which is especially important for handling long sentences. We also compare OAXE with the vanilla XE and AXE on the full test set of WMT14 En⇔De (Appendix A.5), in which OAXE consistently outperforms both XE and AXE.

Syntactically Divergent Language Pair To investigate whether our approach would work for more syntactically

Model	WMT14	
	En-De	De-En
Gold Test Set	0.04%	0.02%
Raw Data		
CMLM	22.80%	22.70%
CMLM + OaXE Loss	1.70%	**1.39%**
+ Loss Truncation	**1.57%**	1.67%
Knowledge Distillation		
CMLM	16.72%	12.31%
CMLM + OaXE Loss	0.90%	0.88%
+ Loss Truncation	**0.81%**	**0.73%**
CMLM + AXE (2020)	1.41%	1.03%

Table 7. Repeated token percentage on the test sets. The results of AXE are reported by Ghazvininejad et al. (2020).

Model	WMT14	
	En-De	De-En
Raw Data		
CMLM	1.30	0.64
CMLM + OaXE Loss	1.00	0.55
+ Loss Truncation	**0.33**	**0.29**
Knowledge Distillation		
CMLM	0.55	0.47
CMLM + OaXE Loss	0.30	0.27
+ Loss Truncation	**0.19**	**0.19**

Table 8. Normalized Corpus-level multimodality (NCM) scores of the generated outputs on the test sets.

divergent language pairs, we conducted experiments on the Japanese-English pair, which differ greatly in the word orders (subject-verb-object language vs. subject-object-verb language). Specifically, we used the first two sections of WAT17 Japanese-English dataset, which consists of 1.9M sentence pairs. We employed Transformer-Base model for distilling the training data. Table 6 shows the results on the WAT17 Japanese-English data. OaXE significantly improves over the fully CMLM model from 20.9 to 28.4 with a Transformer-Base teacher (28.5).

4.4. Analysis

In this section, we present a qualitative analysis to provide some insights where OaXE improves over cross entropy by alleviating the multimodality problem. To better understand the proposed OaXE, we report the results for the two newly introduced components – OaXE loss and loss truncation. To make fair comparison with previous work (Ghazvininejad et al., 2019; 2020), we analyze the generated outputs of the full WMT14 En⇔De test sets (e.g., Tables 3 and 4).

OaXE Reduces Token Repetitions Repeated token percentage is a commonly-used metric of measuring multimodality in a NAT model (Ghazvininejad et al., 2020; Saharia et al., 2020). A NAT model may consider many possible translations at the same time due to the independent predictions of target words. Accordingly, the NAT output typically contains many repetitive tokens.

Table 7 shows the repetition rate for models trained on both distilled and raw datasets. We also list the results of AXE reported by (Ghazvininejad et al., 2020) for reference. OaXE significantly reduces the repetition percentage over the vanilla XE by a multiplicative factor of 13 on both raw and distilled data. In addition, the proposed OaXE outperforms AXE in token repetition, which is consistent with the translation results in Tables 3 and 4.

OaXE Increases Prediction Confidence Conditional entropy is another metric of measuring corpus-level multimodality for NAT model (Sun & Yang, 2020). Since a NAT model considers many possible translations at the same time, it tends to spread too much probability mass over the space of sequences and thus generates translations which have lower generation probabilities. Formally, given a bilingual corpus $(\mathcal{X}, \mathcal{Y})$, the Normalized Corpus-level multimodality (NCM) (Sun & Yang, 2020) is calculated as

$$\mathrm{NCM}_{\mathcal{X}}(\mathcal{Y}) = \frac{\mathbb{E}_{(X,Y)\sim(\mathcal{X},\mathcal{Y})}\left[-\log P\left(Y \mid X\right)\right]}{\mathbb{E}_{Y\sim\mathcal{Y}}[|Y|]},$$

where $|Y|$ is the length of the output sequence Y.

We use NCM to compare the outputs generated by different NAT models on the same test set, as listed in Table 8. The proposed OaXE consistently outperforms XE across language pairs and datasets, which proves that OaXE can increase output confidence by better estimating the probability of generated output in potentially different orders.

OaXE Improves Generation Fluency Some researchers may doubt that removing the order constraint in the OaXE loss will make NAT models generate ungrammatical output (e.g., correct predictions in wrong positions). In response to this problem, we measure the fluency with language models released by Fairseq,which are trained on the News Crawl corpus for the target language. To better evaluate the fluency of the generated output, we use a practical trick *de-duplication* (Saharia et al., 2020) to remove the repetitive tokens.

Results in Table 10 can dispel such doubt by showing that OaXE consistently improves fluency in all cases. We attribute the improvement of fluency to that XE initialization can avoid a large portion of invalid orderings in the search

Table 9. Example for De⇒En translation. CMLM often generates repetitive tokens, while OAXE suffers less from these mistakes.

Source	Passagiere schimpfen häufig über Gepäckzuschläge und andere Gebühren , doch Fluggesellschaften greifen gern darauf zurück .
Reference	Passengers often grumble about baggage charges and other fees , but airlines love them.
CMLM	Passengers often abuse at baggage baggage charges charges and charges charges , airlines airlines like to use them .
+ OAXE	Passengers often complain about baggage charges and other charges , but airlines like to resort to them .

Model	**WMT14**	
	En-De	**De-En**
Gold Test Set	37.0	43.8
Raw Data		
CMLM	511.6	351.8
CMLM + OAXE Loss	146.9	120.1
+ Loss Truncation	**123.1**	**105.4**
Knowledge Distillation		
CMLM	205.6	177.6
CMLM + OAXE Loss	82.6	83.7
+ Loss Truncation	**69.4**	**77.2**

Table 10. PPLs of the generated outputs on the test sets. Lower PPL score denotes better fluency.

space, and the loss truncation can drop invalid predictions to further improve the fluency of the searched ordering. Encouragingly, OAXE trained on the raw data outperforms the standard XE trained on the distilled data, showing that OAXE has the ability to directly learn from the raw data of complex order modes.

Case Study Table 9 shows a translation example, where the vanilla CMLM output contains many repetitive tokens due to suffering from severe multimodality problem. In addition, it also misses some important function words like “other” and “but”, which harm the output fluency. The proposed OAXE can successfully correct these mistakes.

5. Related Work

Compared with AT models, NAT breaks the conditional dependencies among target tokens, which greatly degrades the model performance due to the multimodality problem. To bridge the performance gap between NAT and AT models, several approaches try to build dependencies among the target tokens. For example, Shao et al. (2020) use bag of n-grams loss to capture the target-side sequential dependency, and Ma et al. (2019) model the joint distribution of all predictions with generative flow. Shu et al. (2020) introduce a sequence of continuous latent variables to capture the uncertainty in the target sentence. In this work, we improve the cross entropy loss to ameliorate the effect of multimodality by better handling word reordering. Experimental results show that our approach consistently outperforms previous studies on alleviating the multimodality problem, and set new state of the art on the major WMT benchmarks.

Knowledge distillation is a crucial early step in the training of almost all existing NAT models. Gu et al. (2018) claim that NAT suffers from the multimodality problem (i.e., multiple lexical translations for a source word), and knowledge distillation can simplify the dataset, which is empirically validated by Zhou et al. (2020). Ren et al. (2020) reveal that knowledge distillation reduces the token dependency in target sequence and thus improves the accuracy of NAT models. OAXE narrows the performance gap training on raw data and on the distilled data for its superiority on handling multimodality, which opens up the possibility that NAT models can be trained directly on the raw data without knowledge distillation.

6. Conclusion

In this work, we propose a new training objective OAXE to handle word reordering in translation, which is one common source to the multimodality problem. We validated our approach on six major WMT benchmarks, and showed that OAXE significantly improves translation performance over the standard XE-based NAT models, which closely match the performance of AT models. OAXE-based NAT also narrows the performance gap between training on raw data and on the distilled data, indicating the potential to train purely non-autoregressive models without knowledge distillation. Future directions include validating OAXE on other tasks (e.g., dialogue and summarization) and other NAT models.

Acknowledgements

We thank all anonymous reviewers for their insightful comments. This project is jointly supported by Rhino-Bird Elite Training Program of Tencent AI Lab and the National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative. Any opinions and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore.

References

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Ding, L., Wang, L., Liu, X., Wong, D. F., Tao, D., and Tu, Z. Rejuvenating low-frequency words: Making the most of parallel data in non-autoregressive translation. In *ACL*, 2021a.

Ding, L., Wang, L., Liu, X., Wong, D. F., Tao, D., and Tu, Z. Understanding and improving lexical choice in non-autoregressive translation. In *ICLR*, 2021b.

Ghazvininejad, M., Levy, O., Liu, Y., and Zettloyer, L. Mask-predict: Parallel decoding of conditional masked language models. In *EMNLP*, 2019.

Ghazvininejad, M., Karpukhin, V., Zettlemoyer, L., and Levy, O. Aligned cross entropy for non-autoregressive machine translation. In *ICML*, 2020.

Gu, J., Bradbury, J., Xiong, C., Li, V. O., and Socher, R. Non-autoregressive neural machine translation. In *ICLR*, 2018.

Gu, J., Wang, C., and Zhao, J. Levenshtein transformer. In *NeurIPS*, 2019.

Huang, J., Guestrin, C., Jiang, X., and Guibas, L. Exploiting probabilistic independence for permutations. In *AISTATS*, 2009.

Kang, D. and Hashimoto, T. Improved natural language generation via loss truncation. In *ACL*, 2020.

Kasai, J., Cross, J., Ghazvininejad, M., and Gu, J. Non-autoregressive machine translation with disentangled context transformer. In *ICML*, 2020.

Kasai, J., Pappas, N., Peng, H., Cross, J., and Smith, N. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *ICLR*, 2021.

Kim, Y. and Rush, A. M. Sequence-level knowledge distillation. In *EMNLP*, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Kong, X., Tu, Z., Shi, S., Hovy, E., and Zhang, T. Neural machine translation with adequacy-oriented learning. In *AAAI*, 2019.

Kreutzer, J., Foster, G., and Cherry, C. Inference strategies for sequence generation with conditional masking. In *EMNLP*, 2020.

Kuhn, H. W. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955.

Lee, J., Mansimov, E., and Cho, K. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *EMNLP*, 2018.

Li, Z., Lin, Z., He, D., Tian, F., Qin, T., Wang, L., and Liu, T. Hint-based training for non-autoregressive machine translation. In *EMNLP-IJCNLP*, 2019.

Libovický, J. and Helcl, J. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *EMNLP*, 2018.

Lopes, P. A., Yadav, S. S., Ilic, A., and Patra, S. K. Fast block distributed cuda implementation of the hungarian algorithm. *Journal of Parallel and Distributed Computing*, 130:50–62, 2019.

Ma, S., Sun, X., Wang, Y., and Lin, J. Bag-of-words as target for neural machine translation. In *ACL*, 2018.

Ma, X., Zhou, C., Li, X., Neubig, G., and Hovy, E. Flowseq: Non-autoregressive conditional sequence generation with generative flow. In *EMNLP-IJCNLP*, 2019.

Ott, M., Auli, M., Grangier, D., and Ranzato, M. Analyzing uncertainty in neural machine translation. In *ICML*, 2018.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

Post, M. A call for clarity in reporting bleu scores. In *WMT*, 2018.

Qian, L., Zhou, H., Bao, Y., Wang, M., Qiu, L., Zhang, W., Yu, Y., and Li, L. Glancing transformer for non-autoregressive neural machine translation. *ArXiv*, abs/2008.07905, 2020.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. Sequence Level Training with Recurrent Neural Networks. In *ICLR*, 2016.

Ren, Y., Liu, J., Tan, X., Zhao, Z., Zhao, S., and Liu, T.-Y. A study of non-autoregressive model for sequence generation. In *ACL*, 2020.

Saharia, C., Chan, W., Saxena, S., and Norouzi, M. Non-autoregressive machine translation with latent alignments. In *EMNLP*, 2020.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. In *ACL*, 2016.

Shao, C., Zhang, J., Feng, Y., Meng, F., and Zhou, J. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. In *AAAI*, 2020.

Shen, S., Cheng, Y., He, Z., He, W., Wu, H., Sun, M., and Liu, Y. Minimum risk training for neural machine translation. In *ACL*, 2016.

Shu, R., Lee, J., Nakayama, H., and Cho, K. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. In *AAAI*, 2020.

Sun, Z. and Yang, Y. An em approach to non-autoregressive conditional sequence generation. In *ICML*, 2020.

Sun, Z., Li, Z., Wang, H., Lin, Z., He, D., and Deng, Z.-H. Fast structured decoding for sequence models. In *NeurIPS*, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017.

Zhou, C., Neubig, G., and Gu, J. Understanding knowledge distillation in non-autoregressive machine translation. In *ICLR*, 2020.