HI-CMLM: Improve CMLM with Hybrid Decoder Input

Minghan Wang¹, Jiaxin Guo¹, Yuxia Wang², Yimeng Chen¹, Chang Su¹,

Daimeng Wei¹, Min Zhang¹, Shimin Tao¹, Hao Yang¹

¹Huawei Translation Services Center, Beijing, China

²University of Melbourne, Melbourne, Australia

{wangminghan,guojiaxin1,chenyimeng,suchang8,

weidaimeng, zhangmin186, taoshimin, yanghao30}@huawei.com

yuxiaw@student.unimelb.edu.au

Abstract

Mask-predict CMLM (Ghazvininejad et al., 2019) has achieved stunning performance among non-autoregressive NMT models, but we find that the mechanism of predicting all of the target words only depending on the hidden state of [MASK] is not effective and efficient in initial iterations of refinement, resulting in ungrammatical repetitions and slow convergence. In this work, we mitigate this problem by combining copied source with embeddings of [MASK] in decoder. Notably. it's not a straightforward copying that is shown to be useless, but a novel heuristic hybrid strategy - fence-mask. Experimental results show that it gains consistent boosts on both WMT14 En \leftrightarrow De and WMT16 En \leftrightarrow Ro corpus by 0.5 BLEU on average, and 1 BLEU for lessinformative short sentences. This reveals that incorporating additional information by proper strategies is beneficial to improve CMLM, particularly translation quality of short texts and speeding up early-stage convergence.

1 Introduction

In neural machine translation (NMT), autoregressive models decode tokens one-by-one: $p(Y|X) = \prod_{i}^{T} p(y_i|y_{\leq i}|X)$, which ensures the robustness of intrinsic language models but slows down the inference. Non-autoregressive models break the dependency between adjacent tokens: $p(Y|X) = \prod_{i}^{T} p(y_i|X)$, enabling to generate all outputs in parallel.

Recent years have witnessed impressive advances in non-autoregressive models, such as fully-NAT and its variants (Gu et al., 2018; Guo et al., 2019; Wang et al., 2019), insertion-based models (Stern et al., 2019; Gu et al., 2019) and iterative refinement models (Lee et al., 2018; Ghazvininejad et al., 2019). Mask-predict CMLM (CMLM) stands out of them owing to both significantly-fast inference and remarkable performance (Ghazvininejad et al., 2019). It extends the masked language model (Devlin et al., 2019) and enables it to solving generation tasks with iterative refinement. In each step, the model decodes target conditioned on m well-predicted tokens with high confidence and $(L-m) \times [MASK]$, where L is the length of target (see Section 2 for details). This mechanism leads to the issue that it's liable to generate repeated tokens and slow down the convergence in early-stage iterations. We speculate this is because the proportion of useful tokens, i.e. $m \rightarrow 0$, is too small to provide enough information for the next step prediction. Intuitively, the model tends to predict similar or even identical tokens when observing [MASK] only and constantly.

To alleviate this problem, we ameliorate CMLM by incorporating additional information from source embedding into the decoder input (HI-CMLM in short). Experimental results show that it gains consistent boosts on both WMT14 En \leftrightarrow De and WMT16 En \leftrightarrow Ro corpus by 0.5 BLEU on average, and 1 BLEU for less-informative short sentences. This reveals that incorporating additional information by proper strategies is beneficial to improving CMLM, particularly translation quality of short texts and speeding up the convergence of the first four iterations, compared with CMLM.

2 Conditional Masked Language Models

2.1 Model

The architecture of CMLM is a standard encoderdecoder Transformer (Vaswani et al., 2017) without the decoder self-attention mask because the dependency on left tokens has been removed. Formally, given source/target pair (X, Y), the model first predicts the target length based on X before decoding, with objective function:

$$\mathcal{L}_{\text{LEN}} = log P(L|X;\theta). \tag{1}$$

In token prediction at step t, the model refines unobserved tokens $Y_{\text{mask}}^{(t)}$ by minimizing MLM loss:

$$\mathcal{L}_{\text{MLM}} = \sum_{y_i \in Y_{\text{mask}}^{(t)}} log P(y_i | X, Y_{\text{obs}}^{(t)}; \theta). \quad (2)$$

based on a sequence consisting of observed tokens $Y_{obs}^{(t)}$ and masked tokens $Y_{mask}^{(t)}$. The total loss function is the sum of length loss and MLM loss:

$$\mathcal{L} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{LEN}}.$$
 (3)

2.2 Mask-predict Decoding

The decoder runs a *mask* operation, followed by *predict* for *T* iterations. In each iteration *t*, it masks the *k* tokens with the lowest probability scores, where *k* is determined by a linear decay function of *t*: $k = L \times \frac{T-t}{T}$. Observed tokens $Y_{obs}^{(t+1)}$ and masked tokens $Y_{mask}^{(t+1)}$ are updated by:

$$Y_{\text{mask}}^{(t+1)} = \arg\min(p_i^{(t)}, k) \tag{4}$$

$$Y_{\text{obs}}^{(t+1)} = Y^{(t)} \setminus Y_{\text{mask}}^{(t+1)},$$
 (5)

where $p_i^{(t)}$ is the probability score when the model predicts token y_i at step t:

$$y_i^{(t)} = \arg\max P(y_i = w | X, Y_{\text{obs}}^{(t)}) \qquad (6)$$

$$p_i^{(t)} = \max P(y_i = w | X, Y_{\text{obs}}^{(t)}),$$
 (7)

2.3 Training Strategy

To simulate the decoding process in each step, the ground truth target is corrupted by randomly replacing several tokens with [MASK]. The number and the position of the [MASK] follows the uniform distribution so that every token has equal chance to be masked. Then, the model has to recover the corrupted sequence.

2.4 Rethinking Effectiveness of Mask

In *mask-predict*, what should be highlighted is that for the first iteration: $t = 0 \rightarrow k = L$, the model masks all the tokens, thus it predicts entire target sequence merely depending on a full sequence of [MASK] of length L. This leads to the fact that the decoder always requires more than 5 refinement iterations to converge, which is significantly against the original intention to be faster.

We speculate this may result from following reasons: 1) The proportion of $Y_{\rm obs}$ is too small to support the masked language model generating fluency sentences and 2) The representation of $Y_{\rm mask}$ is less informative and distinguishable, and the consecutive [MASK] padding form exacerbates the situation because of lacking useful information inferred from surrounding tokens. We hypothesize that proper initialization of $Y_{obs}^{(t)}(t \leq 3)$ may be beneficial to speeding up refinement, and improving the final performance. But the question is *what initialization would be helpful?* Put differently, *how to incorporate additional information to* Y_{mask} *to ameliorate prediction in initial steps.*

3 Method

In this section, we propose three hybrid approaches to incorporate source embeddings and describe modifications on training strategy accordingly.

3.1 Copy of Source Embedding

The most straightforward method is to mix mask tokens with the source embedding. To address the inconsistency of length, we follow the prior work by uniform copy or soft copy (Gu et al., 2018; Lee et al., 2018; Guo et al., 2019; Wang et al., 2019) but with a modified copy function which copies tokens according to their relative position instead of the absolute position. We denote the copy function as $z_i = \Phi(\mathbf{e})$:

$$d_{ij} = -\left|\frac{i}{L_Y} - \frac{j}{L_X}\right| \tag{8}$$

$$\alpha_{ij} = \frac{\exp(d_{ij})/\tau}{\sum_{j}^{L_X} \exp(d_{ij})/\tau}$$
(9)

$$z_i = \sum_j^{L_X} \alpha_{ij} \cdot e_j, \tag{10}$$

where d_{ij} is the distance between the target token y_i and source token x_j normalized by specific length, α_{ij} represents the weight and e_j is the embedding of source x_j . τ is the temperature of the softmax function set as 0.2 in our experiment.

3.2 Hybrid Strategy

We compare three strategies to mix copied source embeddings with masks (denoted as Z and Mfor simplicity) with the baseline (All mask) — All copied, Weighted Sum and our heuristicallyproposed Fence Replace, as shown in Figure 1.

All Mask: It's exactly same as CMLM, served as baseline.



Figure 1: The hybrid strategy.

All Copied: We replace all embeddings of mask with copied source embedding added with position embedding, which is equivalent to entirely using the information from source.

Weighted Sum: It mixes the information by adding M and Z elementwise with a certain weight. We test 4 combinations ranging from 0.2 to 0.8 for M and Z with the stride set as 0.2, i.e. (0.2, 0.4, 0.6, 0.8)M + (0.8, 0.6, 0.4, 0.2)Z and report the best result where the weights are 0.6 and 0.4 for M and Z, respectively.

Fence Replace: During experiments, we find it's important to dynamically change the volume of information added to the decoder input with addition of Y_{obs} . Concretely, Z may become noise instead of useful signals when Y_{obs} has been fully capable to support MLM independently.

Therefore, we propose to replace the masked tokens at odd positions with half of Z exploited, avoiding the decoder input to incorporate too much information and ultimately act as noise. More formally, we first define a mask (0, 1, 0, 1, ..., 0, 1) like a fence with length of L and apply it to stagger the M and Z into a mixed embedding Z', where odd positions are filled with Z and even positions are filled with M. Finally, we replace the embedding of Y_{mask} with the mixed embedding of specific positions.

3.3 Training

To fit the proposed method and meanwhile take full advantage of the masked language model, we modify the training strategy by randomly replacing the subset of the original masked token with the copied source embedding, so that the proportion of corrupted tokens can be unchanged. We apply this method to train the model under all hybrid strategies, including All Mask, for convenient comparison, so it differs from the original CMLM in training.

4 **Experiments**

4.1 Experimental Setup

We evaluate HI-CMLMs with the proposed hybrid strategies on standard machine translation benchmarks including WMT14 En \leftrightarrow De and WMT16 En \leftrightarrow Ro in both directions.

Datasets The sizes of the dataset are 4.5M and 610k for $En\leftrightarrow De$ and $En\leftrightarrow Ro$ respectively. We create the knowledge distilled data as suggested in (Gu et al., 2018; Zhou et al., 2020) with same configurations. BPE (Sennrich et al., 2016) is used for tokenization with the vocabulary size set to 42k and 40k for $En\leftrightarrow De$ and $En\leftrightarrow Ro$.

Model Configurations We apply the same weight initialization method and configurations on hyperparameters as prior work: $n_{layers} = 12$, $n_{heads} = 8$, $d_{hidden} = 512$, $d_{FFN} = 2048$ (Ghazvininejad et al., 2019; Vaswani et al., 2017). Our model is trained on 4 Tesla V100 GPUs with the max batch size of 8k tokens per card. Adam (Kingma and Ba, 2015) is used for optimization. The learning rate warms-up for 20k steps to 5e-4 and decays with the inversed-sqrt schedular. We implement models in the experiment with fairseq (Ott et al., 2019).

4.2 Results and Analysis

Table 1 shows the performance of the proposed HI-CMLM with the BLEU score (Papineni et al., 2002). For each language pair, the model obtains consistent improvements with the Fence Replace, but no gains with another two.

Model	En-De	De-En	En-Ro	Ro-En
Transformer (Vaswani et al., 2017) Transformer (Our Implementation)	27.30 27.72	32.04	- 34.03	33.93
CMLM (Ghazvininejad et al., 2019)	27.03	30.53	33.08	33.31
CMLM (Our Implementation)	26.89	30.71	32.94	33.07
HI-CMLM + All Mask	27.01	30.74	32.89	33.03
HI-CMLM + All Copied	26.76	30.82	32.74	32.95
HI-CMLM + Weighted Sum	26.81	30.79	32.80	33.14
HI-CMLM + Fence Replace	27.42 (+0.53)	31.32 (+0.61)	33.36 (+0.42)	33.51 (+0.44)

Table 1: The performance of the AT teacher, the baseline CMLM, and the HI-CMLM with different hybrid strategies.

Length	CMLM	HI-CMLM (Fence Replace)
Overall	26.89	27.42 (+0.53)
[0,10)	22.24	23.27 (+1.03)
[10,23)	26.46	26.98 (+0.52)
[23,+∞)	27.61	27.81 (+0.20)

Table 2: BLEU scores of target sentences with different lengths at the 10-th step.



Figure 2: BLEU scores of every step with max_iter=10 for all hybrid strategies as well as the baseline.

To further investigate why Fence Replace stands out, we draw outputs of each step for four strategies in Figure 2 with max_iter=10. It shows from step 4, the model with All Copied and Weighted Sum strategy start to fall back to the All Mask level, which means for the later steps, the added information turns into noise, but can be appropriately controlled by the Fence Replace. We empirically explain how it controls below.

Results on different length targets We evaluate performance of Fence Replace over three bins based on the length of targets: [0, 10), [10, 23), and $[23, \infty)$. Table 2 shows more gains are obtained on short sentences. Intuitively, we guess the benefits result from the enhanced condition of $p(y_i|X)$, by complementing sparse X of short sentences with informative mix(Z, M) in early steps. But if so,



Figure 3: The proportion of copied source embedding within the masked area for sentences with different length when applying the Fence Replace strategy.

why All Copied and Weighted Sum do not work?

We show it's not the whole story. In Figure 3, the proportion of Z actually used for replacement has been reduced from step 6 for all length bins due to the sparsity of re-masked tokens, particularly for shorter sentences, it's much less than 50% that is pre-determined by fence and dropped faster. So the outstanding performance of Fence Replace is not only attributed to incorporated source embedding but the significantly-reduced proportion of Z in later steps as well, effectively avoiding Z from acting as noise.

This comprehensively reveals that the Fence Replace can flexibly balance the information feed to decoder inputs, more signals in early-stage refinement and less information in later steps.

5 Conclusion

We present HI-CMLM, an extension of CMLM by mixing source embedding with a hybrid strategy — Fence Replace, which can appropriately balance the information applied to the model. It achieves consistent improvements on two benchmarks in both directions, particularly short sentences.

References

- 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics.
- 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 2019, Hong Kong, China, November 3-7, 2019, pages 6111– 6120. Association for Computational Linguistics.
- Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Nonautoregressive neural machine translation. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. Open-Review.net.
- Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. 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 11179–11189.
- 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.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.
- Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, pages 1173– 1182. Association for Computational Linguistics.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and

Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

- 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, July 6-12, 2002, Philadelphia, PA, USA, pages 311–318. ACL.
- 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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers. The Association for Computer Linguistics.
- Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, volume 97 of Proceedings of Machine Learning Research, pages 5976–5985. PMLR.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008.
- Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Non-autoregressive machine translation with auxiliary regularization. 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 5377–5384. AAAI Press.
- Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020. Understanding knowledge distillation in nonautoregressive machine translation. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net.