SJTU-MTLAB's Submission to the WMT23 Word-Level Auto Completion Task

Xingyu Chen
Shanghai Jiao Tong University
galaxychen@sjtu.edu.cn

Rui Wang
Shanghai Jiao Tong University
wangrui12@sjtu.edu.cn

Abstract

Word-level auto-completion (WLAC) plays a crucial role in Computer-Assisted Translation. In this paper, we describe the SJTU-MTLAB's submission to the WMT23 WLAC task. We propose a joint method to incorporate the machine translation task to the WLAC task. The proposed approach is general and can be applied to various encoder-based architectures. Through extensive experiments, we demonstrate that our approach can greatly improve performance, while maintaining significantly small model sizes.

1 Introduction

In recent years, more and more researchers have studied computer-aided translation (CAT) that aims to assist human translators to translate the input text (Alabau et al., 2014; Knowles and Koehn, 2016; Hokamp and Liu, 2017; Santy et al., 2019; Huang et al., 2021; Weng et al., 2019). The word-level auto-completion (WLAC) task (Casacuberta et al., 2022) is the core function of CAT, which involves predicting the word being typed by the translator given the translation context, as illustrated in Figure 1. Effective auto-completion has the potential to reduce keystrokes by at least 60% during the translation process (Langlais et al., 2000). A user survey indicates that 90.2% of participants find the word-level auto-suggestion feature helpful (Moslem et al., 2022). Therefore, WLAC plays an important role in CAT.

There are many existing methods for modeling WLAC, and they mainly differ in model architectures (Li et al., 2021; Yang et al., 2022b; Moslem et al., 2022; Yang et al., 2022a; Ailem et al., 2022). For example, Li et al. (2021); Yang et al. (2022a) design a BERT-like architecture to directly predict the target word while Yang et al. (2022b) employ a model similar to the auto-regressive NMT to predict the BPE tokens of the target word.

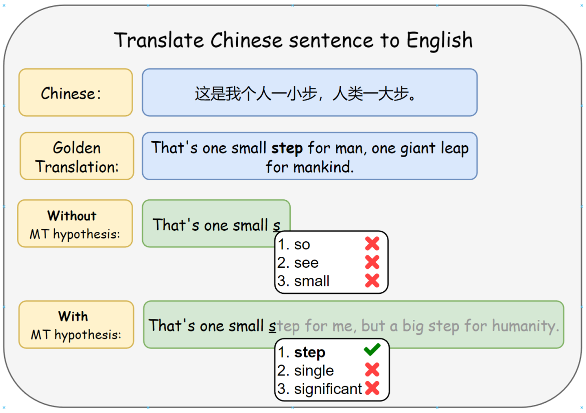
Translate Chinese sentence to English
Chinese:
这是我个人一小步，人类一大步。
Golden Translation:
That's one small **step** for man, one giant leap for mankind.
Without MT hypothesis:
That's one small s
1. so ✗
2. see ✗
3. small ✗
With MT hypothesis:
That's one small step for me, but a big step for humanity.
1. **step** ✓
2. single ✗
3. significant ✗

Figure 1: An example of word-level auto completion. Assume the human translator is going to input the *Golden Translation*. The auto-completion suggests the possible word candidates given the typed characters. It can be more accurate with the help of translation hypothesis from MT models.

The WLAC task comes from a real translation scenario: a human translator is translating a source sentence, who has already translated part of the sentence, and is typing a new word. The input contains three parts: the source sentence **s**, the partial translation **c**, and the typed sequence **t**. The WLAC task is to predict the word **w** that the translator is going to input (Li et al., 2021; Casacuberta et al., 2022). Rooted in the translation natural, we consider a fundamental question: what defines a correct word **w**? Theoretically, a good **w** should appear in the reference translation, as illustrated in Figure 1. Therefore, we attempt to incorporate knowledges from machine translation into the WLAC task. We presents two novel approach to enhance WLAC systems, called joint-inference and joint-training, to combine the MT task and the WLAC task during inference and training, respectively.

The effectiveness of our proposed method is validated through experiments conducted on the four language directions of the WLAC shared task

Proceedings of the Eighth Conference on Machine Translation (WMT), pages 872–876
December 6–7, 2023. ©2023 Association for Computational Linguistics

in WMT2023 (§4). Remarkably, our approach achieves substantial improvements across two distinct backbone models.

2 Backbone Models for WLAC

In this section, we introduce two types of backbone models for the WLAC task. These backbone models serve as the foundation for our proposed techniques and experiments in subsequent sections.

Word-level Model The first backbone is called All-In-One Encoder (AIOE), which adopts a BERT-like(Devlin et al., 2019) Transformer Encoder architecture for word prediction similar to Li et al. (2021). The AIOE takes the concatenation of the source sentence, context, and typed sequence as its input. The input format is: $\boldsymbol{s}$ <*sep*> $\boldsymbol{c}_l$ <*tip*> $\boldsymbol{t}$ <*mask*> $\boldsymbol{c}_r$, where $\mathbf{c}_l$ is the left context to the input and $\mathbf{c}_r$ is the right context. Specifically, we append a <*mask*> token at the end of the typed sequence and leverage the final hidden state of the <*mask*> token for word prediction.

Despite its simplicity and efficiency, the AIOE model suffers from the out-of-vocabulary (OOV) problem, which can significantly hinder its performance. To this end, we introduce a variance of AIOE model that predicts word in sub-word level.

Sub-word-level Model Extending the word-level AIOE model to sub-word-level is straightforward: we consider the task of predicting a sequence of sub-words as a generation problem, and introduce a Transformer Decoder to the AIOE model to perform the generation. We use Byte-Pair Encoding (BPE) (Sennrich et al., 2016) for the sub-word tokenization, and call this model *AIOE-BPE*.

Due to the difficulty of labeling the WLAC data, we generate training data from parallel corpus for training the WLAC models, following the standard practice (Li et al., 2021; Casacuberta et al., 2022).

3 Enhancing WLAC by incorporating MT task

In this section, we propose two different approaches to improve the WLAC task.

3.1 Joint Inference with MT Model

This approach is to jointly consider the WLAC predictions and machine translation results during inference. We begin by generating the top-k predictions from the WLAC model. Next, we examine each word in the predictions and check if it is included in the translation. The first word in the top-k list that exists in the translation is selected as the final prediction. This strategy manually align the prediction with translation in a flexible way: the choice of WLAC model and translation model is arbitrary. The final performance is closely related to the choices of models.

However, this approach heavily relies on the quality of translation. A preliminary analysis show that for a naive MT model, only 44.6% of the WLAC labels exist in the translation. One possible solution is to enhance the input of MT model. We propose a *Context MT* model, which takes additional translation context and typed sequence as input, and generates full target sentence. The input of *Context MT* is the same as WLAC, so it's a better approximation of the golden translation model.

3.2 Joint Training with MT Task

One drawback of joint inference method is that the WLAC model isn't aware of the translation task during training, which means that the top-k predictions may deviate from the ground truth. To overcome this limitation, we propose a joint training approach, wherein the WLAC model and the MT model are trained together using a shared backbone encoder. Specifically, we extend the backbone model by introducing an MT decoder, transforming the whole model into an MT model. Here the MT model is the same as *Context MT* model described in §3.1. We define the training loss of the joint training model as the combination of the WLAC loss and the translation loss, represented as follows:

$$\mathcal{L} = \alpha \cdot L_{\text{WLAC}} + (1 - \alpha) \cdot L_{\text{MT}}, \qquad (1)$$

where α is a hyper-parameter controlling the balance between the two losses. To enhance the interaction between two tasks, we also share the final word prediction layer between the backbone model and the decoder. As described in section 4.1, the training data of WLAC is generated from parallel corpus, so there will be a full agreement between WLAC label and ground truth translation. This agreement enables the WLAC model to learn how to accurately predict words within the translations. Besides, the MT model can learn to generate translations based on the context provided by the WLAC predictions. By jointly training the two models, we enable them to mutually benefit from each other's knowledge and improve their respective tasks.

Model	#Parameters	zh-en	en-zh	en-de	de-en
AIOE	80M	46.71	54.82	51.75	50.64
AIOE-BPE	74M	50.79	53.48	57.23	61.96
AIOE+Joint Training	80M(105M)	51.40	58.70	56.22	54.57
AIOE-BPE+Joint Training	74M(100M)	**56.93**	**61.16**	**67.27**	**68.16**

Table 1: Experiment results on WMT23 WLAC test set. Results are reported as accuracy. The number of parameters in brackets means parameters in training stage.

The key advantage of joint training is that once the training is completed, we can only keep the backbone model and discard the MT decoder. Note that the backbone encoder can receive optimization signals from both the WLAC task and the translation task, so the backbone model has acquired the skill to agree with translation during training process. This enables us to maintain the agreement capabilities while preserving a small and efficient inference model.

4 Experiment

4.1 Datasets

We conduct evaluations of our model on two language pairs: English-Chinese and English-German. The zh-en dataset we used is the UN Parallel Corpus V1.0 from WMT17. For en-de, we use the training data from WMT14. We adopt the following strategy on parallel sentences to generate WLAC training data: firstly, we sample a target word $\mathbf{w}$ from the target language sentence, then we sample spans respectively from the left and right context of the target word, denoted as $\mathbf{c}_l$ and $\mathbf{c}_r$. Additionally, we sample a typed sequence from the target word. To sample typed sequence from Chinese words we use the pypinyin[1] tool. All models are trained on the generated training data, with data generated from the test set of WMT21 translation task serving as the validation set. For evaluation, we utilize the test set from the WMT22 WLAC shared task.

4.2 Experiment Details

For all AIOE model, we use a Transformer Encoder for 6 layers. The embedding size is 512, the dimension for feed-forward layer is 2048. Each layer has 8 attention heads. For AIOE-BPE model, we additionally add a Transformer Decoder with 6 layers. The MT decoder for joint training models are also 6 layers.

For AIOE model, we use a joint-vocabulary with the size of 120000. For AIOE-BPE model, the vocabulary size is 66630 for English-Chinese pair and 59918 for English-German pair.

The learning rate for training is 5e-4. We optimize the model for 200000 steps with a batch size of 32000 tokens. We average five checkpoints for better performance.

4.3 Comparison among Joint Methods

We firstly compare the performance of joint inference method and joint training method. For joint inference method, we use the word-level backbone AIOE model for the WLAC model, and consider two kinds of machine translation model: translation model trained on parallel corpus (MT) and translation model trained on WLAC input and translation output (*Context MT*). For the joint training method, we use AIOE-Joint model. All the experiments are conduct in zh-en direction. We conduct preliminary experiments on the WLAC22 test set and the result is reported in Table 2.

Method	Acc.
AIOE	53.87
AIOE+MT	54.20
AIOE+CMT	56.01
AIOE+JT	59.75

Table 2: Comparison of joint-methods. *Acc.* is the accuracy of WLAC22 task. AIOE+MT and AIOE+CMT is joint-inference method combined with different MT models. AIOE+JT is the joint training method.

It is observed that joint inference methods greatly outperform the baseline model, and the joint training method further improves the performance. The Context MT model is better than normal MT model for joint-inference, suggesting that more translation context is beneficial for the WLAC prediction. However, the overall performance of joint-inference is hindered by the quality of MT models, and the joint-training method can incorporate MT

[1] https://github.com/mozillazg/python-pinyin

Model	#Parameters	zh-en	en-zh	en-de	de-en
GWLAN(Li et al., 2021)	105M	51.11	48.90	40.69	53.87
HW-TSC(Yang et al., 2022b)	526M	59.40	-	63.82	62.06
AIOE+Joint Training	80M(105M)	59.75	56.59	44.67	62.77
AIOE-BPE+Joint Training	74M(100M)	**61.08**	**58.09**	**64.59**	**66.91**

Table 3: Experiment results on WMT22 WLAC test set. We implement the GWLAN model report the performance. The scores of HW-TSC model are copied from Yang et al. (2022b)

.

knowledge with WLAC more effectively. Based on these findings, we only focus on the joint training method for the subsequent experiments.

4.4 Main Results

The evaluation result on WLAC shared task is reported on Table 1. Our BPE-level methods have obtained better performance than word-level model except for en-zh, which indicates the word-level model may suffer from the OOV problem. No matter which backbone is used, our joint training method can greatly improve the backbone performance, indicating that our method is a general framework and has the potential to be applied to more encoder based models. Another obvious advantage of our model is its superior parameter efficiency. Our AIOE-BPE+Joint Training model achieves the best performance with only 100M training parameters and 74M parameters for inference.

4.5 Comparison with other models

We further compare our methods with existing systems. The experiment result on the WLAC22 test set is shown in Table 3. Compared to HW-TSC, our word-level methods have obtained better performance on zh-en and de-en. One exception is en-de, the word-level model performed badly because it suffers from OOV problem, where about 17% labels are OOV. After replacing the backbone with BPE-level model, our method show superior performance in all directions, while maintaining a much smaller size.

4.6 The impact of MT task

The influence of the hyper-parameter α on the model performance, as outlined in equation 1, directly reflects the impact of translation task. By setting α to 0, the model is essentially a translation model with additional context input. If $\alpha = 1$, the model corresponds to the AIOE model without joint training. In Figure 2, we present the accuracy achieved at varying values of α. Notably, as α increases from 0 to 0.75, the accuracy increases rapidly. This observation highlights the difference between the translation task and the WLAC task, emphasizing the necessity of optimizing the model specifically for the WLAC task to achieve better performance. Interestingly, even with α set to 0.99, the performance remains comparable to the best achieved performance. This finding is remarkable, as it suggests that even a small signal from the translation task can greatly enhance the WLAC task's performance when compared to the model with α set to 1. Consequently, our proposed joint training method effectively integrates the translation task into the WLAC task, resulting in substantial improvements.

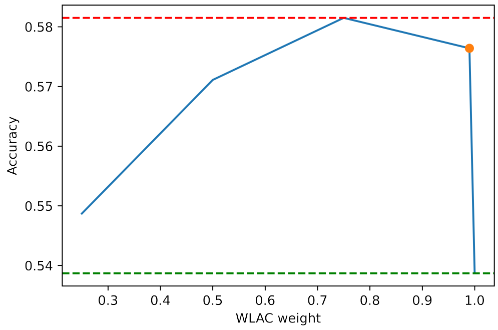

Figure 2: The impact of different α on the AIOE accuracy. Red dashed line is the best performance and the green represents the accuracy without joint training.

5 Conclusion

This paper proposes an effective approach to improve WLAC performance by combining the MT task and the WLAC task. We inject the translation knowledge into the WALC model by jointly train the two tasks. Extensive experiments show that the proposed approach surpasses several strong baselines with much smaller model size.

Acknowledgments

Xingyu and Rui are with MT-Lab, Department of Computer Science and Engineering, School of Electronic Information and Electrical Engineering, and also with the MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200204, China. Xingyu is mainly supported by Tencent Rhinobird Fund (RBFR2023012) . Rui is partially supported by the General Program of National Natural Science Foundation of China (6217020129), Shanghai Pujiang Program (21PJ1406800), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102),CCF-Baidu Open Fund (F2022018), and the Alibaba-AIR Program (22088682)

References

Melissa Ailem, Jingshu Liu, Jean-Gabriel Barthélemy, and Raheel Qader. 2022. Lingua custodia's participation at the wmt 2022 word-level auto-completion shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1170–1175.

Vicent Alabau, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin Hill, Philipp Koehn, Luis A Leiva, et al. 2014. Casmacat: A computer-assisted translation workbench. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

Francisco Casacuberta, George Foster, Guoping Huang, Philipp Koehn, Geza Kovacs, Lemao Liu, Shuming Shi, Taro Watanabe, and Chengqing Zong. 2022. Findings of the word-level autocompletion shared task in wmt 2022. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 812–820.

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.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. *arXiv preprint arXiv:1704.07138*.

Guoping Huang, Lemao Liu, Xing Wang, Longyue Wang, Huayang Li, Zhaopeng Tu, Chengyan Huang, and Shuming Shi. 2021. Transmart: A practical interactive machine translation system. *arXiv preprint arXiv:2105.13072*.

Rebecca Knowles and Philipp Koehn. 2016. Neural interactive translation prediction. In *12th Conferences of the Association for Machine Translation in the Americas: MT Researchers' Track, AMTA 2016, Austin, TX, USA, October 28 - November 1, 2016*, pages 107–120. The Association for Machine Translation in the Americas.

Philippe Langlais, George Foster, and Guy Lapalme. 2000. Transtype: a computer-aided translation typing system. In *ANLP-NAACL 2000 Workshop: Embedded Machine Translation Systems*.

Huayang Li, Lemao Liu, Guoping Huang, and Shuming Shi. 2021. GWLAN: General word-level AutocompletioN for computer-aided translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online.

Yasmin Moslem, Rejwanul Haque, and Andy Way. 2022. Translation word-level auto-completion: What can we achieve out of the box? *arXiv preprint arXiv:2210.12802*.

Sebastin Santy, Sandipan Dandapat, Monojit Choudhury, and Kalika Bali. 2019. INMT: interactive neural machine translation prediction. 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 - System Demonstrations*, pages 103–108. 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.

Rongxiang Weng, Hao Zhou, Shujian Huang, Lei Li, Yifan Xia, and Jiajun Chen. 2019. Correct-and-memorize: Learning to translate from interactive revisions. *arXiv preprint arXiv:1907.03468*.

Cheng Yang, Siheng Li, Chufan Shi, and Yujiu Yang. 2022a. Iigroup submissions for wmt22 word-level autocompletion task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*.

Hao Yang, Hengchao Shang, Zongyao Li, Daimeng Wei, Xianghui He, Xiaoyu Chen, Zhengzhe Yu, Jiaxin Guo, Jinlong Yang, Shaojun Li, et al. 2022b. Hw-tsc's submissions to the wmt22 word-level auto completion task. In *Proceedings of the Seventh Conference on Machine Translation Shared Tasks Papers. Association for Computational Linguistics*.