HW-TSC's Submissions to the WMT22 Word-Level Auto Completion Task

Hao Yang, Hengchao Shang, Zongyao Li, Daimeng Wei, Xianghui He,
Xiaoyu Chen, Zhengzhe Yu, Jiaxin Guo, Jinlong Yang
Shaojun Li, Yuanchang Luo, Yuhao Xie, Lizhi Lei, Ying Qin
Huawei Translation Service Center, Beijing, China
{yanghao30, shanghengchao,lizongyao,weidaimeng,hexianghui,
chenxiaoyu35,yuzhengzhe,guojiaxin1,yangjinlong7,
lishaojun18,luoyuanchang,xieyuhao2,leilizhi,qinying}@huawei.com

Abstract

This paper presents the submissions of Huawei Translation Services Center (HW-TSC) to WMT 2022 Word-Level AutoCompletion Task. We propose an end-to-end autoregressive model with bi-context based on Transformer to solve the current task. The model uses a mixture of subword and character encoding units to realize the joint encoding of human input, the context of the target side and the decoded sequence, which ensures full utilization of information. We use one model to solve four types of data structures in the task. During training, we try using a machine translation model as the pre-trained model and fine-tune it for the task. We also add BERT-style MLM data at the fine-tuning stage to improve model performance. We participate in zh→en, en→de, and de→en directions and win the first place in all the three tracks. Particularly, we outperform the second place by more than 5% in terms of accuracy on the zh→en and en→de tracks. The result is buttressed by human evaluations as well, demonstrating the effectiveness of our model.

1 Introduction

In recent years, machine translation quality has improved significantly with advances in model architecture (Sutskever et al., 2014; Bahdanau et al., 2014; Gehring et al., 2017; Vaswani et al., 2017), bilingual data availability, as well as data augmentation strategies (Liu et al., 2016; Freitag et al., 2017; Johnson et al., 2017; Zhang et al., 2018; Edunov et al., 2018; Wu et al., 2019; Li et al., 2019). In scenarios where machine translation is used to facilitate understanding, machine translation outputs can basically satisfy audience's demands. However, in areas where translation quality is crucial (such as translating product manual, patent description, etc.), post-editing is required. Techniques to improve post-editing efficiency are meaningful and necessary. Researches (Barrachina et al., 2009; Green et al., 2014; Knowles and Koehn, 2016; Santy et al., 2019) in this regard falls into this category of computer-aided translation (CAT).

Word-Level auto Completion, as a new task in WMT22, fall into this category as well. This task aims at auto-completing a target word given a source sentence, translation context, and a human-typed character sequence, so as to improve post-editing efficiency. Li et al. (2021) define the task in detail, offer comprehensive analysis and provide a baseline system.

For this task, we choose the subword-level modeling strategy (Kudo and Richardson, 2018). Comparing with word-level modeling, this strategy enables the usage of pre-trained models from other mainstream tasks, and solves the out-of-vocabulary (OOV) issues at the same time. As human-typed input is just several characters of the target word, the input is not suitable for subword segmentation. We use character-level encoding instead. Our final submission is an autoregressive model with bi-context, ensuring mixed encoding of characters and subwords.

In view of the possible discontinuity between the context and human-typed inputs in the target-side text, we use tags to wrap the inputs, and then encode jointly with the context, in conjunction with the autoregressively decoded pre-token sequence. The joint coding maximizes the usage of information without introducing additional RNN (Cho et al., 2014) or vocabulary reduction modules.

During training, we use a standard machine translation model as the pre-trained model, and fine-tune it for this task. We then add BERT-style Mask Language Model (MLM) (Devlin et al., 2018) data in the fine-tuning stage to enhance the language model capabilities of the decoder, thereby improving the overall model performance.

The inference mechanism is different from that of traditional NMT. In general, the entire decoder sequence must be used for encoding each token,

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1192–1197
Abu Dhabi, December 7–8, 2022.

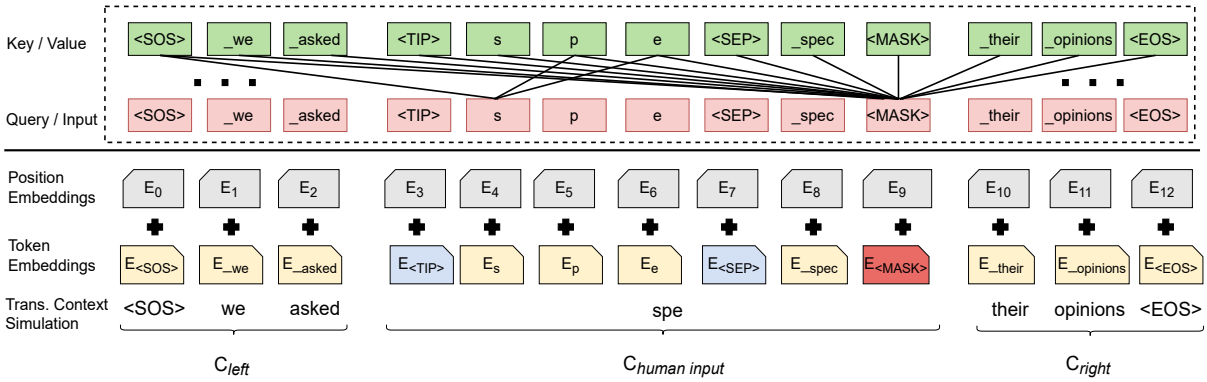

Figure 1: The input representation of our model's decoder. C_{left} and C_{right} are the context. E_s, E_p, E_e are the char embedding of human input "spe". <TIP> is the separator for human input and left context. <SEP> is the separator for human input and decoded sequence. <MASK> represents the potenial target word in this translation context.

which reduces the inference performance to a certain extent. However, given the model's parallel ability and the short decoding sequence (the number of subwords in a word), this issue is not very serious for this task and this strategy is applicable for practical use. Figure 1 shows the input representation of our model's decoder.

We submit results for three language directions. All of them achieve the highest accuracy. Our Zh→En and En→De submissions even outperform the second place by 5% in terms of accuracy, and get a good lead in human evaluation, demonstrating the effectiveness of our strategy.

2 Data Processing

Zh→En data for this task comes from UN V1.0 (about 15.9M) and En↔De data comes from WMT14 (about 4.5M). The task allows the use of additional monolingual data, but we add no additional data on the zh-en track given the amount of bilingual data available. An additional 24M monolingual data is used for the En-de track, and the data comes from the WMT news task as well. we generate synthetic parallel data by sampling BT (Edunov et al., 2018) for the En→De track and by beam BT for the Dn→En track. The specific reasons will be given later.

We follow basic strategies to cleanse the data, including: deduplication, garbled character filtering, XML conversion, and fast-align(Dyer et al., 2013), etc. The data sizes before and after data processing are shown in table 1.

As for subword, we employ sentencpiece on Zh→En track, and set the vocabulary size to 36k. On En→De track, we use the BPE algorithm, and set the vocabulary size to 32K.

Lang-Pair	Origin	Cleaned
Zh-En	15.9M	15.5M
En-De	4.5M	4.3M

Table 1: Overview of training para data.

Figure 2: The joint encoding of context and human input.

3 System design

In this chapter, we introduce the model structure, training strategy, inference strategy and corresponding data generation strategy used for this task. Our model is based on the encoder-decoder architecture of the standard Transformer.

3.1 Model Structure

We use Transformer as our model architecture. For convenience, we only use a 25 encoder layers and 6 decoder layers deep model. The parameters of the model are the same as Transformer-big. We just change the post-layer-normalization to the pre-layer-normalization (Sun et al., 2019), and increase the number of encoder layers to 25.

3.2 Modeling Units

In general, we use a mixed encoding strategy that encoding subword-level and character-level information at the same time. To be more specific, the model conducts subword-level encoding on source and target context information, and character-level encoding on human-typed input, as it is just several characters of a word. Apparently, the model can

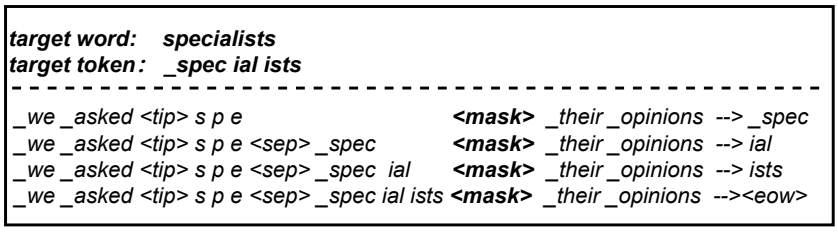

Figure 3: The process of data generation.

encode the two types of information at the same time.

3.3 Joint Encoding

In the above chapter, we discussed our modeling granularity for context and human-typed input. According to the settings of the task, context information and human-typed input information may be discontinuous. Here we insert the tag <tip> before and after the tip to wrap the human-typed input to distinguish the two. Schematic diagram is shown in Figure 2.

The context and human-typed input of the translation test can be jointly encoded, which ensures the maximum usage of information.

3.4 Autoregressive Decoding with Bi-context

In the task, there are four types of data: left-context, right-context, zero-context and bi-context. If we use four models to process these four types of data, the problem can be solved, but the task will be complicated. Instead, we can regard the first three types as special cases of the last type, so we directly design an autoregressive decoding strategy with bi-context, and use a single model to process all types of data.

To be more specific, decoding is performed with Mask token as the anchor point. The encoder encodes the source-side text. The mask, in conjunction with context and human-typed input, is encoded at the decoder side to predict the first subword of the target word. The first subword (pre-token) will be encoded as well. Then the model continues to use the mask for decoding until a complete word is decoded. The overall architecture diagram of the model decoder is shown in figure 1.

Here are two points: 1. The mask token replaces the second tip described in the previous section. The mask token is capable of distinguishing the human-typed input from other information and masking at the same time. 2. The newly added <sep> tag is responsible for distinguishing between human-typed input and decoded pre-token information.

3.5 Data Generation

Based on our previous coding strategy for various types of information, we first use the script provided by the organizer to generate word-level training data. Then, we use the subword-level model to perform subword processing on the source text and translation context. Regarding the target word, we add a tag <eow> at the end of word after subword segmentation. Assuming that the number of subwords is N, we generate N sets of training data to simulate the entire autoregressive process. We call the data as WLAC (Word Level Auto Completion) data to distinguish it from terms. A case study of the generation is show in Figure 3.

In the process of generating data, we also add some rules to improve efficiency. We remove sentences with too short target words or too long human-typed input by a given probability. In addition, we keep only 1/10 of the training samples of eos that predict the end of the sentence.

In order to effectively validate the performance of the model during training, we generate a test set using the same strategy. WMT19 news test is used for the Zh→En track, and WMT14 new-test is used for En↔De tracks. We do not use a filtering strategy when generating the test set.

4 Experiment

During the experiment, we first build a baseline based on the MT model in order to measure our model's performance more accurately. After that, in the training process, we adopt several strategies to improve the model performance, that is, fine-tuning a MT model and introducing BERT-style MLM data. Validation and debugging of these strategies are done on the Zh→En track. We use the finally determined strategy to train models for other tracks.

Lang-Pair	Baseline	MT-tune	Mix-tune	Average	Ensemble	WMT22
zh-en	62%	74%	77.19%	78.69%	78.96%	59.40%
en-de	-	dvivied	81.79%	82.86%	82.80%	62.06%
de-en	-	-	77.83%	79.05%	79.77%	63.82%

Table 2: The main results of our experiments. MT-tune refers to using WLAC data to fine-tune a standard MT model. MLM-Mix-tune refers to using WLAC and BERT-style MLM data to fine-tune the MT model.

4.1 Baseline based on the MT Model

First, we consider whether the current task can be solved by directly relying on the outputs by an MT model trained with bilingual data. By doing so, we lower the requirements of this strategy and regard a case as correct as long as the predicted word appears in the MT result.

We obtain an accuracy of 62.5% on the Zh→En track by using the above-mentioned approach. We use this as a benchmark for optimization. If our designed strategy cannot exceed this level of accuracy, the strategy fails.

4.2 MT Model Finetune

After obtaining the baseline MT model, we then fine-tune it using the generated WLAC data. It should be noted that the self-attention layer of the standard NMT model's decoder does not have the ability to generate attention to the right, and our decoder is a mask-based prediction model, so we need to break this limitation. This is also a gap between the two tasks.

4.3 BERT-Style MLM Data Fine-tune

In the fine-tuning stage on the basis of a MT model, through analyzing each type of data, we find that the the accuracy of prefix is higher than that of the suffix. We assume this is because there is no right-side information during the training a pre-trained NMT model. As a result, the model may learn right-context less efficiently than the left-context.

According to the task setting, the context of the target side is an incomplete fragment and is given randomly. At the same time, tips are not necessarily continuous, so the overall translation is relatively confusing. Source-side information is important so we deepen the number of encoding layers. In addition, using the mask as the decoding anchor causes the decoder to change from the standard language model mode to the mask language model mode. To address these issues, we add a same proportion of BERT-style MLM data with source-side information. Given the availability of original text, we enlarged the probability of the mask to 25%.

4.4 Average and Ensemble

Finally, we adopt commonly used strategies to improve the model performance, including averaging and ensemble, and we find that both of the strategies lead to performance improvement. Particularly, averaging brings significant improvement.

5 Result and Analysis

Due to time restriction, we only conduct detailed comparison experiments on the Zh→En track. En↔De tracks simply follows the final strategy we apply to the Zh→En track. The results are shown in Table 2.

First of all, the performance of the MT baseline we trained is very poor, indicating that the MT task is not well adapted to the current task. So we give up the idea of using the MT results to enhance the model performance.

After our constructed WLAC data is added, the model performance improves by nearly 12 points, indicating the effectiveness of our strategy. But it is worth noting that the En→De model does not converge after adding the ST/BT data. We assume that the quality of the ST data is not good. In addition, the target-side of WLAC data is confusing, resulting in training failure. So for the De→En model, we directly generate BT data based on beam search to avoid the issue.

After MLM data is added, we again obverse a significant improvement. The accuracy on the Zh→En reaches 77.19%. The En→De model, which was not converged at the previous stage, gains an accuracy of 81.79%. The results support our assumption that MLM data can enhance the language model ability of the decoder, while avoiding noise interference from the source-side text.

In the end, model averaging leads to improvements on all three tracks, and the improvement is more significant than ensemble. Ensemble leads to significant improvement on the De→En track but

limited improvement on the other two tracks. We assume this is because the De→En model is not sufficiently trained due to time restriction.

6 Conclusion

In this paper, we detail our team's participation in the WMT22 word-level AutoComplete task. We first analyze the input and output, as well as challenges in this task. We notice that the modeling granularity of human-typed input and context information are different. Therefore, we propose modeling human-typed input at character-level and modeling context information at subword-level, explicitly distinguishing and jointly encoding the two, thereby maximizing information usage in the encoding stage. At the same time, there is a semantic discontinuity between context and human-typed input. We add tags to differentiate the two. Finally we propose an autoregressive model with bi-context to process four types of data at the same time. During training, we use an NMT model as the pre-trained model and fine-tune it for this task. BERT-style MLM data is also introduced to improve the model performance, and at the same time to solve the single-direction decoding issue of the self-attention model. In the end, our models are well adapted to the task and gain safe leads in both automatic and human evaluations.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, et al. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 644–648.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *CoRR*, abs/1702.01802.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR.

Spence Green, Sida I Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christopher D Manning. 2014. Human effort and machine learnability in computer aided translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1225–1236.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Rebecca Knowles and Philipp Koehn. 2016. Neural interactive translation prediction. In *AMTA (1)*, pages 107–120.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The niutrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 257–266.

Huayang Li, Lemao Liu, Guoping Huang, and Shuming Shi. 2021. Gwlan: General word-level autocompletion for computer-aided translation. *arXiv preprint arXiv:2105.14913*.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *Proceedings of the 2016 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 411–416.

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): System Demonstrations*, pages 103–108.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 374–381.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural 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 2019, Hong Kong, China, November 3-7, 2019*, pages 4205–4215.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.