# PRHLT's Submission to WLAC 2022

**Ángel Navarro** and **Miguel Domingo** and **Francisco Casacuberta**
PRHLT Research Center
Universitat Politècnica de València
{annamar8,midobal,fcn}@prhlt.upv.es

## Abstract

This paper describes our submission to the Word-Level AutoCompletion Shared Task of the WMT 2022. We participate in the pair of languages, English–German, in both ways. We propose a segment-based interactive machine translation approach whose central core is a machine translation (MT) model that predicts the complete translation from the context provided by the task and picks the word we were trying to autocomplete from there. We show with this approach that it is possible to use the MT models in the autocompletion task by performing minor changes at the decoding step and obtaining good accuracy.

## 1 Introduction

Machine translation (MT) has significantly improved in recent years with the emergence of neural machine translation (NMT), but it still cannot assure high-quality translations for all tasks (Toral, 2020). For those scenarios with rigorous translation quality requirements, it is critical for professional translators to manually validate the translations generated by the NMT system. The computer-aided translation (CAT) tools show up to improve the validation and editing process carried out by translators. Researchers approached CAT tools from many directions with the aim of reducing the human effort of correcting the automatic translations. Among CAT tools such as translation memory (Zetzche, 2007), augmented translation (Lommel, 2018) and terminology management (Verplaetse and Lambrechts, 2019); we can find autocompletion tools, which help professional translators by providing new partial translations according to the validated parts they have supplied to the system.

Word level autocompletion (WLAC) (Li et al., 2021) is a new shared task introduced in WMT22. Its aim is to complete a target word given a source sentence, a sequence of characters typed by the human translator and a translation context. Four types of context are possible:

**Zero-contex:** no context is given.

**Suffix:** a sequence of translated words located after the word to autocomplete.

**Prefix:** a sequence of translated words located prior to the word to autocomplete.

**Bi-contex:** A combination of the *suffix* and the *prefix* type. That is, there is a sequence of translated words located after the word to autocomplete, and a sequence of translated words located prior to the word to autocomplete.

Note that, in all cases, the word to autocomplete is not necessarily consecutive to these contexts.

We have experimented with a similar CAT tool from the interactive machine translation (IMT) framework. In this field of research, the translation is generated in a collaborative process between the human translator and the MT model. Among the different approaches, the segment-based IMT (Domingo et al., 2017; Peris et al., 2017) protocol presents certain similarities with WLAC: at each step, the user validates sequences of translated words—the context—and makes a correction—the word to autocomplete.

Therefore, in this work we have approached WLAC as a simplification of segment-based IMT, using the context as the validated segments and the typed characters as the word correction; and limiting the process to the first iteration. This has allowed us to tackle WLAC by training a conventional NMT model and adapting it at the decoding step.

## 2 Segment-based interactive machine translation

Segment-based IMT establishes a framework in which a human translator works together with

*Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1182 - 1186
December 7-8, 2022 ©2022 Association for Computational Linguistics

| | | |
|---|---|---|
| **SOURCE** (x): | | Una versión traducida de un texto. |
| **REFERENCE** (y): | | A translated version of a text. |
| **ITER-0** | ($\tilde{\mathbf{f}}$) | ( ) |
| | ($\hat{y}$) | A written version of a story. |
| **ITER-1** | ($\tilde{\mathbf{f}}$) | (A) (version of a) |
| | ($s$) | t |
| | ($\hat{y}$) | A translated version of a document. |
| **ITER-2** | ($\tilde{\mathbf{f}}$) | (A translated version of a) |
| | ($s$) | t |
| | ($\hat{y}$) | A translated version of a text. |
| **FINAL** | ($\hat{y} \equiv y$) | A translated version of a text. |

(a) Segment-based IMT session.

| $c_l$ | s | $c_r$ |
|---|---|---|
| A | t | version of a |
| A | translated | version of a |

(b) WLAC session.

Figure 1: Examples of a segment-based IMT session to translate a sentence from Spanish to English; and a WLAC session for predicting a word for a source sentence, a translation context, and a human-typed character sequence.

the MT system to produce the final translation. This collaboration starts with the system proposing an initial translation hypothesis $y_1^I$ of length $I$. Then, the user reviews this hypothesis and validates those sequence of words which they consider to be correct ($\tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_N$; where $N$ is the number of non-overlapping validated segments). After that, they are able to merge two consecutive segments $\tilde{\mathbf{f}}_i$, $\tilde{\mathbf{f}}_{i+1}$ into a new one. Finally, they correct a word—which introduces a new one-word validated segment, $\tilde{\mathbf{f}}_i$, which is inserted in $\tilde{\mathbf{f}}_1^N$. This correction can also consist in a partially typed word $\tilde{\mathbf{f}}'_i$, in which case the system would complete it as part of its prediction.

The system's reacts to this user feedback by generating a sequence of new translation segments $\widehat{\mathbf{g}}_1^N = \widehat{\mathbf{g}}_1, \ldots, \widehat{\mathbf{g}}_N$; where each $\widehat{\mathbf{g}}_n$ is a subsequence of words in the target language. This sequence complements the user's feedback to conform the new hypothesis:

$$\begin{cases} \hat{y}_1^I = \tilde{\mathbf{f}}_1, \widehat{\mathbf{g}}_1, \ldots, \tilde{\mathbf{f}}'_i\widehat{\mathbf{g}}_i, \ldots, \tilde{\mathbf{f}}_N, \widehat{\mathbf{g}}_N \text{ if } \tilde{\mathbf{f}}'_i \in \tilde{\mathbf{f}}_1^N \\ \hat{y}_1^I = \tilde{\mathbf{f}}_1, \widehat{\mathbf{g}}_1, \ldots, \tilde{\mathbf{f}}_N, \widehat{\mathbf{g}}_N \text{ otherwise} \end{cases} \tag{1}$$

The word probability expression for the words belonging to a validated segment $\tilde{\mathbf{f}}_n$ was formalized by Peris et al. (2017) as:

$$p(y_{i_n+i'} \mid y_1^{i_n+i'-1}, x_1^J, f_1^N; \Theta) = \mathbf{y}_{i_n+i'}^{\top} \mathbf{p}_{i_n+i'}, \quad 1 \leq i' \leq \hat{l}_n \tag{2}$$

where $l_n$ is the size of the non-validated segment generated by the system, which is computed as follows:

$$\hat{l}_n = \underset{0 \leq l_n \leq L}{\arg\max} \frac{1}{l_N + 1} \sum_{i'=i_n+1}^{i_n+l_n+1} \log p(y_{i'} \mid y_1^{i'-1}, x_1^J; \Theta) \tag{3}$$

## 3 Approach

Given a source sentence $x_1^J$, a sequence of typed characters $s_1^K = s_1, \ldots, s_K$ and a context $\mathbf{c} = \{\mathbf{c}_\mathrm{l}, \mathbf{c}_\mathrm{r}\}$, where $\mathbf{c}_\mathrm{l} = c_{\mathrm{l}1}, \ldots, c_{\mathrm{l}S}$ and $\mathbf{c}_\mathrm{r} = c_{\mathrm{r}1}, \ldots, c_{\mathrm{r}R}$; WLAC aims to autocomplete $s_1^K$ to conform the word $w_1^W = s_1, \ldots, s_K, w_{K+1}, \ldots, w_W$. If we consider the context as the sequence of segments validated by the user ($\tilde{\mathbf{f}}_1^N = \mathbf{c}_\mathrm{l}, \mathbf{c}_\mathrm{r}$) and the sequence $s_1^K$ as the partially-typed word correction (which would be inserted in $\tilde{\mathbf{f}}_1^N$ as a new one-word validated segment; leading to $\tilde{\mathbf{f}}_1^N = \mathbf{c}_\mathrm{l}, s_1^K, \mathbf{c}_\mathrm{r}$), we can view WLAC as a simplification of segment-based IMT. With that in mind, we can rewrite Eq. (1) as:

$$\hat{y}_1^I = \mathbf{c}_\mathrm{l}, \widehat{\mathbf{g}}_1, s_1^K \widehat{\mathbf{g}}_2, \mathbf{c}_\mathrm{r}, \widehat{\mathbf{g}}_3 \tag{4}$$

which, knowing that the prediction of the partially-typed correction corresponds to the first word of $\widehat{\mathbf{g}}_2$, can be rewritten as:

$$\hat{y}_1^I = \mathbf{c}_\mathrm{l}, \widehat{\mathbf{g}}_1, s_1^K w_{K+1}^W, \widehat{\mathbf{g}}'_2, \mathbf{c}_\mathrm{r}, \widehat{\mathbf{g}}_3 \tag{5}$$

Therefore, we can obtain the autocompleted word ($w_1^W = s_1^K w_{K+1}^W$) by performing a single step of the segment-based IMT protocol, discarding the rest of the translation prediction.

Figure 1 illustrates an example of segment-based IMT compared to the WLAC task for the same case.

In the segment-based IMT (Fig. 1a) example, at iteration 0, the system generates an incorrect first hypothesis. The, at iteration 1, the user validates a sequence of segments and types the first character of the word 'translated' to help the system fulfill the sequence of words between the first two segments. After that, the system generates a new translation with all the validated segments and the human-typed character sequence. The system repeats this process at the second iteration, ending with a correct translation. The WLAC (Fig. 1b) example simplifies the case that happens at iteration 1. Although we have the same source sentence, validated segments (left and right context) and human-typed character sequence, in this case, the system only has to find one word between the two segments instead of generating the whole sentence.

## 4 Experimental setup

In this section, we present the details of our experimental session.

### 4.1 Evaluation

The WLAC 22 shared task selected accuracy as the automatic metric with which to report the evaluation of the different systems[1]. This metric is computed as the total number of correctly predicted words normalized by the total number of words to complete:

$$\text{Acc} = N_{\text{match}}/N_{\text{all}} \tag{6}$$

where $N_{\text{match}}$ is the number of predicted words that are identical to the human desired word, and $N_{\text{all}}$ is the total number of testing words.

### 4.2 Corpora

We conducted our experiments using the English–German corpus provided by the organizers, which is a version of the WMT14's dataset, preprocessed by Stanford NLP Group. We saved 2000 sentences to use as validation, which we processed with the provided script[2] in order to create the simulated data. Table 1 presents the data statistics.

Table 1: Statistics of the WLAC 2022 corpus. *Avg.* stands for average, *Run.* for running, $K$ for thousands and $M$ for millions.

| Partition | Characteristic | De | En |
|---|---|---|---|
| Training | Sentences | $4M$ | |
| | Avg. Length | 25 | 26 |
| | Run. Words | $110M$ | $116M$ |
| | Vocabulary | $1.6M$ | $800K$ |
| Validation | Sentences | 2000 | |
| | Avg. Length | 27 | 27 |
| | Run. Words | $53K$ | $53K$ |

### 4.3 Systems

Our MT systems were trained using *OpenNMT-py* (Klein et al., 2017). We made use of two different network architectures: recurrent neural network (RNN) (Hochreiter and Schmidhuber, 1997) and Transformer (Vaswani et al., 2017).

The RNN model uses an encoder–decoder architecture with long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) cells. We set the size of the encoder, decoder and word embedding layers to 512. The encoder and decoder models use a single hidden layer of the same size. We used Adam (Kingma and Ba, 2014) as the learning algorithm, with a learning rate of 0.0002 with a batch size of 10.

The Transformer model uses a word embedding size of 512. The hidden and output layers were set to 2048 and 512, respectively. Each multi-head attention layer has eight heads, and we stacked six encoder and decoder layers. We used Adam as the learning algorithm, with a learning rate of 2.0, $b_1$ of 0.9 and $b_2$ of 0.998. We set the batch size to 4096 tokens.

Additionally, we made use of the byte pair encoding (BPE) (Sennrich et al., 2016) algorithm, which was jointly trained on both languages of the dataset, applying a maximum number of 10.000 merges.

Finally, we used our own implementation (based on *OpenNMT-py*) of segment-based IMT, which we adapted for WLAC. This implementation is openly available[3] for the benefit of the community.

## 5 Results

In this section we present our experimental results. We trained four different models, alternating between the RNN and Transformer architectures and the use of the BPE algorithm on the En-

[1] A human evaluation was also performed.

[2] https://github.com/lemaoliu/WLAC/raw/main/scripts/generate_samples.py.

[3] https://github.com/PRHLT/OpenNMT-py/tree/word-level_autocompletion.

Table 2: Experimental results, measured in terms of accuracy. Test values are taken from the official evaluation. Best results from the validation set are denoted in **bold**.

| Partition | Approach | De–En | En–De |
|---|---|---|---|
| Validation | RNN | 0.568 | **0.535** |
| | RNN + BPE | 0.554 | 0.498 |
| | Transformer | 0.563 | 0.524 |
| | Transformer + BPE | **0.586** | **0.534** |
| Test | Transformer + BPE | 0.390 | 0.340 |

glish–German language pairs.

Prior to submitting our systems, we used the synthetic validation dataset created from the provided data (see Section 4.2). As reflected in Table 2, all approaches yielded similar results. They correctly completed the word the user was trying to type around 60% of the time. Since the *Transformer + BPE* combination yielded a two points improvement for De–En, and also achieved—together with the *RNN* approach—the best results for En–De, we selected this model for our submission.

Table 2 also contains the official accuracy scores published by the organizers. For the blind test, our system's performance dropped near a 20%. While we are waiting for the publication of the findings to have a better understanding of the cause of this drop, we suspect that it is related with the test set being from a different domain than the training data, which would have a considerable impact in our MT model.

All in all, these results show that the segment-based IMT methodology is a promising approach to adapting an MT model to the WLAC task. Moreover, due to the shared task constrains, we trained our systems using only the data provided by the organizers. However, one of the benefits of our approach is that any MT system can be easily adapted to be used for WLAC.

## 6 Conclusions

In this work, we have presented our submission to WLAC shared task from WMT22. Our approach consisted in adapting the segment-based IMT methodology to the WLAC task, which allows us to use a conventional NMT model to tackle this task by simply adapting it at the decoding step. We tested some of the most used NMT architectures, achieving very encouraging results.

As a future work, we would like to test our approach using a more robust NMT system, adapted to the domain of the task to perform—instead of training an ad hoc system, as we did in this work due to the task restrictions.

## Acknowledgements

The research leading to these results has received funding from *Generalitat Valenciana* under project *PROMETEO/2019/121*.

## References

Miguel Domingo, Álvaro Peris, and Francisco Casacuberta. 2017. Segment-based interactive-predictive machine translation. *Machine Translation*, 31:1–23.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of the Association for Computational Linguistics: System Demonstration*, pages 67–72.

Huayang Li, Limao Liu, Guoping Huang, and Shuming Shi. 2021. GWLAN: General Word-Level AutocompletioN for computer-aided translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 4792–4802.

Arle Lommel. 2018. Augmented translation: A new approach to combining human and machine capabilities. In *Proceedings of the Conference of the Association for Machine Translation in the Americas. Volume 2: User Track*, pages 5–12.

Álvaro Peris, Miguel Domingo, and Francisco Casacuberta. 2017. Interactive neural machine translation. *Computer Speech & Language*, 45:201–220.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Antonio Toral. 2020. Reassessing claims of human parity and super-human performance in machine translation at WMT 2019. *arXiv preprint arXiv:2005.05738*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Heidi Verplaetse and An Lambrechts. 2019. Surveying the use of CAT tools, terminology management systems and corpora among professional translators: general state of the art and adoption of corpus support by translator profile. *Parallèles*, 31(2):3–31.

Jost Zetzche. 2007. Translation memory: state of the technology. *Multilingual*, 18:34–38.