GWLAN: General Word-Level AutocompletioN for Computer-Aided Translation

Huayang Li Lemao Liu Guoping Huang Shuming Shi
Tencent AI Lab
{alanili,redmondliu,donkeyhuang,shumingshi}@tencent.com

Abstract

Computer-aided translation (CAT), the use of software to assist a human translator in the translation process, has been proven to be useful in enhancing the productivity of human translators. Autocompletion, which suggests translation results according to the text pieces provided by human translators, is a core function of CAT. There are two limitations in previous research in this line. First, most research works on this topic focus on sentence-level autocompletion (i.e., generating the whole translation as a sentence based on human input), but word-level autocompletion is under-explored so far. Second, almost no public benchmarks are available for the autocompletion task of CAT. This might be among the reasons why research progress in CAT is much slower compared to automatic MT. In this paper, we propose the task of general word-level autocompletion (GWLAN) from a real-world CAT scenario, and construct the first public benchmark[1] to facilitate research in this topic. In addition, we propose an effective method for GWLAN and compare it with several strong baselines. Experiments demonstrate that our proposed method can give significantly more accurate predictions than the baseline methods on our benchmark datasets.

1 Introduction

Machine translation (MT) has witnessed great advancements with the emergence of neural machine translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015; Wu et al., 2016; Gehring et al., 2017; Vaswani et al., 2017), which is able to produce much higher quality translation results than statistical machine translation (SMT) models (Koehn et al., 2003; Chiang, 2005; Koehn, 2009). In spite of this, MT systems cannot replace human translators, especially in the scenarios with rigorous translation quality requirements (e.g., translating product manuals, patent documents, government policies, and other official documents). Therefore, how to leverage the pros of MT systems to help human translators, namely, *Computer-aided translation* (CAT), attracts the attention of researchers (Barrachina et al., 2009; Green et al., 2014; Knowles and Koehn, 2016; Santy et al., 2019). Among all CAT technologies (such as translation memory, terminology management, sample sentence search, etc.), *autocompletion* plays an important role in a CAT system in enhancing translation efficiency. Autocompletion suggests translation results according to the text pieces provided by human translators.

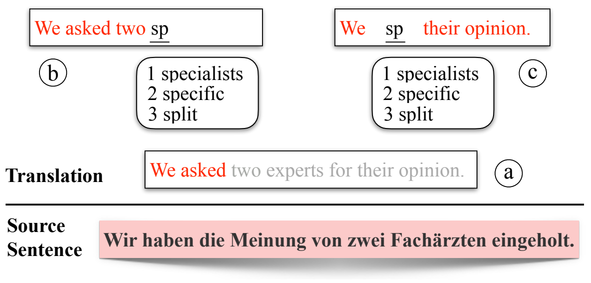

Figure 1: Illustration of Different Autocompletion Methods. The translation context is in red. Sub-figure in (a) is the sentence-level autocompletion, where the gray part is the completion generated by MT system. Both (b) and (c) are word-level autocompletion, underlined text "sp" is the human typed characters and the words in the rounded rectangles are word-level autocompletion candidates.

We note two limitations in previous research on the topic of autocompletion for CAT. First, most of previous studies aim to save human efforts by sentence-level autocompletion (Figure 1 a). Nevertheless, word-level autocompletion (Figure 1 b

[1]The information of benchmark datasets is in https://github.com/ghrua/gwlan

and c) has not been systematically studied. Second, almost no public benchmarks are available for the autocompletion task of CAT. Although some achievements have been made, research progress in CAT is more sluggish than that in automatic MT. The lack of benchmarks has hindered researchers from making continuous progress in this area.

In this work, we propose a **G**eneral **W**ord-**L**evel **A**utocompletio**N** (GWLAN) task, and construct a benchmark with automatic evaluation to facilitate further research progress in CAT. Specifically, the GWLAN task aims to complete the target word for human translators based on a source sentence, translation context as well as human typed characters. Compared with previous work, GWLAN considers four most general types of translation context: prefix, suffix, zero context, and bidirectional context. Besides, as in most real world scenarios, we only know the relative position between input words and the spans of translation context in the GWLAN task. We construct a benchmark for the task, with the goal of supporting automatic evaluation and ensuring a convenient and fair comparison among different methods. The benchmark is built by extracting triples of source sentences, translation contexts, and human typed characters from standard parallel datasets. Accuracy is adopted as the evaluation metric in the benchmark.

To address the variety of context types and weak position information issue, we propose a neural model to complete a word in different types of context as well as a joint training strategy to optimize its parameters. Our model can learn the representation of potential target words in translation and then choose the most possible word based on the human input.

Our contributions are two-fold:

- We propose the task of general word-level autocompletion for CAT, and construct the first public benchmark to facilitate research in this topic.
- We propose a joint training strategy to optimize the model parameters on different types of contexts together. [2]

2 Related Work

Computer-aided translation (CAT) is a widely used practice when using MT technology in the industry. As the the MT systems advanced and improved, various efficient interaction ways of CAT have emerged (Vasconcellos and León, 1985; Green et al., 2014; Hokamp and Liu, 2017; Weng et al., 2019; Wang et al., 2020). Among those different methods, the autocompletion is the most related to our work. Therefore, we will first describe previous works in both sentence-level and word-level autocompletion, then show the relation to other tasks and scenarios.

Sentence-level Autocompletion Most of previous work in autocompletion for CAT focus on sentence-level completion. A common use case in this line is interactive machine translation (IMT) (Green et al., 2014; Cheng et al., 2016; Peris et al., 2017; Knowles and Koehn, 2016; Santy et al., 2019). IMT systems utilize MT systems to complete the rest of a translation after human translators editing a prefix translation (Alabau et al., 2014; Zhao et al., 2020). For most IMT systems, the core to achieve this completion is prefix-constrained decoding (Wuebker et al., 2016).

Another sentence-level autocompletion method, lexically constrained decoding (LCD) (Hokamp and Liu, 2017; Post and Vilar, 2018), recently attracts lots of attention (Hasler et al., 2018; Susanto et al., 2020; Kajiwara, 2019). Compared with IMT, LCD relaxes the constraints provided by human translators from prefixes to general forms: LCD completes a translation based on some unordered words (i.e., lexical constraints), which are not necessary to be continuous (Hokamp and Liu, 2017; Hu et al., 2019; Dinu et al., 2019; Song et al., 2019). Although it does not need additional training, its inference is typically less efficient compared with the standard NMT. Therefore, other works propose efficient methods (Li et al., 2020; Song et al., 2019) by using lexical constraints in a soft manner rather than a hard manner as in LCD.

Word-level Autocompletion Word-level autocompletion for CAT is less studied than sentence-level autocompletion. Langlais et al. (2000); Santy et al. (2019) consider to complete a target word based on human typed characters and a translation prefix. But they require the target word to be the next word of the translation prefix, which limits its application. In contrast, in our work the proposed word-level autocompletion is more general and can be applied to real-world scenarios such as post-editing (Vasconcellos and León, 1985;

[2]This approach has been implemented into a human-machine interactive translation system TranSmart (Huang et al., 2021) at www.transmart.qq.com.

Green et al., 2013) and LCD, where human translators need to input some words (corrections or constraints). Huang et al. (2015) propose a method to predict a target word based on human typed characters, however, this method only uses the source side information and does not consider translation context, leading to limited performance compared with our work.

Others Our work may also be related to previous works in input method editors (IME) (Huang et al., 2018; Lee et al., 2007). However, they are in the monolingual setting and not capable of using the useful multilingual information.

3 Task and Benchmark

In this section, we first describe why we need word-level autocompletion in real-world CAT scenarios. We then present the details of the GWLAN task and the construction of benchmark.

Why GWLAN? Word level autocompletion is beneficial for improving input efficiency (Langlais et al., 2000). Previous works assume that the translation context should be a prefix and the desired word is next to the prefix as shown in Figure 1 (b), where the context is "We asked two" and the desired word is "specialists". However, in some real-world CAT scenarios such as post-editing and lexically constrained decoding, translation context may be discontinuous and the input words (corrections or lexical constraints) are not necessarily conjunct to the translation context. As shown in Figure 1 (c), the context is "We $\cdots$ their opinion" and the human typed characters "sp" is conjunct to neither "We" nor "their" in the context. Therefore, existing methods can not perform well on such a general scenario. This motivates us to propose a general word-level autocompletion task for CAT.

3.1 Task Definition

Suppose $\boldsymbol{x} = (x_1, x_2, \dots, x_m)$ is a source sequence, $\boldsymbol{s} = (s_1, s_2, \dots, s_k)$ is a sequence of human typed characters, and a translation context is denoted by $\boldsymbol{c} = (\boldsymbol{c}_l, \boldsymbol{c}_r)$, where $\boldsymbol{c}_l = (c_{l,1}, c_{l,2}, \dots, c_{l,i})$, $\boldsymbol{c}_r = (c_{r,1}, c_{r,2}, \dots, c_{r,j})$. The translation pieces $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$ are on the left and right hand side of $\boldsymbol{s}$, respectively. Formally, given a source sequence $\boldsymbol{x}$, typed character sequence $\boldsymbol{s}$ and a context $\boldsymbol{c}$, the *general word-level autocompletion* (GWLAN) task aims to predict a target word w which is to be placed in the middle between $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$ to constitute a partial translation. Note that in the partial translation consisting of $\boldsymbol{c}_l$, w and $\boldsymbol{c}_r$, w is not necessary to be consecutive to $c_{l,i}$ or $c_{r,1}$. For example, in Figure 1 (c), $\boldsymbol{c}_l$ = ("We",), $\boldsymbol{c}_r$ = ("their", "option", "."), $\boldsymbol{s}$ = ("sp",), the GWLAN task is expected to predict w = "specialists" to constitute a partial translation "We $\cdots$ specialists $\cdots$ their opinion.", where "$\cdots$" represents zero, one, or more words (i.e., the two words before and after it are not necessarily consecutive).

To make our task more general in real-world scenarios, we assume that the left context $\boldsymbol{c}_l$ and right context $\boldsymbol{c}_r$ can be empty, which leads to the following four types of context:

- Zero-context: both $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$ are empty;
- Suffix: $\boldsymbol{c}_l$ is empty;
- Prefix: $\boldsymbol{c}_r$ is empty;
- Bi-context: neither $\boldsymbol{c}_l$ nor $\boldsymbol{c}_r$ is empty.

With the tuple $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{c})$, the GWLAN task is to predict the human desired word w.

Relation to most similar tasks Some similar techniques have been explored in CAT. Green et al. (2014) and Knowles and Koehn (2016) studied a autocompletion scenario called translation prediction (TP), which provides suggestions of the next word (or phrase) given a prefix. Besides the strict assumption of translation context (i.e., prefix here), compared with GWLAN, another difference is that the information of human typed characters is ignored in their setting. There also exist some works that consider the human typed sequences (Huang et al., 2015; Santy et al., 2019), but they only consider a specific type of translation contexts. Huang et al. (2015) propose to complete a target word based on the zero-context assumption. Despite its flexibility, this method is unable to explore translation contexts to improve the autocompletion performance. The word-level autocompletion methods in Langlais et al. (2000); Santy et al. (2019) have the same assumption as TP, which impedes the use of their methods under the scenarios like post editing and lexically constrained decoding, where human inputs are not necessarily conjunct to the variety of translation contexts.

3.2 Benchmark Construction

To set up a benchmark, firstly we should create a large scale dataset including tuples of $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{c}, w)$ for training and evaluating GWLAN models. Ideally, we may hire professional translators to manually annotate such a dataset, but it is too costly in practice. Therefore, in this work, we propose to automatically construct the dataset from parallel datasets which is originally used in automatic machine translation tasks. The procedure for constructing our data is the same for train, validation, and test sets. And we construct a dataset for each type of translation context.

Assume we are given a parallel dataset $\{(\boldsymbol{x}^i, \boldsymbol{y}^i)\}$, where $\boldsymbol{y}^i$ is the reference translation of $\boldsymbol{x}^i$. Then, we can automatically construct the data $\boldsymbol{c}^i$ and $\boldsymbol{s}^i$ by randomly sampling from $\boldsymbol{y}^i$. We first sample a word $w = \boldsymbol{y}^i_k$ and then demonstrate how to extract $\boldsymbol{c}^i$ for different translation contexts:

- Zero-context: both $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$ are empty;
- Suffix: randomly sample a translation piece $\boldsymbol{c}_r = \boldsymbol{y}_{p_{r,1}:p_{r,2}}$ from $\boldsymbol{y}$, where $k < p_{r,1} < p_{r,2}$. The $\boldsymbol{c}_l$ is empty here;
- Prefix: randomly sample a translation piece $\boldsymbol{c}_l = \boldsymbol{y}_{p_{l,1}:p_{l,2}}$ from $\boldsymbol{y}$, where $p_{l,1} < p_{l,2} < k$. The $\boldsymbol{c}_r$ is empty here;
- Bi-context: sample $\boldsymbol{c}_l$ as in prefix, and sample $\boldsymbol{c}_r$ as in suffix.

Then we have to simulate the human typed characters $\boldsymbol{s}$ based on w. For languages like English and German, we sample a position p from the character sequence and the human input $\boldsymbol{s} = w_{1:p}$, where $1 \leq p < L_w$. For languages like Chinese, the human input is the phonetic symbols of the word, since the word cannot be directly typed into the computer. Therefore, we have to convert w to phonetic symbols that are characters in alphabet and sample $\boldsymbol{s}$ from phonetic symbols like we did on English.

Evaluation Metric To evaluate the performance of the well-trained models, we choose accuracy as the evaluation metric:

$$\text{Acc} = \frac{N_{match}}{N_{all}}, \quad (1)$$

where N_{match} is the number of words that are correctly predicted and N_{all} is the number of testing examples.

4 Proposed Approach

Given a tuple $(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{s})$, our approach decomposes the whole word autocompletion process into two parts: model the distribution of the target word w based on the source sequence $\boldsymbol{x}$ and the translation context $\boldsymbol{c}$, and find the most possible word w based on the distribution and human typed sequence $\boldsymbol{s}$.

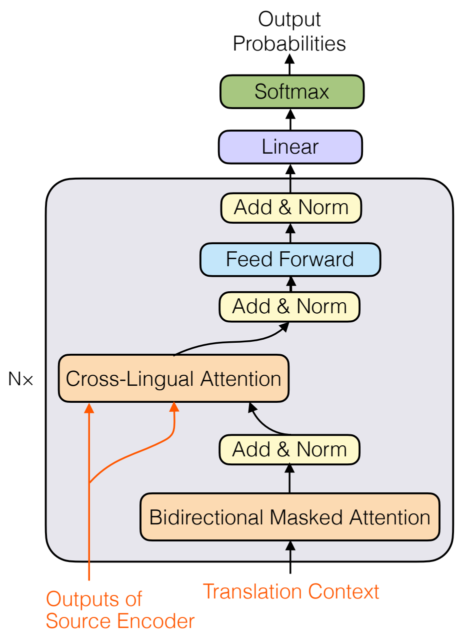

Figure 2: Cross-lingual encoder of the WPM.

Therefore, in the following subsections, we firstly propose a word prediction model (WPM) to define the distribution $p(w|\boldsymbol{x}, \boldsymbol{c})$ of the target word w (§4.1). Then we can treat the human input sequence $\boldsymbol{s}$ as soft constraints or hard constraints to complete $\boldsymbol{s}$ and obtain the target word w (§4.2). Finally, we present two strategies for training and inference (§4.3).

4.1 Word Prediction Model

The purpose of WPM is to model the distribution $p(w|\boldsymbol{x}, \boldsymbol{c})$. More concretely, we will use a single placeholder `[MASK]` to represent the unknown target word w, and use the representation of `[MASK]` learned from WPM to predict it. Formally, given the source sequence $\boldsymbol{x}$, and the translation context $\boldsymbol{c} = (\boldsymbol{c}_l, \boldsymbol{c}_r)$, the possibility of the target word w is:

$$P(w|\boldsymbol{x}, \boldsymbol{c}_l, \boldsymbol{c}_r; \theta) = \text{softmax}\left(\phi(h)\right)[w] \quad (2)$$

where h is the representation of `[MASK]`, ϕ is a linear network that projects the hidden representation h to a vector with dimension of target vocabulary size V, and $\text{softmax}(d)[w]$ takes the component regarding to w after the softmax operation over a vector $d \in \mathbb{R}^V$.

Inspired by the attention-based architectures (Vaswani et al., 2017; Devlin et al., 2019)[3], we

[3]Because the use of attention-based models has become ubiquitous recently, we omit an exhaustive background de-

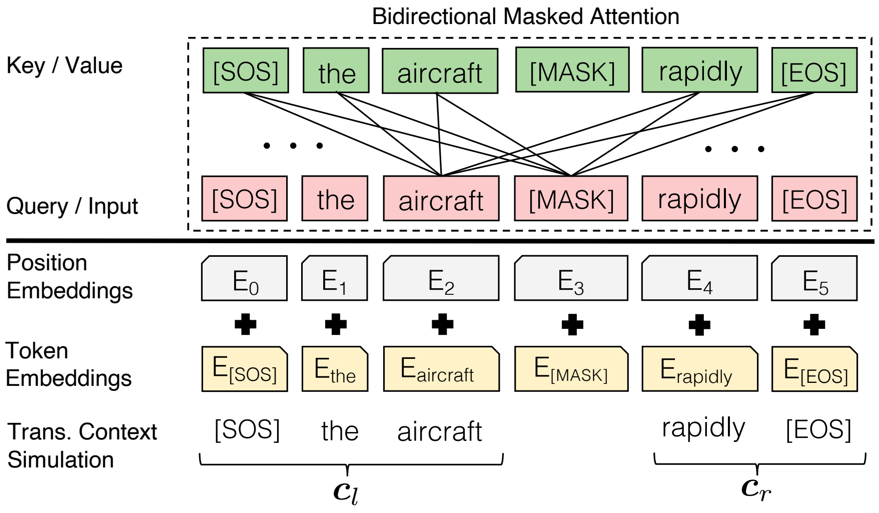

Figure 3: The input representation of our model and architecture of Bidirectional Masked Attention. The input embeddings are the sum of the token embeddings and position embeddings. `[MASK]` represents the potenial target word in this translation context.

use a dual-encoder architecture to learn the representation h based on source sequence $\boldsymbol{x}$ and translation context $\boldsymbol{c}$. Our model has a source encoder and a cross-lingual encoder. The source encoder of WPM is the same as the Transformer encoder, which is used to encode the source sequence $\boldsymbol{x}$. As shown in Figure 2, the output of source encoder will be passed to the cross-lingual encoder later. The cross-lingual encoder is similar to the Transformer decoder, while the only difference is that we replace the auto-regressive attention (ARA) layer by a bidirectional masked attention (BMA) module, due to that the ARA layer cannot use the leftward information flow (i.e., $\boldsymbol{c}_r$).

Specifically, the BMA module is built by a multiple-layer self attention network. As shown in Figure 3, in each layer of BMA, each token in the attention query can attend to all words in translation context $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$. In addition, the input consists of three parts, the `[MASK]` token, and translation contexts $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$, as illustrated in Figure 3. Note that its position embeddings `E` are only used to represent the relative position relationship between tokens. Taking the sentence in Figure 3 as an example, $\mathtt{E}_3$ does not precisely specify the position of the target word w but roughly indicates that w is on the right-hand-side of $\boldsymbol{c}_l$ and on the left-hand-side of $\boldsymbol{c}_r$. Finally, the representation of `[MASK]` as learnt by BMA will be passed to `Add & Norm` layer as shown in Figure 2.

scription of the model and refer readers to Vaswani et al. (2017) and Devlin et al. (2019).

4.2 Human Input Autocompletion

After learning the representation h of the `[MASK]` token, there are two ways to use the human input sequence $\boldsymbol{s}$ to determinate the human desired word. Firstly, we can learn the representation of $\boldsymbol{s}$ and use it as a soft constraint while predicting word w. Taking the sentence in Figure 3 as an example, supposing the human typed sequence is $\boldsymbol{s}$ = "`des`", we can use an RNN network to learn the representation of `des` and concatenate it with h to predict the word `descending`. Alternatively, we can use `des` as a hard constraint:

$$P_{\boldsymbol{s}}[w] = \begin{cases} \frac{P(w|\boldsymbol{x},\boldsymbol{c};\theta)}{Z}, & \text{if } w \text{ starts with } \boldsymbol{s} \\ 0, & \text{otherwise.} \end{cases}$$

where $P(\cdot|\cdot)$ is the probability distribution defined in Eq. (2) and Z is the normalization term independent on w. Then we pick $w^* = \arg\max_w P_{\boldsymbol{s}}[w]$ as the most possible word. In our preliminary experiments, the performances of these two methods are comparable, and there is no significant gain when we use them together. One main reason is that the model can already learn the starts-with action precisely in the soft constraint method. Therefore, we propose to use the human inputs as hard constraints in our later experiments, because of the method's efficiency and simplicity.

4.3 Training and Inference Strategy

Suppose $\mathcal{D}$ denotes the training data for GWLAN, i.e., a set of tuples $(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{s}, w)$. Since there are four different types of context in $\mathcal{D}$ as presented in

§3, we can split $\mathcal{D}$ into four subsets $\mathcal{D}_{\text{zero}}$, $\mathcal{D}_{\text{prefix}}$, $\mathcal{D}_{\text{suffix}}$ and $\mathcal{D}_{\text{bi}}$. To yield good performances on those four types of translation context, we also propose two training strategies. The inference strategy differs accordingly.

Strategy 1: One Context Type One Model For this strategy, we will train a model for each translation context, respectively. Specifically, for each type of context $t \in \{\text{zero}, \text{prefix}, \text{suffix}, \text{bi}\}$, we independently train one model θ_t by minimizing the following loss $\mathcal{L}(\mathcal{D}_t, \theta)$:

$$\mathcal{L}(\mathcal{D}_t;\theta) = \frac{1}{|\mathcal{D}_t|} \sum_{(\boldsymbol{x},\boldsymbol{c},\boldsymbol{s},w)\in\mathcal{D}_t} \log P(w|\boldsymbol{x},\boldsymbol{c};\theta), \tag{3}$$

where $P(w|\boldsymbol{x},\boldsymbol{c};\theta)$ is the WPM model defined in Eq. 2, $|\mathcal{D}_t|$ is the size of training dataset $\mathcal{D}_t$, and t can be any type of translation context. In this way, we actually obtain four models in total after training. In the inference process, for each testing instance $(\boldsymbol{x}, \boldsymbol{c}_l, \boldsymbol{c}_r, \boldsymbol{s})$, we decide its context type t in terms of $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$ and then use $\hat{\theta}_t$ to predict the word w.

Strategy 2: Joint Model The separate training strategy is straightforward. However, it may also make the models struck in the local optimal. To address these issues, we also propose a joint training strategy, which has the ability to stretch the model out of the local optimal once the parameters is over-fitting on one particular translation context. Therefore, using the joint training strategy, we train a single model for all types of translation context by minimizing the following objective:

$$\mathcal{L}(\mathcal{D};\hat{\theta}) = \mathcal{L}(\mathcal{D}_{\text{zero}};\hat{\theta}) + \mathcal{L}(\mathcal{D}_{\text{prefix}};\hat{\theta}) + \mathcal{L}(\mathcal{D}_{\text{suffix}};\hat{\theta}) + \mathcal{L}(\mathcal{D}_{\text{bi}};\hat{\theta})$$

where each $\mathcal{L}(\mathcal{D}_t;\theta)$ is as defined in Eq. 3. In this way, we actually obtain a single model $\hat{\theta}$ after training. In the inference process, for each testing instance $(\boldsymbol{x}, \boldsymbol{c}_l, \boldsymbol{c}_r, \boldsymbol{s})$ we always use $\hat{\theta}$ to predict the target word w.

5 Experiments

5.1 Datasets

We carry out experiments on four GWLAN tasks including bidirectional Chinese–English tasks and German–English tasks. The benchmarks for our experiments are based on the public translation datasets. The training set for two directional Chinese–English tasks consists of 1.25M bilingual sentence pairs from LDC corpora. The toolkit we used to convert Chinese word w to phonetic symbols is pypinyin[4]. As discussed in (§3.2), the training data for GWLAN is extracted from 1.25M sentence pairs. The validation data for GWLAN is extracted from NIST02 and the test datasets for GWLAN are constructed from NIST05 and NIST06. For two directional German–English tasks, we use the WMT14 dataset preprocessed by Stanford[5]. The validation and test sets for our tasks are based on newstest13 and newstest14 respectively. For each dataset, the models are tuned and selected based on the validation set.

The main strategies we used to prepare our benchmarks are shown in §3.2. However, lots of trivial instances may be included if we directly use the uniform distribution for sampling, e.g., predicting word "the" given "th". Therefore, we apply some intuitive rules to reduce the probability of trivial instances. For example, we assign higher probability for words with more than 4 characters in English and 2 characters in Chinese, and we require that the lengths of input character sequence $\boldsymbol{s}$ and translation contexts $\boldsymbol{c}$ should not be too long.

5.2 Systems for Comparison

In the experiments, we evaluate and compare the performance of our methods (WPM-Sep and WPM-Joint) and a few baselines. They are illustrated below,

WPM-SEP is our approach with the "one context one model" training and inference strategy in Section §4.3. In other words, we train our model for each translation context separately.

WPM-JOINT is our approach with the "joint model" strategy in Section §4.3.

TRANSTABLE: We train an alignment model[6] on the training set and build a word-level translation table. While testing, we can find the translations of all source words based on this table, and select out valid translations based on the human input. The word with highest frequency among all candidates is regarded as the prediction. This baseline is inspired by Huang et al. (2015).

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

[5] https://nlp.stanford.edu/projects/nmt/

[6] https://github.com/clab/fast_align

#	Systems	Zh⇒En		En⇒Zh		De⇒En		En⇒De	
		NIST05	NIST06	NIST05	NIST06	NT13	NT14	NT13	NT14
1	TRANSTABLE	41.40	39.78	28.00	26.99	37.43	36.64	32.99	31.12
2	TRANS-PE	34.51	35.50	32.23	34.88	34.45	33.02	31.51	30.65
3	TRANS-NPE	35.97	36.78	34.31	36.19	36.69	36.01	33.25	31.30
4	WPM-SEP	54.15	55.04	53.30	53.67	56.93	55.67	54.54	51.46
5	WPM-JOINT	**55.54**	**55.85**	**53.64**	**54.25**	**57.84**	**56.75**	**56.91**	**52.68**

Table 1: The main results of different systems on Chinese-English and German-English datasets. The results in this table are the averaged accuracy on four translation contexts (i.e., prefix, suffix, zero-context, and bi-context).

#	Systems	Zh⇒En					En⇒Zh				
		Prefix	Suffix	Zero	Bi	Avg.	Prefix	Suffix	Zero	Bi	Avg.
1	TRANSTABLE	41.91	44.99	44.19	43.28	43.59	29.73	32.80	29.73	29.61	30.46
2	TRANS-PE	29.84	38.61	26.08	48.06	35.64	30.64	34.97	22.67	38.95	31.80
3	TRANS-NPE	37.36	40.43	29.50	44.42	37.92	36.10	43.05	32.00	45.79	39.23
4	WPM-SEP	58.43	60.59	53.99	**64.46**	59.36	60.02	61.05	53.76	**64.46**	59.82
5	WPM-JOINT	**59.91**	**60.71**	**55.35**	62.30	**59.56**	**61.39**	**61.73**	**53.87**	63.78	**60.19**

Table 2: The results of different systems on NIST02. We evaluate the performances of those systems on both Zh⇒En and En⇒Zh tasks by accuracy.

TRANS-PE: We train a vanilla NMT model using the Transformer-base model. During the inference process, we use the context on the left hand side of human input as the model input, and return the most possible words based on the probability of valid words selected out by the human input. This baseline is inspired by Langlais et al. (2000); Santy et al. (2019).

TRANS-NPE: As another baseline, we also train an NMT model based on Transformer, but without position encoding on the target side. While testing, we use the averaged hidden vectors of all the target words outputted by the last decoder layer to predict the potential candidates.

5.3 Main Results

Table 1 shows the main results of our methods and three baselines on the test sets of Chinese-English and German-English datasets. It is clear from the results that our methods WPM-SEP and WPM-JOINT significantly outperform the three baseline methods. Results on Row 4 and Row 5 of Table 1 also show that the WPM-JOINT method, which uses a joint training strategy to optimize a single model, achieves better overall performance than WPM-SEP, which trains four models for different translation contexts respectively. In-depth analysis about the two training strategies is presented in the next section.

The method TRANS-PE, which assumes the human input is the next word of the given context, behaves poorly under the more general setting. As the results of TRANS-NPE show, when we use the same model as TRANS-PE and relax the constraint of position by removing the position encoding, the accuracy of the model improves. One interesting finding is that the TRANSTABLE method, which is only capable of leveraging the zero-context, achieves good results on the Chinese-English task when the target language is English. However, when the target language is Chinese, the performance of TRANSTABLE drops significantly.

6 Experimental Analysis

6.1 Effects on Different Translation Context

In this section, we presents more detailed results on the four translation contexts and analyze the features of GWLAN. These analyses can help us to better understand the task and propose effective approaches in the future.

Separate Training VS. Joint Training Compared with WPM-SEP, WPM-JOINT shows two advantages. On one hand, even there is only one model, WPM-JOINT yields better performances than WPM-SEP, enabling simpler deployment. This may be caused by that training on multiple related tasks can force the model learn more expressive representations, avoiding over-fitting. On the other hand, the variance of results on different translation contexts of WPM-JOINT is smaller, which can provide an more steady autocompletion service. From the viewpoint of joint training, the lower variance may be caused by that WPM-JOINT spends more efforts to minimize the one with maximal risk (i.e., zero-context), although sometimes it may slightly sacrifice the task with minimal risk (i.e., bi-context).

The results of WPM-SEP and WPM-JOINT also have some shared patterns. Firstly, the performances of the two methods on prefix and suffix translation contexts are nearly the same. Although the prefix and suffix may play different roles in the SVO language structure, they have little impact on the the autocompletion accuracy using our method. Moreover, among the results on four translation contexts, the performances on bi-context are better than prefix and suffix, and prefix and suffix are better than zero-context. This finding shows that more context information can help to reduce the uncertainty of human desired words.

Comparison with baselines The TRANS-PE method in previous works is more sensitive to the position of human input. The statistical results shows that the averaged distances in the original sentence between the prediction words and translation contexts are various for different translation contexts, which are 7.4, 6.5, 14.1, and 3.2 for prefix, suffix, zero-context, and bi-context, respectively. When the desired words are much closer to the context, TRANS-PE can achieve better performances. Moreover, TRANS-PE can achieve more than 80 accuracy scores when the prediction word is the next word of the given prefix, however, its performance drops significantly when the word is not necessarily conjunct to the prefix. We can also find that TRANS-NPE, which removes the position information of target words, achieves better overall performances compared with TRANS-PE.

In contrast, the performance of TRANSTABLE is less affected by the position of the prediction words, which is demonstrated by the low variances on both tasks in Table 2. The results of TRANSTABLE have also surprised us, which achieves more than 41 accuracy scores on the Zh⇒En task. This observation shows the importance of alignment and the potential of statistical models. Compared with the results on the Zh⇒En task, the overall accuracy on En⇒Zh task is much lower, likely due to that the number of valid words after filtered by the human input on Chinese is much more than that on English. Therefore, it is easier for TRANSTABLE to determine the human desired words in English.

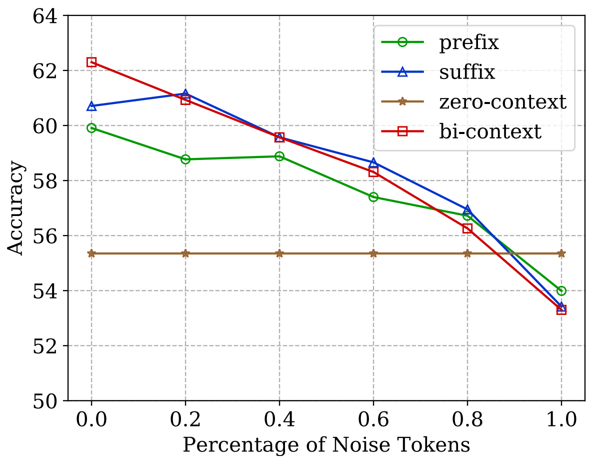

Figure 4: Robustness Analysis. The x-axis represents the percentage of words that have been replaced by noise tokens in NIST02. The model used for this analysis is the WPM-JOINT, which is trained on the Zh⇒En task without noisy translation context.

6.2 Robustness on Noisy Contexts

In this work, the translation contexts are simulated using the references. However, in real-world scenarios, translation contexts may not be perfect, i.e., some words in the translation contexts may be incorrect. In this section, we evaluate the robustness of our model on noisy contexts. We first use the translation table constructed by TRANSTABLE to find some target words that share the same source words with the original target words, and then use those found words as noise tokens.

The robustness results are shown in Figure 4. For all the translation context types except for zero-context, the performance drops slowly when the percentage of noise tokens increases. However, even with 80% words in the context, the performance of WPM-JOINT outperforms the case of zero-context, which shows that our WPM-JOINT method is noise tolerant.

6.3 Discussion

In this work, we formalize the task as a classification problem. However, the generation formalization also deserves to be explored in the future. For example, the generation may happen in two circumstances: word-level completion based on subwords, and phrase-level completion. In the first case, although the autocompletion service provided for human translators is word-level, in the internal system we can generate a sequence of subwords (Sennrich et al., 2015) that satisfy the human typed characters, and provide human translators with the merged subwords. This subword sequence generation can significantly alleviate the OOV issue in the word-level autocompletion. In the phrase-level autocompletion case, if we can predict more than one desired words, the translation efficiency and experience may be improved further. We would like to leave it as future work.

It is also worth noting that we did not conduct human studies in this work. We think evidences in previous work can already prove the effectiveness of word-level autocompletion when assisting human translators. For example, TransType (Langlais et al., 2000) is a simple rule-based tool that only considers the prefix context, but the majority of translators said that TransType improved their typing speed a lot. Huang et al. (2015) hired 12 professional translators and systematically evaluate their word autocompletion tool based on zero-context. Experiments show that the more keystrokes are reduced, the more time can be saved for translators. Since the prediction accuracy is highly correlated with the keystrokes, we think higher accuracy will make translators more productive. That is the main reason that we use accuracy to automatically evaluate the model performance. Besides, the automatic evaluation metric also makes the GWLAN task easier to follow.

7 Conclusion

We propose a General Word-Level AutocompletioN (GWLAN) task for computer-aided translation (CAT). In our setting, we relax the strict constraints on the translation contexts in previous work, and abstract four most general translation contexts used in real-world CAT scenarios. We propose two approaches to address the variety of context types and weak position information issues in GWLAN. To support automatic evaluation and to ensure a convenient and fair comparison among different methods, we construct a benchmark for the task. Experiments on this benchmark show that our method outperforms baseline methods by a large margin on four datasets. We believe that this benchmark to be released will push forward future research in CAT.

Acknowledgments

We would like to thank three anonymous reviewers for their invaluable discussions on this work. The corresponding is Lemao Liu.

References

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 Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 25–28.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

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 (CL)*, 35(1):3–28.

Shanbo Cheng, Shujian Huang, Huadong Chen, Xinyu Dai, and Jiajun Chen. 2016. Primt: A pick-revise framework for interactive machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1240–1249.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 263–270.

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3063–3068.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1243–1252.

Spence Green, Jeffrey Heer, and Christopher D Manning. 2013. The efficacy of human post-editing for language translation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 439–448.

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 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1225–1236.

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 506–512.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1535–1546.

J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved lexically constrained decoding for translation and monolingual rewriting. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 839–850.

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*.

Guoping Huang, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2015. A new input method for human translators: integrating machine translation effectively and imperceptibly. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1163–1169.

Yafang Huang, Zuchao Li, Zhuosheng Zhang, and Hai Zhao. 2018. Moon ime: neural-based chinese pinyin aided input method with customizable association. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL): System Demonstrations*, pages 140–145.

Tomoyuki Kajiwara. 2019. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6047–6052.

Rebecca Knowles and Philipp Koehn. 2016. Neural interactive translation prediction. In *Proceedings of the Association for Machine Translation in the Americas (AMTA)*, pages 107–120.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 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*.

Kai-fu Lee, Zheng Chen, and Jian Han. 2007. Language input architecture for converting one text form to another text form with modeless entry. US Patent 7,165,019.

Huayang Li, Guoping Huang, Deng Cai, and Lemao Liu. 2020. Neural machine translation with noisy lexical constraints. *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 28:1864–1874.

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

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1314–1324.

Sebastin Santy, Sandipan Dandapat, Monojit Choudhury, and Kalika Bali. 2019. INMT: Interactive neural machine translation prediction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 103–108.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1715–1725.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 449–459.

Raymond Hendy Susanto, Shamil Chollampatt, and Liling Tan. 2020. Lexically constrained neural machine translation with levenshtein transformer. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3536–3543.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3104–3112.

Muriel Vasconcellos and Marjorie León. 1985. Spanam and engspan: machine translation at the pan american health organization. *Computational Linguistics (CL)*, 11(2-3):122–136.

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 (NeurIPS)*, pages 5998–6008.

Qian Wang, Jiajun Zhang, Lemao Liu, Guoping Huang, and Chengqing Zong. 2020. Touch editing: A flexible one-time interaction approach for translation. In *Proceedings of the Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (AACL-IJCNLP)*, pages 1–11.

Rongxiang Weng, Hao Zhou, Shujian Huang, Lei Li, Yifan Xia, and Jiajun Chen. 2019. Correct-and-memorize: Learning to translate from interactive revisions. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5255–5263.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Joern Wuebker, Spence Green, John DeNero, Saša Hasan, and Minh-Thang Luong. 2016. Models and inference for prefix-constrained machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 66–75.

Tianxiang Zhao, Lemao Liu, Guoping Huang, Zhaopeng Tu, Huayang Li, Yingling Liu, Guiquan Liu, and Shuming Shi. 2020. Balancing quality and human involvement: An effective approach to interactive neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 9660–9667.