A Study of Syntactic Multi-Modality in Non-Autoregressive Machine Translation

Anonymous ACL submission

Abstract

It is difficult for non-autoregressive translation (NAT) models to capture the multi-modal distribution of target translations due to their conditional independence assumption, which is known as the "multi-modality problem", including the lexical multi-modality and the syntactic multi-modality. While the first one has been well studied, the syntactic multi-modality brings severe challenge to the standard cross entropy (XE) loss in NAT and is under studied. In this paper, we conduct a systematic study on the syntactic multi-modality problem. Specifically, we decompose it into short- and long-range syntactic multi-modalities and evaluate several recent NAT algorithms with advanced loss functions on both carefully designed synthesized datasets and real datasets. We find that the Connectionist Temporal Classification (CTC) loss and the Order-Agnostic Cross Entropy (OAXE) loss can better handle short- and long-range syntactic multi-modalities respectively. Furthermore, we take the best of both and design a new loss function to better handle the complicated syntactic multi-modality in real-world datasets. To facilitate practical usage, we provide a guide to use different loss functions for different kinds of syntactic multi-modality.

1 Introduction

Traditional Neural Machine Translation (NMT) models predict each target token conditioned on previous generated tokens in an autoregressive way (Vaswani et al., 2017), resulting in high latency in inference. Non-Autoregressive Translation (NAT) models generate all the target tokens in parallel (Gu et al., 2018), significantly reducing inference latency. A disadvantage of NAT is that it suffers from the multi-modality problem (Gu et al., 2018) when a source sentence corresponds to multiple correct translations (Ott et al., 2018).

There are two types of multi-modalities: the lexical one and the syntactic one. The former one has been adequately studied (Gu et al., 2018; Zhou et al., 2019; Ding et al., 2020), while the latter one brings severe challenges to the widely used cross entropy (XE) loss in NAT. With standard XE loss, the generated tokens are required to be strictly aligned with ground truth tokens in the same positions, which fails to provide positive feedback for correctly predicted words on different positions as shown in Fig. 1a. Therefore, advanced loss functions are introduced to provide better feedback for NAT training: Connectionist Temporal Classification (CTC) loss (Libovický and Helcl, 2018) considers all possible monotonic alignments between a generated sequence and the ground truth; Aligned Cross-Entropy (AXE) loss (Ghazvininejad et al., 2020) selects the best monotonic alignment; and Order-Agnostic cross entropy (OAXE) loss (Du et al., 2021) calculates the XE loss with the best alignment based on maximum bipartite matching algorithm.

Even if with those advanced loss functions, we find they do not perform consistently across datasets and languages. In addition, diverse grammar rules in natural language (Comrie, 1989) implies the existence of different kinds of syntactic multi-modality. Inspired by Odlin (2008), we categorize the syntactic multi-modality into two sub types: the long-range and short-range ones. The long-range multi-modality is mainly caused by long-range word order diversity (e.g., an adverbial of place may appear at the beginning or the end of a sentence). The short-range multi-modality is mainly caused by short-range word order diversity (e.g., an adverb may appear either in front of or behind the corresponding verb) and optional words (e.g., in some languages, determiners and prepositions may be optional (Ott et al., 2018)). Based on the above categorization of syntactic multi-modality, we further ask two research questions: (1) Which kinds of syntactic multi-modality do these loss functions excel at respectively? (2)

How to better address this problem by taking advantage of different loss functions?

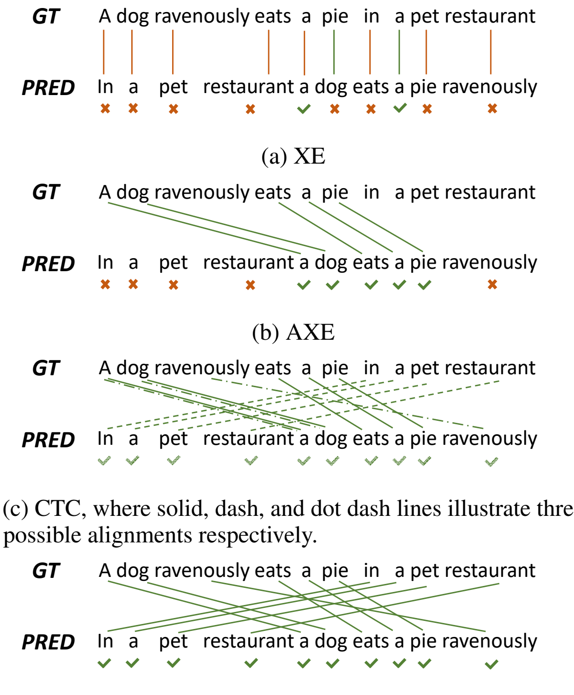
In this paper, we conduct a systematic study to answer these questions:

- Since the short-range and long-range syntactic multi-modalities are usually entangled together in real-world datasets, we first design synthesized datasets to decouple them to better evaluate existing NAT algorithms (§3). We find that the CTC loss (Libovickỳ and Helcl, 2018) can better handle the short-range syntactic multi-modality while the OAXE loss (Du et al., 2021) is good at the long-range one. Though carefully designed, the synthesized datasets are still different from the real-world datasets. Accordingly, we further conduct analyses on real-world datasets (§4), which also show consistent findings with that in synthesized datasets.
- We design a new loss function that takes the best of both CTC and OAXE, and performs better to handle the short- and long-range syntactic multi-modalities simultaneously (§5), as verified by experiments on benchmark datasets including WMT14 EN-DE, WMT17 EN-FI, and WMT14 EN-RU. Moreover, we further provide a practical guide to use different loss functions for different kinds of syntactic multi-modality (§5).

2 Background

Non-Autoregressive Translation Given the source sentence $x = (x_1, x_2, ..., x_{T_x})$, traditional NMT model generates the target sentence $y = (y_1, y_2, ..., y_{T_y})$ from left to right and token by token: $P(y|x) = \prod_{t=1}^{T_y} P(y_t|y_{<t}, x; \theta_{\text{enc}}, \theta_{\text{dec}})$, where $y_{<t}$ indicates the target tokens generated before the t-th timestep, T_x and T_y denote the length of source and target sentence, θ_{enc} and θ_{dec} denote the encoder and decoder parameters respectively. This autoregressive way suffers from high latency while inference. Non-Autoregressive Translation (NAT) (Gu et al., 2018) is proposed to reduce the inference time by generating the whole sequence in parallel, $P(y|x) = P(T_y|x) \cdot \prod_{t=1}^{T_y} P(y_t|x; \theta_{\text{enc}}, \theta_{\text{dec}})$, where $P(T_y|x)$ indicates the length prediction function. While the inference speed is boosted, the translation accuracy is sacrificed due to that target tokens are generated conditional independently.

Multi-Modality Problem The multi-modality problem (Gu et al., 2018; Zhou et al., 2019) indicates that one source sentence may have multiple correct target translations, which brings challenges to NAT models as they generate each target token independently. Specifically, we categorize the multi-modality problem into two sub-problems, i.e., lexical and syntactic multi-modalities. The lexical multi-modality indicates that a source token can be translated into different target synonym tokens (i.e., "thank you" in English can be translated into both "Danke" or "Vielen Dank" in German), while the syntactic multi-modality indicates the inconsistency of word-orders (e.g., an adverb may appear either in front of or behind the corresponding verb) and the existence of optional words between source and target languages (e.g., in some languages, determiners and prepositions may be optional) (Ott et al., 2018). The lexical multi-modality problem has been adequately studied in recent works. Sequence-level knowledge distillation (Gu et al., 2018; Zhou et al., 2019) is shown capable to reduce the lexical diversity of the dataset and thus alleviate the problem. Some works also introduce extra loss functions such as KL-divergence (Ding et al., 2020) and bag-of-ngram (Shao et al., 2020) to alleviate the lexical multi-modality problem.


GT
A dog ravenously eats a pie in a pet restaurant
PRED
In a pet restaurant a dog eats a pie ravenously

(a) XE

(b) AXE

(c) CTC, where solid, dash, and dot dash lines illustrate three possible alignments respectively.

(d) OAXE

Figure 1: The illustration of different loss functions, where "GT" stands for ground truth, "PRED" stands for predicted sequence, the green check indicates that credit is provided to the token.

On the contrary, there still lacks a systematic study on the syntactic multi-modality problem. Generally, it is difficult to solve this problem because the order and optional words vary from different languages. For example, the word order of Russian is quite flexible (Kallestinova, 2007), thus the syntactic multi-modality may exist more frequently in Russian corpora. In contrast, the structure of English sentences is mostly subject–verb–object (SVO) (Givón, 1983), which results in less variation on word order. In this paper, we categorize the syntactic multi-modality problem into short-range and long-range instances, and provide detailed analyses accordingly.

Loss Functions in NAT Standard cross-entropy (XE) loss requires the predicted tokens are strictly aligned with ground truth tokens, which fails to deal with the syntactic multi-modality problem. Different loss functions are proposed to solve the problem, and here we consider some most recent works. The CTC loss sums XE losses of all possible monotonic alignments and has been widely used in speech recognition (Graves et al., 2006, 2013), and the effectiveness of the CTC loss in NAT has been validated (Libovickỳ and Helcl, 2018; Gu and Kong, 2020). AXE (Ghazvininejad et al., 2020) selects the monotonic alignment between the predicted sequence and the ground truth with the minimum XE loss. OAXE (Du et al., 2021) further relaxes the position constraint and only considers the best alignment. The illustration for each loss function is provided in Fig. 1. Though effective in different datasets, these works ignore fine-grain features of the multi-modality problem such as short/long syntactic multi-modalities. In this work, we analyse the performance of these loss functions in different syntactic scenarios, and provide a practical guide to use appropriate loss functions for different kinds of syntactic multi-modality.

3 Analyses on Synthesized Datasets

To make fine-grained analyses on the syntactic multi-modality problem, we first categorize it into long-range and short-range types, where the long-range one is mainly caused by long-range word order diversity, and the short-range one is mainly caused by short-range word order diversity and optional words. Then, we would like to evaluate the accuracy of different losses on different types of syntactic multi-modality. However, in real-world corpora, the different types are usually entangled, making it difficult to control and analyse one aspect without changing the other. Thus, we construct synthesized datasets based on phrase structure rules (Chomsky, 1959) to manually control the degree of syntactic multi-modality in different aspects, and evaluate the performance of different existing techniques.

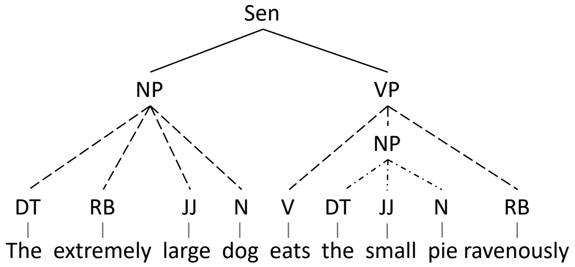


Figure 2: An illustration of generating a syntax tree for a source sentence. In the first iteration, "Sen" consists of ("NP", "VP") as the solid lines. In the second iteration, "NP" consists of ("DT", "RB", "JJ", "N") and "VP" consists of ("V", "NP", "RB") as the dash lines. In the third iteration, "NP" consists of ("DT", "JJ", "N") as the dot-and-dash lines.

3.1 Synthesized Datasets

We first employ phrase structure rules (Chomsky, 1959) to synthesize the source sentences, where the rules are based on the syntax of languages. Considering that translation can be decomposed to word reordering and word translation (Bangalore and Riccardi, 2001; Sudoh et al., 2011), we then "translate" the synthesized source sentences to synthesized target sentences in two steps: 1) word reordering by changing its syntax tree; 2) and word translation by substituting the source words into target words.

Source Sentence Synthesis. We first generate the syntax tree of the source sentence. Specifically, we use the notations of the constituents in syntax tree according to the Penn Treebank syntactic and part of speech (POS) tag sets[1] (Marcus et al., 1993), and generate the syntax tree of a source sentence as following (Rosenbaum, 1967):

- Sen $\rightarrow$ NP VP,
- NP $\rightarrow$ (DT) (RB)* (JJ)* N,
- VP $\rightarrow$ V (NP) (RB)*,

[1] "Sen":sentence; "NP": noun phrase; "VP": verb phrase; "DT": determiner; "JJ": adjective; "RB": adverb; "N": noun; "V": verb.

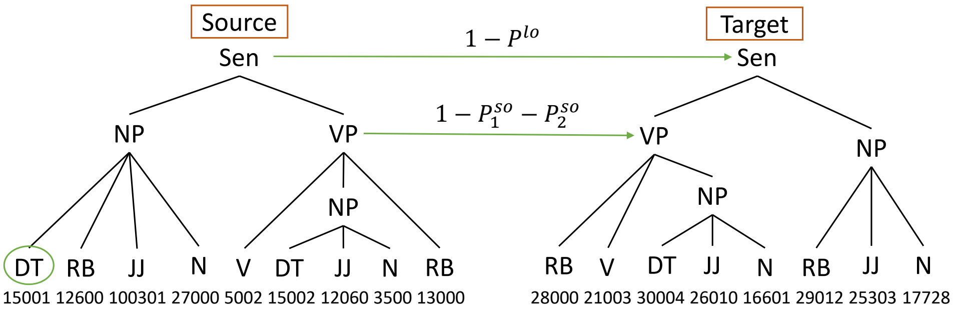


Figure 3: An illustration of "translation", where the constituent order of "Sen" is changed to "VP NP" with probability $1 - P^{lo}$, the constituent order of "VP" is changed to "RB V NP" with probability $1 - P_1^{so} - P_2^{so}$, and the circled "DT" is removed with probability P^{op}. Meanwhile, the numbers in the source sentence are replaced with the ones in the target sentence based on mappings.

where the constituent on the left side of the arrow consists of the constituents on the right side in sequence, "$(\cdot)$" means that the constituent is optional, and "$(\cdot)$*" denotes that the constituent is not only optional but can also be repetitive. For each sentence, we start with a single constituent Sen and iteratively decompose "Sen", "NP", and "VP" according to the rules until all the constituents are decomposed to "DT", "JJ", "RB", "V", and "N". An illustration of generating a syntax tree is depicted in Fig. 2. To synthesize the source sentence according to the syntax tree, we use numbers as the words in the synthesized source sentences and use different ranges of numbers to represent words with different POS, where the details of the ranges are provided in Appendix A. Then, a number is randomly sampled in the corresponding range for each word in the syntax tree.

Word Reordering. To introduce syntactic multi-modality, we consider multiple possible rules for "Sen", "NP", and "VP" in the target sentences. Specifically, we consider three options: 1) The word order of "Sen" is with probability P^{lo} to be the same with the source sentence (i.e., NP VP) and with probability $\left(1 - P^{lo}\right)$ to swap the "NP" and "VP" (i.e., VP NP), which represents for the long-range word order; 2) For the word order in "VP", it is considered to be the same with the source sentence with probability P_1^{so}, place "RB" between "V" and "NP" with probability P_2^{so}, and place "RB" before "V" with probability $(1 - P_1^{so} - P_2^{so})$, which represents for the short-range word order; 3) To introduce the syntactic multi-modality of optional words, we change the existence of "DT" in each "NP" of the source sentence with probability P^{op} (i.e, remove "DT" if it exists in the source sentence and add "DT" if it does not exist in the source sentence).

Probability	Default	Effect
P^{lo}	1	long-range word order
P_1^{so}	1	short-range word order
P_1^{so}	0	short-range word order
P^{op}	0	optional words

Table 1: Default values of the probabilities to adjust the syntactic multi-modality.

Word Translation. Same as in the source sentences, we use different range of numbers to represent words with different POS in target sentences. To do the word translation, we first randomly build mappings between the source and target words with different POS respectively. Since we focus on studying the syntactic multi-modality, we consider each source word is mapped to a single target word to eliminate the lexical multi-modality. Then, we replace the words in the source sentence based on the mappings to generate the target sentence. An illustration of "translation" is shown in Fig. 3.

3.2 Experiments and Analyses

We conduct experiments to compare existing loss functions on different kinds of syntactic multi-modality on the synthesized datasets, by changing the probabilities (i.e., P^{op}, P_1^{so}, P_2^{so}, and P^{lo}) as listed in Table 1. In the following, we first provide the experimental settings, then show the results on the long-range and short-range syntactic multi-modalities, and finally conclude the key findings.

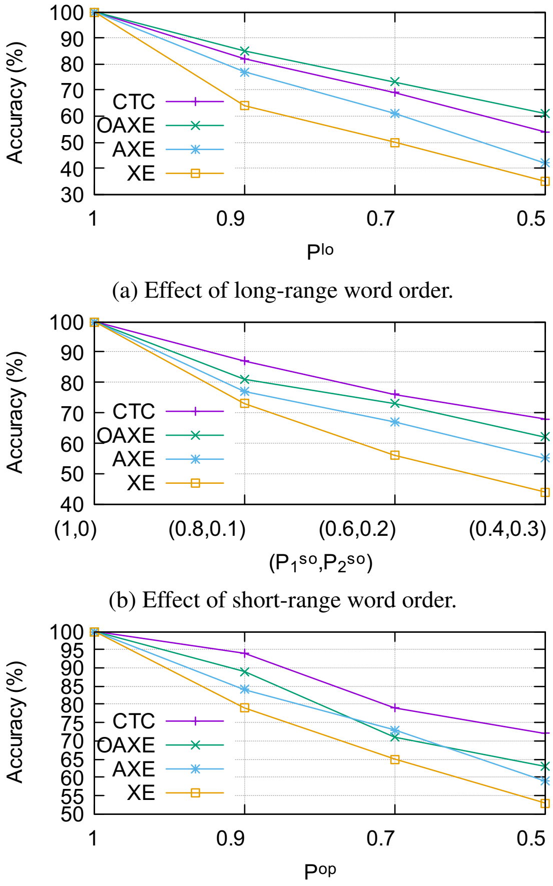
Experimental Settings. We consider two separate vocabularies for the source and target sentences, each containing 15K words. 0.3M, 5K, and 5K synthesized sentence pairs are generated as training, validation, and test sets respectively. We use the same hyper-parameters in the transformer-base model (Vaswani et al., 2017), which is commonly used in the NAT models (Gu et al., 2018; Du et al., 2021; Saharia et al., 2020). All settings are trained on 4 Nvidia V100 GPUs with 16k tokens in a batch. For the model with OAXE loss, we train the first 50K updates with XE loss and the next 50K updates with OAXE loss (Du et al., 2021). For the other losses, we train the model for 100K updates. The length of the decoder input is set as twice the length of the source sequence for CTC loss (Saharia et al., 2020), while the golden target length is used for OAXE, AXE, and XE. To evaluate the accuracy of the predicted sequence, we first calculate the longest common sub-sequence between the predicted and the golden sequences, and then the accuracy is defined as the ratio between the length of the longest common sub-sequence and the golden sequence. The accuracy on the test set is calculated as the average accuracy among all the predicted sentences.

Long-Range Syntactic Multi-modality. To consider the effect of long-range diversity, we change the corresponding probability P^{lo}, while keeping the others unchanged to eliminate the short-range syntactic multi-modality. It is observed in Fig. 4a that CTC loss always performs better than AXE, and OAXE is the best with different degree of long-range multi-modality.

Short-Range Syntactic Multi-modality. Similarly, we only change the probabilities P_1^{so} and P_2^{so} to adjust the degree of short-range word order diversity. The results are shown in Fig. 4b, where OAXE loss performs better than AXE loss, and CTC loss outperforms all the other losses with varied degree of short-range word order diversity. In order to study the effect of optional words, we vary the probability P^{op} to change the existence of "DT". As shown in Fig. 4c, OAXE loss is slightly better than AXE loss, and CTC loss performs the best, indicating that CTC loss is superior in the syntactic multi-modality problem caused by optional words.

Analyses and Discussions. Based on the results in Fig. 4, we can get the following observations:

- OAXE loss is superior in handling the long-range syntactic multi-modality (i.e., long-range word order). OAXE loss is order-agnostic, which is able to provide fully positive feedback to the word in different positions compared to the ground truth sequence. Accordingly, OAXE is suitable for datasets with long-range word order diversity. Though it can deal with high diversity of word order, it may also incur wrong predictions on word order, which may be why OAXE is not suitable when the diversity only exists in short-range.



(a) Effect of long-range word order.

(b) Effect of short-range word order.

(c) Effect of optional words.

Figure 4: The accuracy of different loss functions on synthesized datasets.

- CTC loss is the best choice for dealing with short-range syntactic multi-modality (i.e., short-range word order and optional words). CTC loss is generally considered to handle monotonic matching, which seems not effective in handling the multi-modality caused by word order (Saharia et al., 2020). However, it is observed in Fig. 4a and 4b that CTC loss outperforms AXE and XE when dealing with long-range word order diversity and performs the best on the multi-modality caused by short-range word order. Since CTC considers all the monotonic alignments, it can partially provide positive feedback to the words with different order through multiple monotonic alignments. As shown in Fig. 1c, all the words can be considered in the three alignments.

Considering that AXE loss does not show superiority on any type of the syntactic multi-modality, we will only focus on CTC and OAXE losses in the following.

4 Analyses on Real Datasets

Though carefully designed, the synthesized sentence pairs consisting of numbers are still different from the real sentence pairs. Therefore, in this section, we validate the findings in Section 3 based on real datasets. Considering that different types of syntactic multi-modality are highly coupled in the real corpus, we conduct experiments on carefully selected sub-datasets from a corpus, to approximately decompose the syntactic multi-modality. In the following, we first show the details of the approach to decompose the syntactic multi-modality, and then provide the analytical results based on the real datasets.

Analytical Approach. In order to decompose the long-range and short-range types of syntactic multi-modality, we select sentences that only contain subject and verb phrases from a corpus, and divide them into two sub-datasets according to the relative order of subject and verb (i.e., subject first that denoted as “SV”, or verb first that denoted as “VS”). Meanwhile, we only consider the declarative sentence pairs in the corpus to eliminate the word order difference caused by mood. Following this method, the long-range multi-modality is eliminated in each sub-dataset (i.e., “SV” and “VS”), which can be used to evaluate the effect of short-range multi-modality. To analyse the long-range multi-modality, we can adjust the degree of long-range word order diversity by sampling data from the two sub-datasets with varied ratios, while roughly keeping the degree of short-range word order diversity unchanged. Specifically, considering that Russian is flexible on word order (Kallestinova, 2007) and it is feasible to select sentences on both the “SV” and “VS” order, we use an English-Russian (EN-RU) corpus from Yandex[2] that contains 1M EN-RU sentence pairs, from which we get 0.2M and 0.1M sentence pairs data with “SV” order and “VS” order respectively. For the models with CTC loss, we train for 300K updates. For the models with OAXE loss, we train with XE loss for 100K updates and then train with OAXE loss for 200K updates.

Table 2: BLEU scores of models with CTC and OAXE losses, where the models are evaluated on the WMT’19 EN-RU test set. The percentage of sentences with “RB V” among the sentences with both “RB V” and “V RB” orders are shown in column “RB V”. The percentage of sentences with “JJ N” among the sentences with both “JJ N” and “N JJ” orders are shown in column “JJ N”.

“SV”:“VS”	CTC	OAXE	“RB V”	“JJ N”
100% : 0%	17.7	16.5	68%	84%
75% : 25%	17.2	16.9	63%	82%
50% : 50%	16.8	17.3	70%	79%

Analytical Results. We keep the total number of sentence pairs in the training set as 0.2M (i.e., the number of Russian sentences in the “VS” sub-dataset), and change the ratio of sentence pairs sampled from two sub-datasets (i.e., “SV” and “VS”). The results are shown in Table 2, where the training parameters are the same as that used in Section 3. It is observed that CTC loss outperforms OAXE loss when all data samples are from the “SV” sub-dataset, which indicates that CTC loss performs better on short-range syntactic multi-modality problem. When the ratio of the data sizes on the two sub-datasets is changed to 75% : 25%, the gap between the performance of CTC and OAXE losses diminished, while CTC loss still performs slightly better than OAXE loss. When the ratio changed to 50% : 50%, model with OAXE loss becomes better than that with CTC loss. In summary, OAXE loss is better at handling long-range syntactic multi-modality while CTC loss is better on short-range syntactic multi-modality, which validates the key observations we obtained on the synthesized datasets in Section 3.

In order to demonstrate whether we have decomposed the long- and short-range syntactic multi-modalities, we verify whether the degree of short-range multi-modality remains almost the same when varying the degree of long-range multi-modality. We evaluate the short-range syntactic diversity based on the relative order between: 1) adverb and verb (“RB V”); 2) adjective and noun (“JJ N”). As shown in Table 2, when the ratio of the data sizes on the two sub-datasets varied from 100% : 0% to 50% : 50% (i.e., the ratio between “SV” and “VS” changes), the relative order on “RB

[2] https://translate.yandex.ru/corpus

V" and "RB V" (which can represent the degree of short-range word order diversity) does not vary much. These analyses verify the rationality of our analytical approach in this section.

5 Better Solving the Syntactic Multi-Modality Problem

As shown in previous sections, the CTC and the OAXE loss functions are good at dealing with short- and long-range syntactic multi-modalities respectively. While in real-world corpora, different types of multi-modalities usually occur together and vary in different languages. Accordingly, it may be better to use different loss functions for different languages. In this section, we first introduce a new loss function named Combined CTC and OAXE (CoCO), which takes advantage of both CTC and OAXE to better handle the long-range and short-range syntactic multi-modalities simultaneously, and then provide a guideline on how to choose the appropriate loss function for different scenarios.

5.1 CoCO Loss

To obtain a general loss that performs well at both types of multi-modalities, it is natural to combine the two loss functions studied above. However, the output length is mismatched between the models using CTC and OAXE, where the output length is required to be longer than the target sequence with CTC loss, and is required to be the same as the target sequence with OAXE loss. To solve this length mismatch problem, we consider using the same output length as in CTC loss, and modify the OAXE loss to make it suitable on this output length by allowing consecutive tokens in the output to be aligned with the same token in the reference sequence. The details of the modified OAXE loss are provided in Appendix B. Then, the proposed CoCO loss is defined as a linear combination of the CTC and modified OAXE losses as:

$$\mathcal{L}_{CoCO} = \lambda \mathcal{L}_{CTC} + (1 - \lambda)\mathcal{L}_{M-OAXE}, \quad (1)$$

where $\mathcal{L}_{M-OAXE}$ denotes the modified OAXE loss and λ is a hyper-parameter that balances the two losses.

5.2 Choosing Appropriate Loss Function

The degree of different types of multi-modalities varies among different languages. In order to find the insight to choose the appropriate loss function for different languages, we conduct experiments on several languages including Russian (RU), Finnish (FI), German (DE), Romanian (RO), and English (EN). These languages have different requirements on the positions of subject (S), verb (V), and object (O), which is one major influence factor on the large-range syntactic multi-modality. Specifically, the order in RU and FI is quite flexible, where all the 6 possible orders of "S", "V", and "O" are valid. In DE, the verb is required to be placed on the second position, which is called verb-second word order. Meanwhile, in RO and EN, the order is restricted to "SVO".

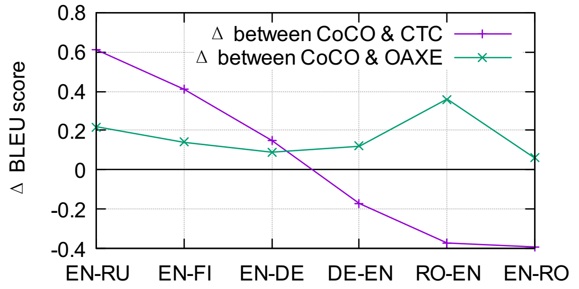


Figure 5: Comparing different losses on different language pairs.

We evaluate the accuracy of different loss functions (i.e., CTC, OAXE, and CoCO) on WMT'14 EN-RU, WMT'17 EN-FI, WMT'14 EN-DE, and WMT'16 EN-RO datasets with 1.5M, 2M, 4M, and 610K sentence pairs, respectively. The λ in COCO loss is set as 0.1 so that $\lambda\mathcal{L}_{CTC}$ and $(1-\lambda)\mathcal{L}_{M-OAXE}$ are in the same order of magnitude. Following Du et al. (2021), for the models with OAXE and CoCO loss, we first train with XE or CTC loss for 100K updates and then train with OAXE or CoCO loss for 200K updates, respectively. For CTC loss, we train for 300K updates. For decoding, we follow Gu and Kong (2020); Huang et al. (2021) to use beam search with language model scoring[3] for CTC and CoCO. The other training settings are the same as used in Section 3. We report the tokenized BLEU score to keep consistent with previous works. We show the difference values of BLEU score in Fig. 5 and provide the corresponding absolute BLEU scores in Appendix C. According to Fig. 5, we have several observations: 1) The proposed CoCO loss consistently improves the translation accuracy on all the language pairs compared to OAXE loss; 2) The CoCO loss outperforms CTC loss when the target language is with flexible word order or verb-second

[3] https://github.com/kpu/kenlm

Table 3: BLEU scores of NAT models.

Model	WMT14		WMT16	WMT14	WMT17	
	EN-DE	DE-EN	EN-RO	EN-RU	EN-FI	Speedup
Autoregressive						
Transformer	27.48	31.39	33.70	27.2	28.12	1.0 ×
Non-Autoregressive						
Vanilla NAT (Gu et al., 2018)	17.69	21.47	27.29	–	–	15.0 ×
BoN (Shao et al., 2020)	20.90	24.60	28.30	–	–	10.0 ×
AXE (Ghazvininejad et al., 2020)	23.53	27.90	30.75	–	–	15.3 ×
Imputer (Saharia et al., 2020)	25.80	28.40	32.30	–	–	18.6 ×
OAXE (CMLM) (Qian et al., 2020)	26.10	30.20	32.40	–	–	15.6 ×
GLAT (Qian et al., 2020)	26.39	29.84	32.79	–	–	14.6 ×
CTC (VAE) (Gu and Kong, 2020)	27.49	30.46	33.79	–	–	16.5 ×
CTC (GLAT) (Gu and Kong, 2020)	27.20	31.39	33.71	–	–	16.8 ×
CTC (DSLP) (Huang et al., 2021)	27.02	31.61	34.17	21.38	22.83	14.8 ×
CoCO (DSLP)	27.41	31.37	34.32	21.82	23.25	14.2 ×

word order (i.e., EN-RU, EN-FI, and EN-DE); 3) CTC loss performs the best when the target language is "SVO" language (i.e., DE-EN, RO-EN, and EN-RO).

We would also like to evaluate the CoCO loss based on SOTA NAT models. Though the proposed CoCO loss can be used in both iterative and non-iterative models, we only show the results on non-iterative models in this paper and leave the iterative models as future work. We use CoCO loss on a recently proposed Deeply Supervised, Layer-wise Prediction-aware (DSLP) transformer (Huang et al., 2021), which achieves competitive results. The details of how CoCO loss is applied on DSLP are provided in Appendix D. The results are shown in Table 3. Compared to DSLP with CTC loss (Huang et al., 2021), DSLP with CoCO loss consistently improves the BLEU scores on three language pairs, including EN-RU, EN-FI, and EN-DE. On the contrary, DSLP with CTC loss is better or comparable to DSLP with CoCO loss when the target language is restricted to the "SVO" word order, including EN-RO and DE-EN.

According to the experiments on language pairs with different kinds of requirements on word order, we suggest to: 1) use CoCO loss when the word order of the target language is relatively flexible (e.g., RU and FI, where word order on "S" "V" "O" is free, and DE, where the verb is required to be placed on the second position); 2) use CTC loss when the target language is with relatively strict word order (e.g., RO and EN, which are "SVO" languages).

6 Conclusion

In this paper, we conduct a systematic study on the syntactic multi-modality problem in non-autoregressive machine translation. We first categorize this problem into long-range and short-range types and study the effectiveness of different loss functions on each type. Considering the different types are usually entangled in real-world datasets, we design and construct synthesized datasets to control the degree of one type of multi-modality without changing another for analyses. We find that CTC loss is good at short-range syntactic multi-modality while OAXE loss is better at the long-range one. These findings are further verified on real-world datasets with our designed analytical approach. Based on these analyses, we propose a CoCO loss that can better handle the complicated syntactic multi-modality in real-world datasets, and a practical guide to use different loss functions for different kinds of syntactic multi-modality: CoCO loss is preferred when the word order of target language is relatively flexible while CTC loss is preferred when target language is with strict word order. Our study in this paper can facilitate better understanding of the multi-modality problem and provide insights to better solve this problem in non-autoregressive translation. Besides, there still remain some open problems that need future investigation. For example, we generally consider long-range and short-range types for syntactic multi-modality, while there may be more fine-gained categorizations on the syntactic multi-modality due to the complex syntax of natural language.

References

Srinivas Bangalore and Giuseppe Riccardi. 2001. A finite-state approach to machine translation. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2001. ASRU'01.*, pages 381–388. IEEE.

Noam Chomsky. 1959. On certain formal properties of grammars. *Information and control*, 2(2):137–167.

Bernard Comrie. 1989. *Language universals and linguistic typology: Syntax and morphology*. University of Chicago press.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F Wong, Dacheng Tao, and Zhaopeng Tu. 2020. Understanding and improving lexical choice in non-autoregressive translation. *arXiv preprint arXiv:2012.14583*.

Cunxiao Du, Zhaopeng Tu, and Jing Jiang. 2021. Order-agnostic cross entropy for non-autoregressive machine translation. *arXiv preprint arXiv:2106.05093*.

Marjan Ghazvininejad, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy. 2020. Aligned cross entropy for non-autoregressive machine translation. In *International Conference on Machine Learning*, pages 3515–3523. PMLR.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. *arXiv preprint arXiv:1904.09324*.

Talmy Givón. 1983. Topic continuity in spoken english. *Topic continuity in discourse: A quantitative cross-language study*, 3:347–363.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 273–278. IEEE.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Jiatao Gu and Xiang Kong. 2020. Fully non-autoregressive neural machine translation: Tricks of the trade. *arXiv preprint arXiv:2012.15833*.

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 arXiv:2110.07515*.

Elena Dmitrievna Kallestinova. 2007. *Aspects of word order in Russian*. The University of Iowa.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Jindřich Libovickỳ and Jindřich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. *arXiv preprint arXiv:1811.04719*.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank.

Terence Odlin. 2008. A handbook of varieties of english. *Language*, 84(1):193–196.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965. PMLR.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2020. Glancing transformer for non-autoregressive neural machine translation. *arXiv preprint arXiv:2008.07905*.

Peter S Rosenbaum. 1967. Phrase structure principles of english complex sentence formation. *Journal of Linguistics*, 3(1):103–118.

Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. 2020. Non-autoregressive machine translation with latent alignments. *arXiv preprint arXiv:2004.07437*.

Chenze Shao, Jinchao Zhang, Yang Feng, Fandong Meng, and Jie Zhou. 2020. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 198–205.

Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011. Post-ordering in statistical machine translation. In *Proc. MT Summit*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Chunting Zhou, Graham Neubig, and Jiatao Gu. 2019. Understanding knowledge distillation in non-autoregressive machine translation. *arXiv preprint arXiv:1911.02727*.

Appendix

A Number Ranges to Synthesis the Source and Target Sentences

We use [1, 5000], [5001, 10000], [10001, 12500], [12501, 15000], and 15001, 15002, 15003 to represent “N” “V” “JJ” “RB” “DT” in the source sentences, and [15004, 20003], [20004, 25003], [25004, 27503], [27504, 30003], and 30004, 30005, 30006 to represent “N” “V” “JJ” “RB” “DT” in the target sentences.

B Modified OAXE Loss

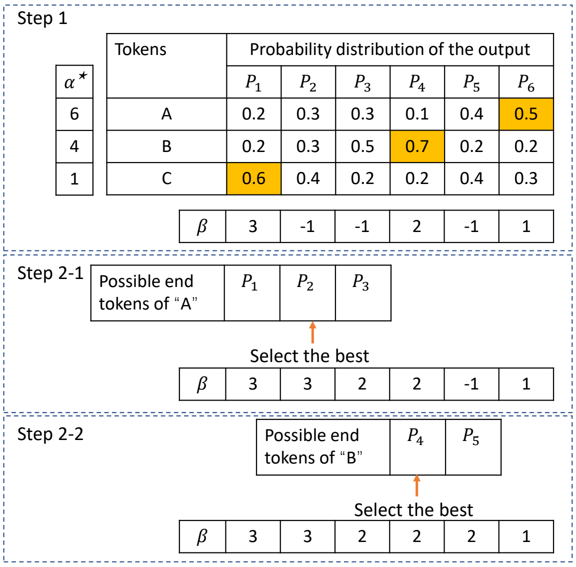
Specifically, we consider one training pair (X,Y), where there are n tokens in the ground truth sequence, denoted as $Y = (y_1, y_2, \ldots, y_n)$. The corresponding output sequence has m tokens with probability distributions $P_1, P_2, \ldots, P_m$, where $m > n$. According to OAXE, we first get the alignment between the ground truth sequence and the output sequence that minimizes the cross entropy loss based on maximum bipartite matching algorithm (Kuhn, 1955):

$$\alpha^\star = \arg\min_{\alpha \in \gamma(\alpha)} \left(- \sum_i \log P_{\alpha(i)}(y_i|X,\theta) \right), \quad (2)$$

where α denotes the alignment from the ground truth sequence to the output sequence, $\gamma(\alpha)$ is the set of all possible alignments, and y_i is aligned with the $\alpha(i)$-th token of the output. We consider each output token can only be aligned to one ground truth token (i.e., $\alpha(i) \neq \alpha(j)$ if $i \neq j$). Then, we can get the alignment from the output sequence to the ground truth sequence, based on $\alpha^\star$:

$$\beta(k) = \begin{cases} i & \text{if } \alpha^\star(i) = k, \\ -1 & \text{if } \forall i \in [1,n], \alpha^\star(i) \neq k, \end{cases} \quad (3)$$

where the k-th token of the output is aligned to $y_{\beta(k)}$ and $\beta(k) = -1$ denotes the token has not been aligned. We provide an illustration as the “step 1” in Fig. 6, where we consider 3 tokens in the target sequence and 6 tokens in the output and the best alignment is “A”-“P_6”, “B”-“P_4”, and “C”-“P_1”. Since consecutive repetitive tokens are merged when decoding with CTC loss, we consider that consecutive tokens in the output can be aligned to the same ground truth token. Accordingly, we enumerate the end of each ground truth token in the output sequence respectively, and select the one that minimize the cross entropy loss. For example, given $\beta(k_1) = i$, $\beta(k_2) = j$ and $\beta(k) = -1$ when $k_1 \leq k \leq k_2$, we select $k^\star$ according to:


Step 1

$\alpha^\star$	Tokens	P_1	P_2	P_3	P_4	P_5	P_6
		Probability distribution of the output					
6	A	0.2	0.3	0.3	0.1	0.4	0.5
4	B	0.2	0.3	0.5	0.7	0.2	0.2
1	C	0.6	0.4	0.2	0.2	0.4	0.3

β	3	-1	-1	2	-1	1

Step 2-1

Possible end tokens of “A”	P_1	P_2	P_3

Select the best

β	3	3	2	2	-1	1

Step 2-2

Possible end tokens of “B”	P_4	P_5

Select the best

β	3	3	2	2	2	1

Figure 6: An illustration of the modified OAXE loss.

$$k^\star = \arg\min_{k_1 \leq k' < k_2} \left(- \sum_{k_1 \leq k \leq k'} \log P_k(y_i|X,\theta) - \sum_{k' < k \leq k_2} \log P_k(y_j|X,\theta) \right), \quad (4)$$

and align the $(k_1, k^\star]$-th output token to i and the $(k^\star, k_2)$-th output token to j as:

$$\beta(k) = \begin{cases} i & \text{if } k \in (k_1, k^\star] \\ j & \text{if } k \in (k^\star, k_2). \end{cases} \quad (5)$$

As the illustration in Fig. 6, we enumerate all the possible end tokens of ’A’ and ’B’ to find the best one. Then, we get the modified OAXE loss as:

$$\mathcal{L}_{M-OAXE} = - \sum_{1 \leq k \leq m} \log P_k \left(y_{\beta(k)} | X, \theta \right). \quad (6)$$

C BLEU Scores of Different Losses on Different Language Pairs.

The BLEU scores of models with CTC, OAXE and CoCO loss on different languages pairs are shown in Table 4.

D Use CoCO Loss in DSLP

Partially feeding ground truth tokens to the decoder during training shows promising performance in

Table 4: BLEU scores of models with different losses on different language pairs.

Loss	EN-RU	EN-FI	EN-DE	DE-EN	RO-EN	EN-RO
CTC	20.84	22.86	26.10	30.36	33.68	33.06
OAXE	21.23	23.13	26.16	30.07	33.25	32.31
CoCO	21.45	23.27	26.25	30.19	33.31	32.67

NAT (Ghazvininejad et al., 2019; Saharia et al., 2020; Qian et al., 2020; Huang et al., 2021). For the models training with golden length of the ground truth sentence using XE loss, the ground truth token embedding is placed to the position of the corresponding input (Qian et al., 2020). When using CTC loss, the inputs of the decoder are always longer than the ground truth sentences, where Gu and Kong (2020) proposes to use the best monotonic alignment between the ground truth and output sequences, and provides the ground truth to the corresponding input position of the decoder. With the proposed CoCO loss, we use the best alignment which is not required to be monotonous. In addition, DSLP requires deep supervision on each layer of the decoder. We find that only replacing CTC loss with CoCO loss on the first layer is better than using CoCO loss on all layers. Accordingly, when using CoCO loss in DSLP transformer, we use CoCO loss in the first layer and CTC loss for all the other layers in the DSLP transformer.