MR-P: A Parallel Decoding Algorithm for Iterative Refinement Non-Autoregressive Translation

Hao Cheng
Academy for Advanced Interdisciplinary Studies, Peking University
hao.cheng@pku.edu.cn

Zhihua Zhang
School of Mathematical Sciences, Peking University
zhzhang@math.pku.edu.cn

Abstract

Non-autoregressive translation (NAT) predicts all the target tokens in parallel and significantly speeds up the inference process. The Conditional Masked Language Model (CMLM) is a strong baseline of NAT. It decodes with the Mask-Predict algorithm which iteratively refines the output. Most works about CMLM focus on the model structure and the training objective. However, the decoding algorithm is equally important. We propose a simple, effective, and easy-to-implement decoding algorithm that we call MaskRepeat-Predict (MR-P). The MR-P algorithm gives higher priority to consecutive repeated tokens when selecting tokens to mask for the next iteration and stops the iteration after target tokens converge. We conduct extensive experiments on six translation directions with varying data sizes. The results show that MR-P significantly improves the performance with the same model parameters. Specifically, we achieve a BLEU increase of 1.39 points in the WMT'14 En-De translation task. Our code is available at https://github.com/chynphh/MaskRepeat-Predict.

1 Introduction

The autoregressive neural machine translation (AT) model based on encoder-decoder framework (Sutskever et al., 2014) has achieved great success (Bahdanau et al., 2015; Wu et al., 2016; Vaswani et al., 2017). The decoder predicts target tokens step by step conditioned on source tokens and previously predicted tokens. Such dependency between target tokens inevitably leads to the decoding latency. Non-autoregressive neural machine translation (NAT) models (Gu et al., 2018; Ghazvininejad et al., 2019) remove the dependency between tokens in the target sentence and generate all tokens in parallel, significantly improving the inference speed.

The assumption of conditional independence in target tokens makes it more difficult for NAT models to learn the target distribution. NAT models' translation is often incomplete or repetitive, especially for long sentences. An approach for alleviating this problem is to iteratively refine the model output and make a trade-off between inference speed and model performance (Lee et al., 2018; Ghazvininejad et al., 2019; Kasai et al., 2020). Many refinement-based models are based on CMLM (Ghazvininejad et al., 2019) and use the Mask-Predict (M-P) (Ghazvininejad et al., 2019) algorithm for decoding. Most works attempt to improve the model from the model structure and the training method.

In this work, we propose a novel decoding algorithm for refinement-based models that we call MaskRepeat-Predict (MR-P). Our algorithm prefers the consecutive repeated tokens when selecting tokens to mask. And the iteration will stop in advance when the target sentence converges, which reduces the number of iterations and avoid over-refinement.We verify the effectiveness of MR-P in six translation directions of three standard datasets with varying data sizes. Under the same model parameters, the model's performance is significantly improved using the MR-P decoding algorithm.

The main contributions of this work are as follows:

- We devise a new decoding algorithm that is simple, effective, and easy-to-implement. The algorithm can achieve a consistent improvement and a lower perplexity on the six translation tasks.
- The algorithm can reduce the average iteration numbers and accelerate the overall translation speed when using a large maximum number of iterations.
- The algorithm is model-agnostic and can be used as long as the conditional masked language model is used for training.

Findings of the Association for Computational Linguistics: ACL 2022, pages 285 - 296
May 22-27, 2022

Iteration	1	2	3	4	10
Short	2.23	0.72	0.35	0.23	0.06
Long	11.83	4.33	1.84	1.11	0.27
All	6.59	2.36	1.03	0.63	0.15

Table 1: The average number of consecutive repeated tokens per sentence with different iterations on the WMT14' De-En test set. We divide all samples into Short and Long according to whether the sentence length is less than 25.

2 Methodology

The Mask-Predict algorithm selects tokens according to the generation probabilities. There is a problem with this strategy. When the probabilities of consecutive repeated tokens are high, they will not be selected and remain in the results.

As can be seen from Table 1, there are many consecutive repeated tokens in the results of the Mask-Predict algorithm, especially in long sentences. So it is necessary to mask the consecutive repeated tokens and re-predict them. Consecutive repeated short phrases occur infrequently, so only consecutive repeated tokens are considered.

2.1 MaskRepeat-Predict

We introduce the **MaskRepeat-Predict** algorithm, a convenient and effective decoding algorithm based on Mask-Predict. In each iteration, the algorithm preferentially selects consecutive repeated tokens, retains the token with the highest confidence among them, and masks the other tokens. Secondly, the lower confidence tokens are selected to mask from other positions. It should be noted that if the target sentence converges, the iteration will be stopped early.

Formal Description The algorithm runs T iterations at most. Let $\mathbf{y}^t = \{y_1^t, ..., y_{M_y}^t\}$ represent the tokens generated in the iteration t, M_y denote the length of the target sentence, and the probability of each token correspond to $\mathbf{p}^t = \{p_1^t, ..., p_{M_y}^t\}$. Let $\mathbf{y}_k^t = \{y_{k_i}^t, i = 1, ..., M_{y_k}\}$ and $\mathbf{p}_k^t = \{p_{k_i}^t, i = 1, ..., M_{y_k}\}$ indicate the k-th group of consecutive repeated tokens and corresponding probabilities generated in the iteration t, which means that positions k_i and k_{i+1} should be actually consecutive and all the tokens in $\mathbf{y}_k^t$ are the same. M_{y_k} means the length of the k-th group of consecutive repeated tokens. $n_t = M_y \cdot \frac{T-(t-1)}{T}$ denotes the number of masked tokens in the t-th iteration.

MaskRepeat For the first iteration, we mask all the tokens. For later iterations, we mask consecutive repeated tokens firstly. For each set of consecutive repeated tokens, we reserve the token $y_{k_i}^{t-1}$ with the highest probability. All the reserved tokens constitute $\mathbf{y}_{top_r}^t$:

$$\mathbf{y}_{top_r}^t = \bigcup_k^K \left\{ y_{k_i}^{t-1} \mid k_i = \arg\max_i \left\{ p_{k_i}^{t-1} \right\} \right\}, \quad (1)$$

where K denotes the number of consecutive repeated tokens groups. All other repeated tokens $\mathbf{y}_{mask_r}^t$ are masked:

$$\mathbf{y}_{mask_r}^t = \bigcup_k^K \{\mathbf{y}_k^{t-1}\} \setminus \mathbf{y}_{top_r}^t, \quad (2)$$

Next, we mask the tokens with lower probabilities in the whole sentence:

$$\mathbf{y}_{mask_p}^t = \{y_i^{t-1} \mid p_i^t \in \text{topk}(-\mathbf{p}^t, k = m), i\}, \quad (3)$$

where $m = \max\{n_t - |\mathbf{y}_{mask_r}^t|, 0\}$. Then we have

$$\mathbf{y}_{mask}^t = \mathbf{y}_{mask_p}^t \cup \mathbf{y}_{mask_r}^t, \quad (4)$$
$$\mathbf{y}_{obs}^t = \mathbf{y}^{t-1} \setminus \mathbf{y}_{mask}^t. \quad (5)$$

Predict The prediction process is the same as Mask-Predict. The model predicts the masked tokens $\mathbf{y}_{mask}^t$ conditioned on the source tokens $\mathbf{x}$ and the observed tokens $\mathbf{y}_{obs}^t$. The token with the highest probability at each masked position is selected to update prediction tokens, and the probabilities are updated accordingly. For $y_i^{t-1} \in \mathbf{y}_{mask}^t$,

$$y_i^t = \arg\max_w P\left(y_i = w \mid \mathbf{x}, \mathbf{y}_{obs}^t\right),$$
$$p_i^t = \max_w P\left(y_i = w \mid \mathbf{x}, \mathbf{y}_{obs}^t\right).$$

Unmasked positions retain the same probability value as the previous iteration. For $y_i^{t-1} \in \mathbf{y}_{obs}^t$,

$$y_i^t = y_i^{t-1},$$
$$p_i^t = p_i^{t-1}.$$

Early Stop The iteration will be stopped early if the target sentence converges:

$$\mathbf{y}^t = \mathbf{y}^{t-1}.$$

In particular, we set $\mathbf{y}_{obs}^0 = \{\texttt{Mask}, ..., \texttt{Mask}\}$ to predict $\mathbf{y}^0$. We use the Mask-Predict algorithm when $t < \lfloor T/2 \rfloor$. See Alg. 1 in Appendix A for a full pseudo-code.

	source	Eine stand@@ sichere Mauer ist Voraussetzung für einen von Sch@@ ül@@ ern benutzen Schul@@ hof , was durch die aktuellen Bef@@ es@@ tigungs@@ arbeiten erfolgt ist .													
iter=0	M-P / MR-P	A	**stur@@**	**wall**	wall	wall	**is**	**prerequisite**	**for**	**for**	school	school	school	school	**school**
		0.875	0.144	0.591	0.652	0.817	0.391	0.451	0.343	0.408	0.815	0.811	0.645	0.681	0.511
		students	**which**	**has**	**been**	**done**	**by**	**the**	**the**	**fast@@**	**forti@@**	**work**	.		
		0.307	0.421	0.435	0.284	0.521	0.218	0.554	0.467	0.456	0.177	0.538	0.902		
iter=1	M-P	A	**shel@@**	**proof**	wall	wall	is	**prerequisite**	for	a	school	school	school	school	**,**
		0.875	0.231	0.457	0.652	0.817	0.866	0.391	0.733	0.672	0.815	0.811	0.645	0.681	0.228
		,	**which**	has	**been**	done	**by**	the	current	fast@@	**forti@@**	work	.		
		0.316	0.327	0.470	0.377	0.492	0.303	0.737	0.615	0.520	0.151	0.654	0.902		
	MR-P	A	**shel@@**	proof	**wall**	wall	is	**prerequisite**	for	a	school	**school**	**school**	**school**	**,**
		0.875	0.231	0.457	0.652	0.817	0.866	0.391	0.733	0.672	0.815	0.811	0.645	0.681	0.228
		,	which	has	been	done	by	the	current	fast@@	**forti@@**	work	.		
		0.316	0.327	0.470	0.377	0.492	0.303	0.737	0.615	0.520	0.151	0.654	0.902		
iter=2	M-P	A	stand@@	proof	wall	wall	is	required	for	a	school	school	school	school	students
		,	which	has	been	done	through	the	current	fast@@	ening	work	.		
	MR-P	A	stand@@	proof	proof	wall	is	required	for	a	school	yard	used	by	students
		,	which	has	been	done	by	the	current	fast@@	forti@@	work	.		

Figure 1: An example from the WMT'14 De-En test set illustrates how MaskRepeat-Predict (MR-P) and Mask-Predict (M-P) generate text with three iterations. The numbers below tokens represent their probabilities. The highlighted tokens are masked for the next iteration and re-predicted.

Example Figure 1 shows an example from the WMT'14 De-En test set when CMLM uses Mask-Predict and MaskRepeat-Predict to decode with three iterations. At the end of the second iteration (iter $= 1$), Mask-Predict selects nine tokens with lower confidence to mask. It can be seen that there are four consecutive `schools` with higher probabilities in the result, so they are not masked and re-predicted. However, these words should be chosen for re-prediction, regardless of their probability. The MaskRepeat-Predict algorithm starts to mask the consecutive repeated tokens in the middle of iterations. As we can see, in the second iteration, the repeated tokens `school` and `wall` that have low probabilities are masked instead of other unique tokens with lower probabilities. The result at the end of iterations also has higher quality.

For consecutive repeated tokens and corresponding probabilities, we take the sentence of the second iteration (iter $= 1$) in Figure 1 as an example:

$$\mathbf{y}_1^1 = \{\texttt{wall}, \texttt{wall}\},$$
$$\mathbf{p}_1^1 = \{0.652, 0.817\};$$
$$\mathbf{y}_2^1 = \{\texttt{school}, \texttt{school}, \texttt{school}, \texttt{school}\},$$
$$\mathbf{p}_2^1 = \{0.815, 0.811, 0.645, 0.681\}.$$

3 Experiments

3.1 Experimental Settings

We evaluate our algorithms on six directions from three standard datasets with various training data sizes: WMT'16 En-Ro (610K pairs), WMT'14 En-De (4.5M pairs), WMT'17 En-Zh (20M pairs). For a fair comparison, we used the distillation data provided by Kasai et al. (2020), and all data processing methods and hyperparameters settings are the same. Please see Appendix C for details. Our code is based on `CMLM`[1] and `DisCo`[2].

3.2 Overall Results

Table 2 shows the results on WMT'14 En-De and WMT'16 En-Ro test sets with CMLM and DisCo. We use pre-trained DisCo models provided by original authors (Kasai et al., 2020) for testing the decoding algorithm. CMLM models are implemented by ourselves. It can be seen that the results with MR-P have a different range of improvements compared to the ones with M-P for different iterations. The fewer iterations, the more obvious the pronounced performance improvement. Especially when only iterating two steps, the result on the WMT'14 En-De test set is improved by 1.39 BLEU points. Even with the ten iterations, there is an improvement of 0.39 BLEU on the WMT'16 Ro-En test set. It is worth noting that this is only a change in the decoding algorithm, no changes have been made to the model, and even the decoding algorithm parameters are the same.

Table 3 shows the results with CMLM on the WMT'17 En-Zh test set. Pre-trained models are provided by original authors (Ghazvininejad et al., 2019). There is a gain of 1.26 BLEU improvement

[1]https://github.com/facebookresearch/Mask-Predict
[2]https://github.com/facebookresearch/DisCo

Models	MaxIter.	En-De	De-En	En-Ro	Ro-En
CMLM +M-P	2	23.97	28.62	32.15	32.11
	3	25.99	30.15	32.75	33.14
	4	26.58	30.62	32.99	33.42
	10	27.26	31.07	33.44	33.79
CMLM +MR-P	2	25.10(+1.13)	29.41(+0.79)	32.45(+0.30)	32.88(+0.77)
	3	26.43(+0.44)	30.46(+0.31)	33.17(+0.42)	33.55(+0.41)
	4	26.78(+0.20)	30.73(+0.11)	33.25(+0.26)	33.80(+0.38)
	10	27.42(+0.16)	31.34(+0.27)	33.41(-0.03)	34.16(+0.37)
DisCo +M-P	2	23.02	28.28	32.05	32.49
	3	25.31	29.72	32.41	32.80
	4	25.83	30.15	32.63	32.92
	10	27.06	30.89	32.92	33.12
DisCo +MR-P	2	24.41(+1.39)	29.24(+0.96)	32.33(+0.28)	33.01(+0.52)
	3	25.48(+0.17)	29.99(+0.27)	32.56(+0.15)	32.98(+0.18)
	4	25.96(+0.13)	30.47(+0.32)	32.81(+0.18)	33.20(+0.28)
	10	27.11(+0.05)	30.91(+0.02)	33.15(+0.23)	33.33(+0.21)

Table 2: The performance (BLEU) of CMLM and DisCo with MaskRepeat-Predict (MR-P), compared to that with Mask-Predict (M-P).

Alg.	MaxIter.	En-Zh	Zh-En
M-P	2	30.50	18.79
	3	32.03	21.46
	4	32.63	21.90
MR-P	2	31.41(+0.91)	19.96(+1.26)
	3	32.34(+0.31)	21.76(+0.30)
	4	32.82(+0.19)	22.19(+0.29)

Table 3: The performance (BLEU) of CMLM with MaskRepeat-Predict(MR-P) on WMT'17 En-Zh, compared to that with Mask-Predict(M-P).

MaxIter.	En-De	De-En	En-Ro	Ro-En
4	3.66	3.55	3.40	3.41
10	5.97	5.22	4.58	4.57

Table 4: The average iteration numbers of CMLM decoding with MR-P.

Alg.	De-En	Ro-En	Zh-En
Ground Truth	166.3	223.1	142.1
M-P	407.7	491.2	198.2
MR-P	322.2	459.8	187.7

Table 5: The perplexity of CMLM decoding with a maximum of ten iterations.

over M-P on Zh-En with two iterations.

Tables 9 in Appendix show more details for CMLM, DisCo, and CCAN (Ding et al., 2020).

3.3 Analysis

Iteration Numbers The MR-P algorithm will stop the iteration when the target sentence converges, so sometimes it will not reach the maximum number of iterations. As shown in Table 4, we can see that the average number of iterations is significantly reduced when the maximum number of iterations is relatively large.

Perplexity We make a more in-depth comparison from the Perplexity(PPL). We use pre-trained GPT-2 (Radford et al., 2019) provided by Hugging-Face (Wolf et al., 2020) as our language model. As we can see in Table 5, the perplexity is significantly reduced when using MR-P instead of M-P, which means that sentences generated using MR-P are more reasonable.

Remove Duplicates The problem of repeated translation can also be alleviated simply by removing all consecutive duplicated tokens in translation results. Table 6 shows the BLEU of CMLM on the WMT'14 En-De test set. Remove Duplicates(RD) can improve performance, but is not as good as using MR-P. A possible reason is that MR-P can

MaxIter.	2	3	4	10
M-P	23.97	25.99	26.58	27.26
+RD	24.53	26.29	26.77	27.30
MR-P	25.10	26.43	26.78	27.42
+RD	25.34	26.62	26.84	27.41

Table 6: The performance of whether uses RD or not.

MaxIter.	2	3	4	10
Short	0.35	0.19	0.11	0.03
Long	1.45	0.91	0.44	0.11
All	0.85	0.52	0.26	0.07

Table 7: The average number of consecutive repeated tokens per sentence on WMT'14 De-En test of MR-P.

affect the generation process, while RD cannot. It is worth noting that RD can also improve the performance of MR-P when the maximum number of iterations is relatively small.

Consecutive Repeated Translation We compute the average number of consecutive repeated tokens per sentence on the WMT14' De-En test set. The result is shown in Table 7 and Table 1. The MR-P algorithm benefits from its inherent principle and can significantly reduce the repetition rate. Especially when iterating only two steps, the repetition rate is reduced from 2.36 to 0.85.

Different Source Lengths We split the source sentences into different length buckets to analyze the effect of source input length. Figure 2 shows the BLEU of CMLM with two iterations at most on the WMT'14 En-De test set. The longer the source sentences are, the more considerable the margin between MR-P and M-P is.

4 Conclusion

In this paper, we have proposed the MR-P decoding algorithm. MR-P prefers to mask consecutive repeated tokens and stops the iteration early when target tokens converge. The experiments on different models and datasets have shown that MR-P is effective and model-agnostic. The algorithm can achieve a consistent improvement and a lower perplexity on the six translation tasks.

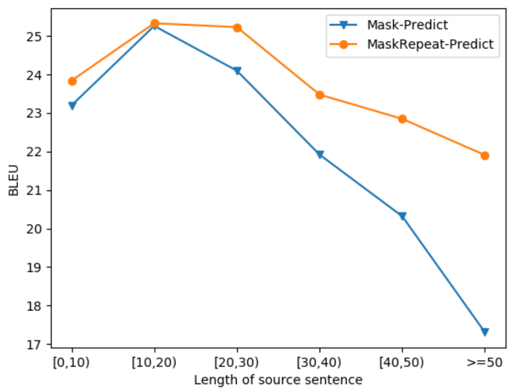

Figure 2: The BLEU points on the test set of WMT'14 En-De over sentences in different length buckets.

Acknowledgments

This work is supported by National Key R&D Program of China(No. 2020AAA0104400).

References

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

Yu Bao, Hao Zhou, Jiangtao Feng, Mingxuan Wang, Shujian Huang, Jiajun Chen, and Lei Li. 2019. Non-autoregressive transformer by position learning. *arXiv preprint arXiv:1911.10677*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

PK Diederik and B Jimmy. 2014. Adam: A method for stochastic optimization. iclr. *arXiv preprint arXiv:1412.6980*.

Liang Ding, Longyue Wang, Di Wu, Dacheng Tao, and Zhaopeng Tu. 2020. Context-aware cross-attention for non-autoregressive translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4396–4402, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.

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

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

Junliang Guo, Xu Tan, Di He, Tao Qin, Linli Xu, and Tie-Yan Liu. 2019. Non-autoregressive neural machine translation with enhanced decoder input. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3723–3730. AAAI Press.

Junliang Guo, Xu Tan, Linli Xu, Tao Qin, Enhong Chen, and Tie-Yan Liu. 2020a. 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.

Junliang Guo, Linli Xu, and Enhong Chen. 2020b. Jointly masked sequence-to-sequence model for non-autoregressive neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 376–385, Online. Association for Computational Linguistics.

Junliang Guo, Zhirui Zhang, Linli Xu, Hao-Ran Wei, Boxing Chen, and Enhong Chen. 2020c. Incorporating BERT into parallel sequence decoding with adapters. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. 2020. Non-autoregressive machine translation with disentangled context transformer. 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 5144–5155. PMLR.

Xiang Kong, Zhisong Zhang, and Eduard Hovy. 2020. Incorporating a local translation mechanism into non-autoregressive translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1067–1073, Online. Association for Computational Linguistics.

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

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

Zhuohan Li, Zi Lin, Di He, Fei Tian, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Hint-based training for non-autoregressive machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5708–5713, Hong Kong, China. Association for Computational Linguistics.

Jindřich Libovický and Jindřich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3016–3021, Brussels, Belgium. Association for Computational Linguistics.

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.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Dario Amodei, Daniela Amodei, Jack Clark, Miles Brundage, and Ilya Sutskever. 2019. Better language models and their implications. *OpenAI Blog https://openai.com/blog/better-language-models*.

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

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

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

Raphael Shu, Jason Lee, Hideki Nakayama, and Kyunghyun Cho. 2020. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. In *AAAI*, pages 8846–8853.

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.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

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

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

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. 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, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Jiawei Zhou and Phillip Keung. 2020. Improving non-autoregressive neural machine translation with monolingual data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1893–1898, Online. Association for Computational Linguistics.

A Algorithm

Algorithm 1: MaskRepeat-Predict

Input: Source sentence: $\mathbf{x}$
Predict target length: M_y;
Compute $\mathbf{y}^0$ use $\mathbf{y}^0_{obs}$;
for $t \in 1, ..., T-1$ **do**
  if $t < \lfloor T/2 \rfloor$ **then**
    set $\mathbf{y}^t_{mask_r} = \emptyset$;
    compute $\mathbf{y}^t_{mask_p}$ by (3);
    compute $\mathbf{y}^t_{mask}$ by (4);
  else
    compute $\mathbf{y}^t_{top_r}$ by (1);
    compute $\mathbf{y}^t_{mask_r}$ by (2);
    compute $\mathbf{y}^t_{mask_p}$ by (3);
    compute $\mathbf{y}^t_{mask}$ by (4);
  end
  compute $\mathbf{y}^t_{obs}$ by (5);
  predict $\mathbf{y}^t$;
  if $\mathbf{y}^t = \mathbf{y}^{t-1}$ **then**
    return $\mathbf{y}^t$;
  end
end
return $\mathbf{y}^{T-1}$

B Examples

Figure 3 shows an additional example from the WMT'14 De-En test set of CMLM with different decoding algorithm.

	source	„ Der Bo@@ ard@@ ing-@@ Prozess gehörte zu den reibungs@@ los@@ esten , die ich in meiner Lauf@@ bahn in der Luft@@ fahrt erlebt habe “ , sagte er .											
iter=0	M-P / MR-P	“	The	boarding	process	process	was	one	the	most	smooth	I	I
		0.916	0.909	0.357	0.486	0.763	0.691	0.516	0.614	0.496	0.412	0.579	0.830
		experienced	experienced	my	career	aviation	aviation	,	”	he	said	.	
		0.699	0.665	0.506	0.505	0.498	0.491	0.893	0.920	0.921	0.944	0.903	
iter=1	M-P	“	The	boarding	process	process	was	one	the	most	smooth	I	I
		0.916	0.909	0.374	0.589	0.721	0.661	0.481	0.637	0.575	0.464	0.691	0.782
		experienced	experienced	in	my	aviation	aviation	,	”	he	said	.	
		0.696	0.774	0.437	0.448	0.481	0.492	0.893	0.920	0.921	0.944	0.903	
	MR-P	“	The	boarding	process	process	was	one	the	most	smooth	I	I
		0.916	0.909	0.374	0.589	0.721	0.661	0.481	0.637	0.575	0.464	0.691	0.782
		experienced	experienced	in	my	aviation	aviation	,	”	he	said	.	
		0.696	0.774	0.437	0.448	0.481	0.492	0.893	0.920	0.921	0.944	0.903	
iter=2	M-P	“	The	boarding	process	process	was	among	the	most	smooth	I	I
		experienced	experienced	in	my	aviation	career	,	”	he	said	.	
	MR-P	“	The	bo@@	arding	process	was	among	the	most	smooth	one	I
		have	experienced	in	my	career	aviation	,	”	he	said	.	

Figure 3: An example from the WMT’14 De-En test set illustrates how MaskRepeat-Predict (MR-P) and Mask-Predict (M-P) generate text with three iterations.

C Experimental Settings

Datasets We evaluate our inference algorithms on six directions from three standard datasets with various training data sizes: WMT'16 En-Ro (610K pairs), WMT'14 En-De (4.5M pairs), WMT'17 En-Zh (20M pairs). All datasets are tokenized into subword units by BPE (Sennrich et al., 2016). Specially, use joint BPE on WMT'16 En-Ro and WMT'14 En-De. We use the same preprocessed data as Kasai et al. (2020) for a fair comparions with other models (WMT'16 En-Ro: Lee et al. (2018); WMT'14 En-De: Vaswani et al. (2017)). We evaluate performance with BLEU (Papineni et al., 2002) for all language pairs except that using SacreBLEU (Post, 2018)[3] for pair from En to Zh.

Hyperparameters We follow the hyperparameters for a transformer base (Vaswani et al., 2017; Ghazvininejad et al., 2019; Kasai et al., 2020): 6 layers for the encoder and the decoder, 8 attention heads, 512 model dimensions, and 2048 hidden dimensions per layer. We sample weights from $\mathcal{N}(0, 0.02)$, initialize biases to zero and set layer normalization parameters to $\beta = 0$, $\gamma = 1$, following the weight initialization scheme from BERT (Devlin et al., 2019). Set dropout rate to 0.3, and apply weight decay with 0.01 and label smoothing with $\epsilon = 0.1$ for regularization. We train batches of approximately $16K \cdot 8$ (8 GPUs with 16K per GPU) tokens using Adam (Diederik and Jimmy, 2014) with $\beta = (0.9, 0.999)$ and $\epsilon = 10^{-6}$. The learning rate warms up to $5 \cdot 10^{-4}$ for the first 10K steps, and the decays with the inverse square-root schedule. We train models for 300K steps with mixed precision floating point arithmetic (Micikevicius et al., 2018) on 8 TITAN RTX GPUs, and average the 5 best checkpoints as the final model. Following the previous works (Ghazvininejad et al., 2019; Kasai et al., 2020), we apply length beam with the size of 5.

D Experiments

Seen in Table 8 are the results of strong non-autoregressive machine translation models similar with CMLM on the WMT'14 En-De and WMT'16 En-Ro test set. Basic models that use the MaskRepeat-Predict decoding algorithm can achieve comparable results with other advanced models. It is worth noting that the models such as Imputer, LAT, SMART, JM-NAT, and ENGINE all employ the Mask-Predict decoding algorithm, which means that they can also use the MaskRepeat-Predict decoding algorithm.

Table 9 shows the average iteration number (AveIter.) and performance (BLEU) for Self-CMLM, Pre-trained-CMLM, DisCo, and CCAN. Our CMLM results are much better than the results reported in the original paper. The difference in the final BLEU points comes from batch size and averaging checkpoints with 5 top BLEU points on validation. These two techniques come from Kasai et al. (2020). Comparing self-implemented models and pre-trained models, we can conclude that the MaskRepeat-Predict algorithm still works after the model is enhanced.

Models	En-De	De-En	En-Ro	Ro-En
Imputer	28.20	31.80	34.40	34.10
LAT	27.35	32.04	32.87	33.26
SMART	27.65	31.27	-	-
JM-NAT	27.69	32.24	33.52	33.72
ENGINE	-	-	-	34.04
CMLM	27.03	30.53	33.08	33.31
DisCo	27.34	31.31	33.22	33.25
CCAN	27.50	-	-	33.70
+MR-P				
CMLM	27.42	31.34	33.41	34.14
CCAN	27.47	31.36	33.50	33.84

Table 8: The performance of non-autoregressive machine translation methods on the WMT'14 En-De and WMT'16 En-Ro test set.

[3] SacreBLEU hash: BLEU+case.mixed+lang.en-zh+numrefs.1+smooth.exp+test.wmt17+tok.zh+version.1.3.7.

	En-De		De-En		En-Ro		Ro-En	
Models	AveIter.	BLEU	AveIter.	BLEU	AveIter.	BLEU	AveIter.	BLEU
Pre-trained-CMLM +MP	2	22.91	2	27.16	2	31.08	2	31.91
	3	25.00	3	29.11	3	32.19	3	32.93
	4	25.94	4	29.90	4	32.53	4	33.23
	10	27.03	10	30.53	10	33.08	10	33.31
Pre-trained-CMLM +MR-P	2	24.29	2	28.27	2	31.73	2	32.75
	2.92/3	25.50	2.89/3	29.51	2.84/3	32.49	2.82/3	33.33
	3.67/4	26.25	3.61/4	30.13	3.44/4	32.76	3.39/4	33.51
	6.00/10	27.07	5.38/10	30.54	4.83/10	33.14	4.47/10	33.66
DisCo +MP	2	23.02	2	28.28	2	32.05	2	32.49
	3	25.31	3	29.72	3	32.41	3	32.80
	4	25.83	4	30.15	4	32.63	4	32.92
	10	27.06	10	30.89	10	32.92	10	33.12
DisCo +MR-P	2	24.41	2	29.24	2	32.33	2	33.01
	2.92/3	25.48	2.88/3	29.99	2.77/3	32.56	2.74/3	32.98
	3.71/4	25.96	3.59/4	30.47	3.32/4	32.81	3.21/4	33.20
	6.58/10	27.11	5.69/10	30.91	4.23/10	33.15	3.86/10	33.33
Self-CMLM +M-P	2	23.97	2	28.62	2	32.15	2	32.11
	3	25.99	3	30.15	3	32.75	3	33.14
	4	26.58	4	30.62	4	32.99	4	33.42
	10	27.26	10	31.07	10	33.44	10	33.79
Self-CMLM +MR-P	2	25.10	2	29.41	2	32.45	2	32.88
	2.91/3	26.43	2.87/3	30.46	2.83/3	33.17	2.83/3	33.55
	3.66/4	26.78	3.55/4	30.73	3.40/4	33.25	3.41/4	33.80
	5.97/10	27.42	5.22/10	31.34	4.58/10	33.41	4.57/10	34.16
CCAN +M-P	2	23.80	2	28.54	2	31.36	2	32.59
	3	25.88	3	30.02	3	32.32	3	33.15
	4	26.50	4	30.56	4	32.77	4	33.18
	10	27.30	10	31.25	10	33.13	10	33.64
CCAN +MR-P	2	24.86	2	29.05	2	31.97	2	33.05
	2.90/3	26.26	2.87/3	30.25	2.82/3	32.74	2.80/3	33.26
	3.67/4	26.89	3.57/4	30.67	3.42/4	33.07	3.35/4	33.47
	5.97/10	27.47	5.28/10	31.36	4.84/10	33.50	4.43/10	33.84

Table 9: The performance (BLEU) of CMLM, DisCo and CCAN, with MaskRepeat-Predict (MR-P), compared to that with Mask-Predict (M-P). All Pre-trained-CMLM and DisCo models trained by the original authors (Ghazvininejad et al., 2019; Kasai et al., 2020) are used to decode without any change. Self-CMLM and CCAN are implemented by ourselves.

E Ablation Study

Strategies We compare several design strategies of MR-P. MR-P-W: MR-P without early stopping, that is, all the sentence is continually refined until the preset maximum number of iterations. MR-P-A: MR-P is used all the time, including when $t < \lfloor T/2 \rfloor$. MR-P-F: MR-P is used when $t < \lfloor T/2 \rfloor$ and M-P is used when $t \geq \lfloor T/2 \rfloor$. As shown in Table 10, we can see that most of the time, the results of the MR-P algorithm are optimal. There is a slight decline in performance without early stopping. We think this is because some sentences are over-refinement, misleading to the scoring of candidate sentences. Using M-P in the first half of iterations will lay a good foundation for the following iterations.

Expand to other algorithms The Easy-First (E-F) is a decoding algorithm proposed by Kasai et al. (2020) for the DisCo. The condition $\mathbf{y}_{obs}$ of each token is different. Each token can be refined conditioned on all other tokens with a lower probability than itself. The conditional dependence is determined by the probability generated in the first iteration and fixed for the following iterations. We can easily integrate the ideas of MaskRepeat into Easy-First. For repeated tokens that appear continuously, except for the token with the highest probability, the confidence is set to the lowest no matter how high their probability is. This means that consecutive repeated tokens do not become the context of any other token. Then one updates this consecutive repeated tokens part's order in the second iteration. We call that MaskRepeat-Easy-First(MR-E-F). As shown in Table 11, the performance is improved, especially in WMT'14 En-De with 0.16 BLEU points.

		En-De	De-En	En-Ro	Ro-En
MR-P-W	2	25.08	29.37	32.39	32.83
	3	26.30	30.40	33.01	33.37
	4	26.78	30.70	33.18	33.63
	10	27.29	31.06	33.53	33.89
MR-P-A	2	25.10	29.41	32.45	32.88
	3	26.42	30.65	33.08	33.57
	4	26.70	30.54	33.38	33.81
	10	27.28	31.25	33.45	34.01
MR-P-F	2	25.10	29.41	32.45	32.88
	3	26.24	30.61	32.96	33.42
	4	26.73	30.57	33.32	33.76
	10	27.29	31.21	33.49	34.03
MR-P	2	25.10	29.41	32.45	32.88
	3	26.43	30.46	33.17	33.55
	4	26.78	30.73	33.25	33.80
	10	27.42	31.34	33.41	34.16

Table 10: The performance of self-implemented CMLM with different design strategies of MR-P.

Alg.	En-De	De-En	Ro-En	Zh-En
E-F	27.35	31.31	33.24	23.83
MR-E-F	27.51	31.36	33.25	23.97

Table 11: The performance of DisCo (Kasai et al., 2020) decodes with Easy-First (E-F) and MaskRepeat-Easy-First (MR-E-F).

F Related Work

In order to speed up the translation process, Gu et al. (2018) introduced non-autoregressive translation for the first time. A lot of works based on iterative refinement are proposed to make a trade-off between performance and decoding speed (Lee et al., 2018; Ghazvininejad et al., 2019; Kasai et al., 2020; Guo et al., 2020b; Lee et al., 2020; Ghazvininejad et al., 2020b; Ding et al., 2020). Other approaches include improving training objectives (Libovický and Helcl, 2018; Shao et al., 2020; Ghazvininejad et al., 2020a; Saharia et al., 2020), enhancing the decoder input (Guo et al., 2019; Bao et al., 2019; Ran et al., 2019), adding regularization terms on the decoder (Wang et al., 2019; Li et al., 2019), latent variable-based model (Ma et al., 2019; Shu et al., 2020), adding a lite autoregressive module (Sun et al., 2019; Kong et al., 2020), learning or transforming from autoregressive model (Guo et al., 2020a; Sun and Yang, 2020; Tu et al., 2020; Liu et al., 2020), training with monolingual data (Zhou and Keung, 2020), and incorporating the pre-trained model (Guo et al., 2020c).