Findings of the Word-Level AutoCompletion Shared Task in WMT 2023*

Lemao Liu [1] **Francisco Casacuberta** [2] **George Foster** [3] **Guoping Huang** [1]
Philipp Koehn [4] **Geza Kovacs** [5] **Shuming Shi** [1] **Taro Watanabe** [6] **Chengqing Zong** [7]
[1] Tencent AI Lab [2] Universitat Politècnica de València [3] Google
[4] Johns Hopkins University [5] LILT [6] Nara Institute of Science and Technology
[7] Institute of Automation, Chinese Academy of Sciences

Abstract

This paper presents the overview of the second Word-Level autocompletion (WLAC) shared task for computer-aided translation, which aims to automatically complete a target word given a translation context including a human typed character sequence. We largely adhere to the settings of the previous round of the shared task, but with two main differences: 1) The typed character sequence is obtained from the typing process of human translators to demonstrate system performance under real-world scenarios when preparing some type of testing examples; 2) We conduct a thorough analysis on the results of the submitted systems from three perspectives. From the experimental results, we observe that translation tasks are helpful to improve the performance of WLAC models. Additionally, our further analysis shows that the semantic error accounts for a significant portion of all errors, and thus it would be promising to take this type of errors into account in future.

1 Introduction

Computer-aided translation (CAT) helps human translators produce high-quality translations with the assistance of machine translation systems (Koehn et al., 2003; Vaswani et al., 2017), and it has witnessed a lot of attention during the past decades (Bowker, 2002; Koehn, 2009; Foster et al., 1997; Langlais et al., 2000; Barrachina et al., 2009; Alabau et al., 2014; Knowles and Koehn, 2016; Santy et al., 2019; Huang et al., 2021). Among all the tasks in CAT, Word-Level autocompletion (WLAC) is one of the most fundamental tasks and its goal is to autocomplete a word when a human translator types a sequence of characters (Huang et al., 2015; Li et al., 2021), in order to accelerate the editing process for human translators under CAT settings. To facilitate the research in WLAC, the first WLAC shared task was held in WMT 2022 (Casacuberta et al., 2022; Yang et al., 2022; Ángel Navarro et al., 2022; Moslem et al., 2022; Ailem et al., 2022). This year, we continue holding the second edition of WLAC shared task in WMT 2023.

In this paper, we summarize the overview of the WLAC shared task in WMT 2023, which is named by WLAC 2023 for brevity, including data preparation process, submitted systems and their evaluation results. Specifically, WLAC 2023 involves two language pairs, i.e. Chinese-English and German-English, and contains four directional sub-tasks in total, similar to WLAC 2022 shared task. For training data preparation, we follow the common practice of leveraging a bilingual corpus for simulation. For test data preparation, however, there is one important difference in this year to make the test data more similar to realistic scenarios: for some testing examples (see §2.2), their typed character sequences are obtained from the typing process of human translators.

We have received twenty-one submissions in total from four teams in WLAC 2023. We evaluate all these submissions and present their overall evaluation results. In particular, we conduct a thorough analysis of submitted systems to better understand the challenges and difficulties emerged in WLAC tasks. The analysis of these systems is investigated according to three perspectives which include the frequency of target words, the size of context, as well as the human defined error types. From all the perspectives, we observe some insights which might be useful for further improvement on WLAC in future. In summary, our main findings are highlighted as follows:

1. Through effective use of translation models, it is able to substantially benefit the WLAC models in terms of accuracy.

2. Among all type of errors, the semantic error

* The authors are listed alphabetically except the first author.

Proceedings of the Eighth Conference on Machine Translation (WMT), pages 654–662
December 6–7, 2023.

makes up the majority of error cases, where predicted words are semantically deviated from the ground-truth words.

3. It is possible to directly use large language models (LLMs) for WLAC tasks, but the results show that currently LLMs can not effectively handle WLAC without fine-tuning.

2 Task Description and Data Preparation

2.1 Task Definition

WLAC tasks aim to auto-complete a target word for the CAT process. The definition of WLAC is as follows: given a source sequence x, translation context $c = (c_l, c_r)$, where c_l and c_r are left and right side context respectively, and a character-level typed sequence s by human translators, WLAC aims to predict the target word w with s as its prefix, which should be the most appropriate to be placed between c_l and c_r (Huang et al., 2015; Li et al., 2021). Formally, we expect to model the relationship following the below equation:

$$w = f(x, s, c_l, c_r) \quad (1)$$

More generally, the right or left side context could be empty in real-world CAT systems. Consequently, there are four types of situations should be considered in WLAC tasks:

1. zero-context: both c_l and c_r are empty;
2. suffix: c_l is empty while c_r is non empty;
3. prefix: c_l is non empty while c_r is empty;
4. bi-context: both c_l and c_r are non empty.

	EN-DE	EN-ZH
Sentence Pairs	4,465,840	15,886,041
Words (src/tgt)	120M/114M	441M/395M

Table 1: The statistical description of the total number of sentence pairs and the scale of tokenized words on English ⇔ German and English ⇔ Chinese language pairs.

2.2 Data Preparation

We mainly follow the previous edition settings for data preparation, which includes two language pairs, i.e. English ⇔ Chinese and English ⇔ German. Both translation directions are considered in the evaluation, resulting in four directional tasks.

Training Data Following previous edition settings, we employ simulated training data $\langle x, s, c, w \rangle$ for this year WLAC. The construction of which follows the algorithm proposed by Li et al. (2021) [1]. The reason of such a simulation is to compensate for the limited size of manually annotated training data.

Specifically, for English ⇔ German language pair, we use the WMT14 EN-DE training dataset preprocessed by Stanford NLP Group [2], which is about 4.5 million sentence pairs; For English ⇔ Chinese pair, we leverage UN Parallel Corpus dataset [3] from WMT17, which consists of 15 million sentence pairs. Moses tokenizer [4] is applied to both English and German sentences while Jieba [5] is used to segment Chinese sentences. The detailed statistical description of the datasets is shown in Table 1.

For a fair comparison, only the above-mentioned corpus is allowed to be employed for bilingual training. However, there is no limitation for any monolingual data usage and even for pre-trained language models (Devlin et al., 2018) or large language models such as ChatGPT and Llama (Touvron et al., 2023).

Testing Data Similar to the data preparation in WLAC 2022, testing data in this year consists of two types of datasets as well. **Type I** is the conventional simulation on bilingual data which follows the same construction rules as the training data; **Type II** testing data is obtained from the real-world post-editing scenario. To alleviate any information leakage about the testing sets, the bilingual dataset and post-editing data are created by a third-party company [6] to guarantee that both data are not included in the training data.

In details, to create the testing examples for Type II testing set, we focus on the words that the translators had modified and then sample their context according to four types. [7] In particular, unlike WLAC 2022 where the typed sequence is randomly sam-

[1] The scripts for simulation is available at https://github.com/lemaoliu/WLAC.

[2] https://nlp.stanford.edu/projects/nmt/data

[3] https://conferences.unite.un.org/UNCorpus/Home/DownloadOverview

[4] https://github.com/moses-smt/mosesdecoder

[5] https://github.com/fxsjy/jieba

[6] We paid about 10,000 dollars to obtain the test data from the third-party company.

[7] Since the sentences from post-editing naturally belong to bi-context type, we need to obtain all types of examples via randomly sampling context.

Date Type	ZH⇒EN	EN⇒ZH	DE⇒EN	EN⇒DE
Sentence Pairs				
Type I	11341	11430	9653	9367
Type II	5044	5173	4910	5172
Overall	16385	16603	14539	14564
Averaged Length (src/tgt)				
Type I	28.88/4.71	31.88/4.46	28.73/4.47	29.18/4.43
Type II	32.29/5.71	35.66/5.42	32.93/5.46	33.61/5.24
Overall	29.16/4.85	32.22/4.58	29.19/4.59	29.72/4.53

Table 2: The total number of testing examples for both Type I and II cases over four language pair directions. A/B denotes that A is the averaged number of source words in the source sentences and B is the averaged number of target words in the context.

Data Type	ZH⇒EN	EN⇒ZH	DE⇒EN	EN⇒DE
Bi-context				
Type I	2489	2514	2081	2021
Type II	1107	1139	1060	1117
Overall	3596	3653	3141	3138
Prefix				
Type I	3884	3902	3416	3315
Type II	1729	1766	1739	1830
Overall	5613	5668	5155	5145
Suffix				
Type I	2499	2534	2098	2033
Type II	1113	1147	1066	1123
Overall	3612	3681	3164	3156
Zero-Context				
Type I	2466	2479	2058	1997
Type II	1098	1122	1046	1103
Overall	3564	3601	3104	3100

Table 3: The number of testing examples on four types of context cases for each sub-tasks.

pled according to target words, in WLAC 2023 the typed sequences for Type II dataset are obtained according to the typing process of human translators. This makes examples in Type II data more realistic than those in WLAC 2022.

Finally, when generating testing examples from the parallel sentences and post-edited sentences, we increase the proportion of *Prefix* type this year because the *Prefix* context type is more likely to match the popular left-to-right interactive translation systems. The statistics of sentence pairs are shown on Table 2 and the statistics of the different context types are shown on Table 3.

3 Experimental Setting

3.1 Evaluation Metric

According to the findings from WLAC 2022 (Casacuberta et al., 2022), the automatic evaluation result is highly consistent with the human evaluation result on the same dataset. Hence, in this year, we only employ the automatic evaluation for the submitted systems. Specifically, we use accuracy as the automatic evaluation metric (Li et al., 2021) to demonstrate the performance of all submitted systems:

$$acc = \frac{N_{\text{match}}}{N} \quad (2)$$

where N_{match} is the total number of correctly predicted words and N is the total number of all testing samples.

3.2 Submitted Systems

We received 21 submissions from 4 teams. We briefly summarize their approaches below.

SJTU-MTLAB The SJTU-MTLAB participates in all language directions. They submitted both word-level model and BPE-level model and their BPE-level model performs better (Chen and Wang, 2023). The BPE-level model is based on the Transformer architecture with encoder and decoder, where the encoder take the source sentence and all context as input and the decoder is responsible for generating the target word. They also introduce another decoder to generate the full target sentence, and jointly train the full model with WLAC task and machine translation task. The translation decoder is discarded during inference to maintain a reasonable inference cost. For more details about this system, it can be found in Chen et al. (2023).

Systems	ZH-EN	EN-ZH	EN-DE	DE-EN
Traditional Supervised Method				
SJTU-MTLAB	56.93	61.16	67.27	68.16
HW-TSC	56.40	57.80	66.42	68.10
PRHLT/sys1	-	-	37.05	39.98
PRHLT/sys2	-	-	37.38	43.56
Few-Shot Method				
KnowComp/0-shot	9.82	-	9.72	7.53
KnowComp/1-shot	21.43	-	14.96	15.34
KnowComp/5-shot	27.74	-	21.98	22.95

Table 4: Official evaluation results for all submitted systems. The score is reported in accuracy.

HW-TSC The Huawei Translation Services Center (HW-TSC) participates in all language directions. They model the WLAC task in the BPE level and iteratively generates a subword to compose the prediction word. [8] Specifically, they employ an encoder-decoder architecture, where the encoder encodes the source sentence and the decoder takes as input the target side context. They first train a machine translation task as a baseline and then they fine-tune the baseline with WLAC data and BERT-style MLM data to get the final model.

KnowComp KnowComp group proposes a large language model (LLM) based system for this year's WLAC task. They first randomly sample in-context examples as prompts to obtain the row ChatGPT outputs and extract the final prediction by post-processing (Wu et al., 2023). Specifically, they provide the source sentence x and target sentence with a special token `[mask]` as a placeholder for x (i.e., $(c_l, \texttt{[mask]}, c_r)$), and let LLMs predict the word that should fill in the mask position. Since more than one word may be generated, they search for the first word that starts with the pre-typed sequence s as the final prediction. They evaluate the submitted systems in Chinese ⇒ English, German ⇒ English, and English ⇒ German directions.

PRHLT PRHLT group participates in English ⇔ German and German ⇔ English categories. Their submitted system is developed on a segment-based interactive machine translation (IMT) system (Ángel Navarro et al., 2023). It predicts the results by word correction task based on a sequence of segmented contexts. Moreover, to further enhance the system performance under zero-context situations, they developed a dictionary-based translation module for zero-context word completion. Additionally, they made a second submission which fine-tunes an LLM (mT5) (Xue et al., 2020) to adapt it to the WLAC task. To perform this fine-tuning they created a new parallel dataset in which source sentences are the concatenation of the original source sentences + left context + right context + typed sequence, and the target sentences are the autocompletions.

[8] HW-TSC team does not submit the system report this year, but it is told that the system is very similar to that used in WLAC 2022 by personal communication with the team members.

4 Experimental Result and Analysis

4.1 Evaluation Results

Overall result The overall evaluation results of all submissions are reported on Table 4. The performance of HW-TSC and SJTU-MTLAB are comparable, and both systems perform the best among all the submissions. Both HW-TSC and SJTU-MTLAB make use of the knowledge from machine translation, and the large gains indicate the effectiveness to incorporate WLAC task with machine translation task according to the experiments in the system report of Chen and Wang (2023). Furthermore, we can see that fine-tuning the large mT5 model (PRHLT/sys2) delivers substantial improvements over PRHLT/sys1. It is worth noting that the KnowCamp system does not involve re-training for WLAC tasks, and thereby it is unfair to compare it with other systems which are trained with the large scale of the supervised training data. Anyway, its evaluation result still shows that the large language model can not handle the WLAC task well without fine-tuning on the training data.

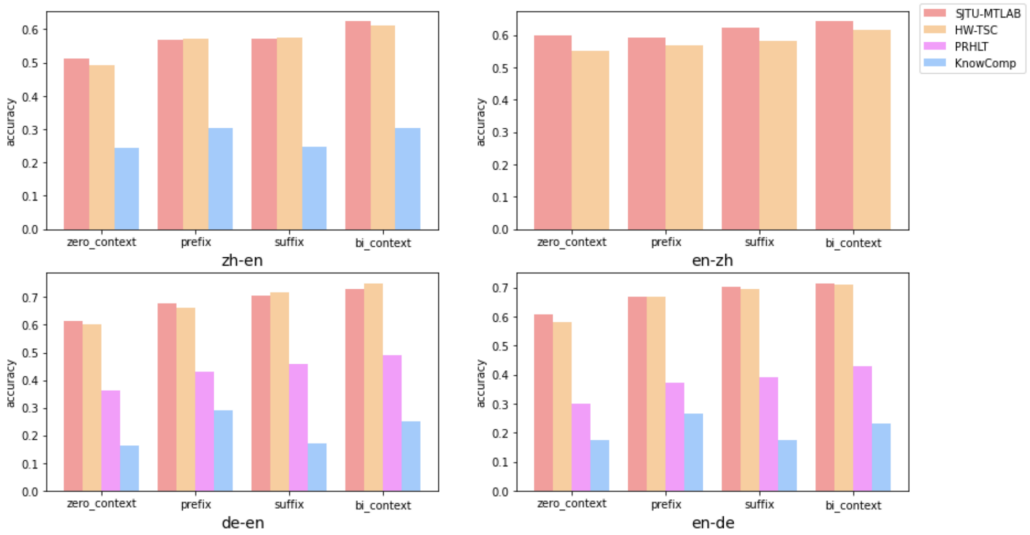

Figure 1: The accuracy of all language directions among different context types.

Result for context types In addition to the overall result, we also evaluate the submissions according to different context types of testing examples for all sub-tasks. The accuracy of four systems for four context types is illustrated in Figure 1, where only the best system from each team is evaluated. As we can see, for most systems, the accuracy increases from *zero_context* to *bi_context*. This indicates that more context can bring better performance. One exception is KnowComp, which does not perform well in *zero_context* and *suffix*. One of the possible reasons is that the large language models would find it difficult to make a correct prediction with little(*zero_context*) or unusual context type (the setting of *suffix* is contradictory with the left-to-right paradigm in large language models).

4.2 Analysis

In this subsection, we conduct a thorough analysis on the evaluation results from three perspectives. Since the analysis results are similar across different language directions, we conduct the following analysis on the de-en direction, because all the systems have submitted results on this direction.

Frequency The first perspective is to analyze the accuracy according to the word frequency. To this end, we divide testing examples into 16 bins according to the frequency of their ground-truth word: suppose an example is with a frequency of f ($f \geq 1$), then it is placed into the bin with id as the rounding number of $\min(16, \log f)$. Then we calculate the accuracy for each bin and the result is depicted in Figure 2. From the figure it is observed that it is very difficult to predict the rare words (i.e. their frequency is zero) in WLAC, which is in line with the task of neural machine translation (Luong et al., 2015). When the frequency is more than one, the accuracy is much higher than that for frequency of zero; however, the accuracy does not strictly increase as the frequency gets higher than one.

Context size The second perspective is to analyze the accuracy of each system according to the context size. The number of words in the left and right contexts indicate whether the context provides the sufficient information to predict the target word and thus the accuracy of each system might be influenced by the context size. Since different examples may have different length in the sentence, we group the examples into bins according to the relative context size defined by the ratio of the context size to the size of the source sentence. Then we measure the accuracy for each bin and the results for all systems are illustrated in Figure 3. As shown from this figure, for all the supervised systems the accuracy generally increases when the relative context size becomes larger. However, the KnowComp system seems to be insensitive to the context size. This fact may indicate that KnowComp does not make full use the context, which provides an explanation why KnowComp does not work well for WLAC.

Error Analysis In order to look deeper into the reason why the models make wrong prediction, we propose to manually analyze the errors made by each system, which is the third perspective. To this end, we first define three types of errors for each

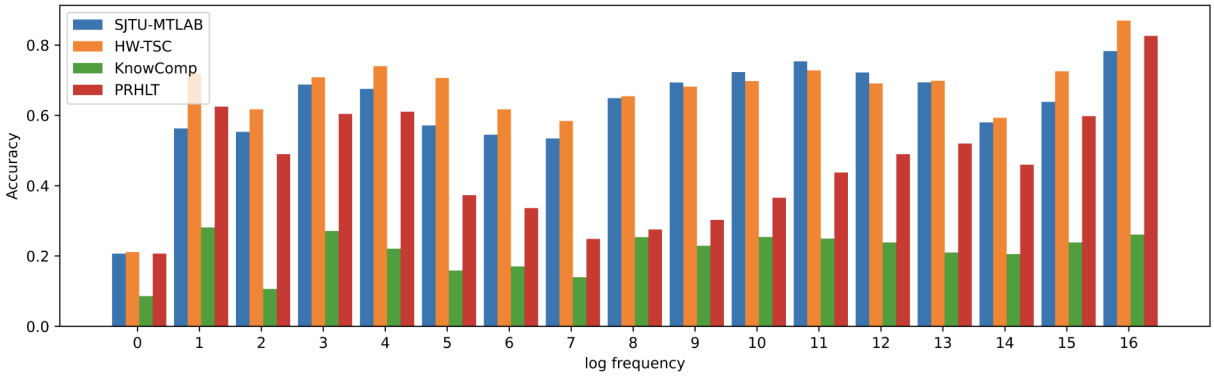

Figure 2: The accuracy of different bins organized according to the frequency of ground-truth target word. Each bin id corresponds to the rounding number of $\min(16, \log f)$ with f as the frequency of the target word.

incorret prediction. The first one is the *constraint error*, where the predicted word fails to meet the constraint of typed character sequence. There are two main reasons to this error: 1) the system does not use the hard constraint manner during inference; 2) the system uses the hard constraint during inferenct but still can not predict a word which satisfies the constraint due to some unusual typed character sequence. Another common type of errors is morphology error, where the prediction has similar semantics with ground truth but has different morphology. For example, the ground truth is *needs* while the prediction is *need*. We detect this type of error by *nltk.stem* [9] tool. The third error is called semantic error, where the predictions are completely deviated from the ground-truth words in semantic. To measure how much the prediction deviate from the ground truth, we use the fastText[10] tool to compute the semantic similarity of predictions and ground truths. We report the proportion of errors where the semantic similarity of prediction and label is less than 0.3.

The results for all systems are reported in Table 5. According to the constraint errors, the SJTU-MTLAB and HW-TSC can meet the constraint well, while the LLM based method, KnowComp, often fails to generate a proper word with given typed character sequence. After manually checking the results from all these systems, we find that both SJTU-MTLAB, PRHLT and KnowComp does not employ the hard constraint during inference and HW-TSC sometimes can not predict a word satisfying the constraint due to unusual typed sequences. In addition, according to the morphology error, as we can see from Table 5, there are still a non-negligible amount of predictions fall into this group, indicating the potential for further improvement. Finally, according to the semantic error, as reported in Table 5, most of the errors belongs to this group. This is the most critical error type, and we recommend reducing this part of the error is very promising to improve the overall performance.

4.3 Discussion on future direction

Through the overall results and analysis, we point out some possible direction of further improvement:

- Incorporating machine translation task. The SJTU-MTLAB and HW-TSC introduce machine translation into the WLAC task and show superior performance. This indicates the importance of adding translation knowledge into WLAC and we encourage more effective method to combine these two tasks.

- Improving large language models for WLAC. KnowComp employs the large language models through in-context learning for WLAC. Although its performance is not as good as other systems, it still exhibits potential because it does not leverage the large-scale supervised data for training. Indeed, simply fine-tuning the large mT5 model on the supervised data yields respectful results (see PRHLT/sys2). Therefore, it is promising to further improve LLMs by using of the supervised data.

- Alleviating the semantic error. The large amount of semantic error indicates that the current systems still fail to model the problem in many cases. We expect the development of more powerful models to push the SOTA forward by taking semantic error into account.

[9] https://www.nltk.org/api/nltk.stem.html
[10] https://fasttext.cc/

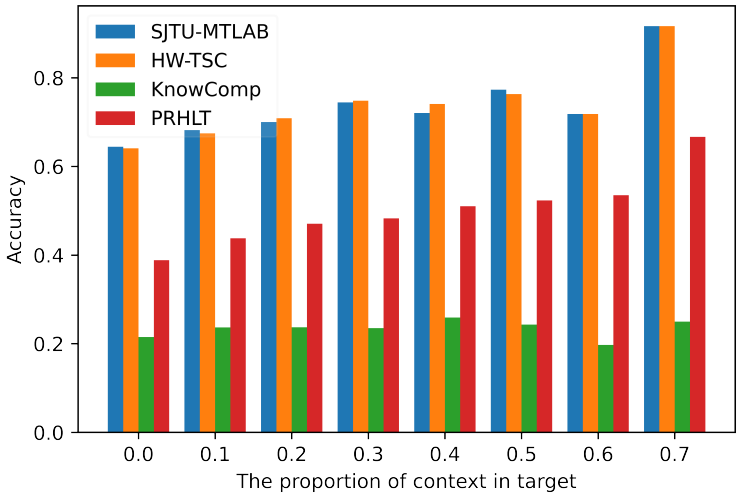

Figure 3: The accuracy of different bins organized according to the relative context size of each example. The relative context size is defined by the ratio of the size of left and right contexts to the size of the source side for each example.

Systems	Constraint	Morphology	Semantic
SJTU-MTLAB	4.12%	12.62%	57.69%
HW-TSC	1.89%	10.98%	56.28%
KnowComp	22.30%	6.77%	74.57%
PRHLT	7.18%	10.23%	59.85%

Table 5: The proportion of different types of error among constraint error, morphology error, and the semantic error respectively. The sum of each line does not equal to 1 because different types or error may share overlaps.

5 Conclusion

This paper presents the overview for the shared task of Word-level Auto-Completion, which is the key component of computer-aided translation. We describe the task definition, data preparation process, the submitted systems, evaluation metric and evaluation results of the systems. We have received twenty-one submissions from four participants this year. We report the evaluation results of all systems, conduct a thorough analysis on the prediction results of these systems and obtain some insightful findings. We hope that our findings can encourage the emerge of more powerful models and attract more researchers to participate the study of computer-aided translation.

Acknowledgements

We would like to appreciate the annotators for their creating test data on this shared task. In addition, we thank the participants for their contributions on this shared task.

References

Melissa Ailem, Jinghsu 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 Shared Tasks Papers*. Association for Computational Linguistics.

Vicent Alabau, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin L 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*, pages 25–28.

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.

Lynne Bowker. 2002. *Computer-aided translation technology: A practical introduction*. University of Ottawa Press.

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.

Xingyu Chen, Lemao Liu, Guoping Huang, Zhirui Zhang, Mingming Yang, Shuming Shi, and Rui Wang. 2023. Rethinking word-level auto-completion in computer-aided translation. In *Proceedings of EMNLP*. Association for Computational Linguistics.

Xingyu Chen and Rui Wang. 2023. Sjtu-mtlab's submission to the wmt23 word-level auto completion task. In *Proceedings of the Seventh Conference on Machine Translation Shared Tasks Papers*. Association for Computational Linguistics.

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*.

George Foster, Pierre Isabelle, and Pierre Plamondon. 1997. Target-text mediated interactive machine translation. *Machine Translation*, 12:175–194.

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*.

Guoping Huang, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2015. A new input method for human translators: integrating machine translation effectively and imperceptibly. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Rebecca Knowles and Philipp Koehn. 2016. Neural interactive translation prediction. In *Conferences of the Association for Machine Translation in the Americas: MT Researchers' Track*, pages 107–120.

Philipp Koehn. 2009. A process study of computer-aided translation. *Machine Translation*, 23:241–263.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

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)*, pages 4792–4802.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China. Association for Computational Linguistics.

Yasmin Moslem, Rejwanul Haque, and Andy Way. 2022. Word-level auto-completion: What can we achieve out of the box? In *Proceedings of the Seventh Conference on Machine Translation Shared Tasks Papers*. Association for Computational Linguistics.

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.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yi Wu, Haochen Shi, Weiqi Wang, and Yangqiu Song. 2023. Knowcomp submission for wmt23 word-level autocompletion task. In *Proceedings of the Seventh Conference on Machine Translation Shared Tasks Papers*. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

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, and Ying Qin. 2022. 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.

Ángel Navarro, Miguel Domingo, and Francisco Casacuberta. 2022. Prhlt's submission to wlac 2022. In *Proceedings of the Seventh Conference on Machine Translation Shared Tasks Papers*. Association for Computational Linguistics.

Ángel Navarro, Miguel Domingo, and Francisco Casacuberta. 2023. Prhlt's submission to wlac 2023. In *Proceedings of the Seventh Conference on Machine Translation Shared Tasks Papers*. Association for Computational Linguistics.