WeTS: A Benchmark for Translation Suggestion

Anonymous ACL submission

Abstract

Translation suggestion (TS), which provides alternatives for specific words or phrases given the entire documents generated by machine translation (MT), has been proven to play a significant role in post-editing (PE). There are two main pitfalls for previous researches in this line. First, most conventional works only focus on the overall performance of PE but ignore the exact performance of TS, which makes the progress of PE sluggish and less explainable; Second, as no publicly available golden dataset to support in-depth research for TS, almost all of the previous works conduct experiments on their in-house datasets or the noisy datasets built automatically, which makes their experiments hard to be reproduced and compared. To break these limitations mentioned above and spur the research in TS, we create a benchmark dataset, called *WeTS*, which is a golden corpus annotated by expert translators on four translation directions. Apart from the golden corpus, we also propose several methods to generate synthetic corpus which can be used to improve the performance substantially through pre-training. As for the model, we propose the segment-aware self-attention based Transformer for TS. Experimental results show that our approach achieves state-of-the-art results on all four directions, including English-to-German, German-to-English, Chinese-to-English, and English-to-Chinese.[1]

1 Introduction

Computer-aided translation (CAT) (Barrachina et al., 2009; Green et al., 2014; Knowles and Koehn, 2016; Santy et al., 2019) has attained more and more attention for its promising ability in combining the high efficiency of machine translation (MT) (Cho et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017) and high accuracy of human translation (HT). A typical way for CAT tools to combine MT and HT is PE (Green et al., 2013; Zouhar et al., 2021), where the human translators are asked to provide alternatives for the incorrect word spans in the results generated by MT. To further reduce the post-editing time, researchers propose to apply TS into PE, where TS provides the sub-segment suggestions for the annotated incorrect word spans in the results of MT, and their extensive experiments show that TS can substantially reduce translators' cognitive loads and the post-editing time (Wang et al., 2020; Lee et al., 2021).

As there is no explicit and formal definition for TS, we observe that some previous works similar or related to TS have been proposed (Alabau et al., 2014; Santy et al., 2019; Wang et al., 2020; Lee et al., 2021). However, there are two main pitfalls for these works in this line. First, most conventional works only focus on the overall performance of PE but ignore the exact performance of TS. This is mainly because that the golden corpus for TS is relatively hard to collect. As TS is an important sub-module in PE, paying more attention to the exact performance of TS can boost the performance and interpretability of PE. Second, almost all of the previous works conduct experiments on their in-house datasets or the noisy datasets built automatically, which makes their experiments hard to be followed and compared. Additionally, experimental results on the noisy datasets may not truly reflect the model's ability on generating the right predictions in practical test scene, making the research deviate from the correct direction. Therefore, the community is in dire need of a benchmark for TS to enhance the research in this area.

To address the limitations mentioned above and spur the research in TS, we make our efforts to construct a high-quality benchmark dataset with human annotation, named *WeTS*,[2] which can truly reflect the model's ability in real application scenario.

[1] For reviewers, codes and corpus can be found in the attached files, and we will make them publicly available after the double-blind phase.

[2] *WeTS*: We Establish a benchmark for Translation Suggestion

Considering collecting the golden dataset is expensive and labor-consuming, we further propose several methods to automatically construct synthetic corpus, which can be utilized to improve the TS performance through pre-training. As for the model, we for the first time propose the segment-aware self-attention based Transformer for TS, named SA-Transformer, which achieves superior performance to the naive Transformer (Vaswani et al., 2017). To conclude, the main contributions of this paper are summarized as follows:

- We construct and share a benchmark dataset for TS in four translation directions, including English-to-German, German-to-English, Chinese-to-English, and English-to-Chinese.
- We give formal definitions for TS and further divide TS finely into two sub-tasks, namely *naive TS* and *TS with hints*, according to whether the translators' hints are considered.
- We provide strong baseline models for this community. Specifically, we make a detailed comparison between the Transformer-based and XLM-based models, and propose the segment-aware self-attention based Transformer for TS, which achieves State-Of-The-Art (SOTA) results on the benchmark dataset.
- We thoroughly investigate different ways for building the synthetic corpus. Since constructing the golden corpus is expensive and labor-consuming, it is very essential and promising to build the synthetic corpus by making full use of the parallel corpus of MT.
- We conduct extensive experiments and provide deep analyses about the strengths and weaknesses of the proposed approach, which are expected to give some insights for further researches on TS.

2 WeTS

In this section, we will introduce the proposed benchmark dataset *WeTS*. To make the process of constructing *WeTS* understood easily, we first formally define the two sub-tasks of TS.

2.1 Task

We finely divide the task of TS into two sub-tasks, namely *vanilla TS* and *TS with hints*, according to whether the translators' hints are considered.

Vanilla TS. Given the source sentence $\boldsymbol{x} = (x_1, \ldots, x_s)$, the translation sentence $\boldsymbol{m} = (m_1, \ldots, m_t)$, the incorrect words or phrases $\boldsymbol{w} = \boldsymbol{m}_{i:j}$ where $1 \leq i \leq j \leq t$, and the correct alternative $\boldsymbol{y}$ for $\boldsymbol{w}$, the task of *vanilla TS* is optimized to maximize the conditional probability of $\boldsymbol{y}$ as follows:

$$P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{m}^{-\boldsymbol{w}}, \theta) \quad (1)$$

where θ represents the model parameter, and $\boldsymbol{m}^{-\boldsymbol{w}}$ is the masked translation where the incorrect word span $\boldsymbol{w}$ is replaced with a placeholder. [3]

TS with Hints. In the sub-task *TS with hints*, the hints of translators are considered as some soft constraints for the model, and the model is expected to generate suggestions meeting these constraints. The format of the translator's hint is very flexible, which usually requires only a few types on the keyboard by the translator. For English and German, the hints can be the character sequence which includes the initials of words in the correct alternative. As for Chinese, the hints can be the character sequence which includes the initials of the phonetics of words in the correct alternative. In this setting, the model is optimized as:

$$P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{m}^{-\boldsymbol{w}}, \boldsymbol{h}, \theta) \quad (2)$$

where $\boldsymbol{h}$ indicates the hints provided by translators.

Translation Direction	Train	Valid	Test
En⇒De	14,957	1000	1000
De⇒En	11,777	1000	1000
Zh⇒En	21,213	1000	1000
En⇒Zh	15,769	1000	1000

Table 1: The statistics about *WeTS*. "En⇒De" refers to the translation direction of English-to-German, and "En⇒Zh" refers to the direction of English-to-Chinese.

2.2 Dataset

This sub-section describes the construction of *WeTS*, which serves as a benchmark dataset for TS. *WeTS* is a golden corpus for four different translation directions, including English-to-German, German-to-English, Chinese-to-English, and English-to-Chinese. All samples in *WeTS* are annotated by expert translators.

As the starting point, we collect the monolingual corpus for English and German from the raw

[3] $\boldsymbol{w}$ is null if i equals j, and the model will predict whether some words need to be inserted in position i.

Source Sentence	他们也许并不知道这是一个“假理财”骗局，但也察觉到了诸多可疑之处，然而最终还是按照张颖的指使进行了违法违规操作。
Translation	They may not know this is a “fake financial management” scam, but also aware of many **suspicious**, and ultimately conduct illegal operations according to Zhang Ying’s instructions.
Suggestions	1. suspects **(s)**; 2. doubtful points **(d p)**; 3. questionable points **(q p)**

Figure 1: One training example in *WeTS*. For the incorrect word "suspicious" (in red color) in the translation, there are three correct alternatives and the corresponding hints, i.e., the character sequence in the bracket (in blue color).

Wikipedia dumps, and extract Chinese monolingual corpus from various online news publications. We first clean the monolingual corpus with a language detector to remove sentences belonging to other languages. For all monolingual corpus, we remove sentences that are shorter than 20 words or longer than 80 words. In addition, sentences which exist in the available parallel corpus are also removed. Then, we get the translations by feeding the cleaned monolingual corpus into the corresponding well-trained NMT model.[4] Finally, the translators are required to mark the incorrect word spans in the translation sentence and provide correct alternatives, based on the source sentence and its translation. The core rule for the translator is annotating the incorrect span as local as possible, as generating correct alternatives for long sequences is much harder than that of shorter sequences.

During annotating, we mainly focus on the following three kinds of errors: 1) Under-translation or over-translation: While the problem of under-translation or over-translation has been alleviated with the popularity of Transformer, it is still one of the main mistakes in NMT and seriously destroys the readability of the translation. 2) Semantic errors: For the semantic error, we mean that some source words are incorrectly translated according to the semantic context, such as the incorrect translations for entities, prop nouns, and ambiguous words. Another branch of semantic mistake is that the source words or phrases are only translated superficially and the semantics behind are not translated well. 3) Grammatical or syntactic errors: Such errors usually appear in translations of long sentences, including the improper use of tenses, passive voice, syntactic structures, etc.

All of the annotated corpora are cross-validated to ensure the accuracy rate above 95%. After annotation, we generate the hints of the correct alternatives automatically.[5] One training example for Chinese to English is presented in Figure 1. The statistics about *WeTS* are presented in Table 1.

3 Construct Synthetic Corpus

Since constructing the golden corpus is expensive and labor-consuming, automatically building the synthetic corpus is very promising for enhancing the performance. In this section, we describe several ways for constructing synthetic corpus for TS based on the parallel corpus of MT and the well-trained MT model.

3.1 Sampling on Golden Parallel Corpus

Sampling on the golden parallel corpus of MT is the most straightforward and simplest way for constructing synthetic corpus for TS. Given the sentence pair $(\boldsymbol{x}, \boldsymbol{r})$ in the parallel corpus of MT, where $\boldsymbol{x}$ is the source sentence and $\boldsymbol{r}$ is the corresponding target sentence, we denote $\boldsymbol{r}^{\backslash i:j}$ as a masked version of $\boldsymbol{r}$ where its fragment from position i to j is replaced with a placeholder ($1 \leq i \leq j \leq |\boldsymbol{r}|$). The $\boldsymbol{r}^{i:j}$ denotes the fragment of $\boldsymbol{r}$ from position i to j. We treat $\boldsymbol{r}^{i:j}$ and $\boldsymbol{r}^{\backslash i:j}$ as the correct alternative ($\boldsymbol{y}$ in Equation 1) and masked translation ($\boldsymbol{m}^{-\boldsymbol{w}}$ in Equation 1) respectively. In this approach, the masked translation in each example is part of the golden target sentence. However, in production, the TS model needs to predict the correct suggestions based on the context of the machine translated sentence. Therefore, the mismatch of distribution between the golden target sentence and machine translated sentence is the potential pitfall for this approach.

[4] We will release the models we utilized here. We train the NMT model for Chinese-English language pairs on the in-house parallel corpus, which contains 80M sentence pairs. The NMT models for English-German language pairs are trained on the parallel corpus of WMT14 English-German.

[5] For Chinese, we apply the tool of pypinyin (https://github.com/mozillazg/python-pinyin) to convert the alternative into its phonetic symbols, i.e., pinyin.

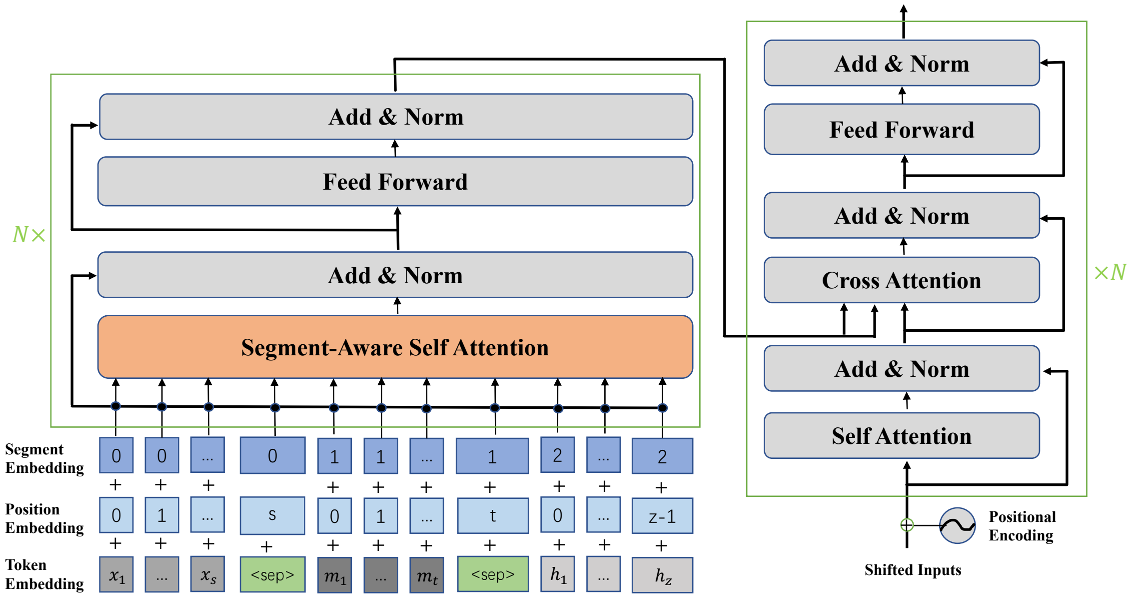

Figure 2: The whole architecture of the proposed SA-Transformer. For generality, the hints are also considered in this example. The s, t, and z are the lengths for $\boldsymbol{x}$, $\boldsymbol{m}^{-\boldsymbol{w}}$ and $\boldsymbol{h}$ respectively.

3.2 Sampling on Pseudo Parallel Corpus

The second approach we apply to construct the synthetic corpus for TS is sampling on the pseudo parallel corpus of MT. Given the source sentence $\boldsymbol{x}$ and the MT model $\boldsymbol{T}_\theta$, we first get the translated sentence $\tilde{\boldsymbol{y}}$ by feeding $\boldsymbol{x}$ into $\boldsymbol{T}_\theta$, and $(\boldsymbol{x}, \tilde{\boldsymbol{y}})$ is treated as the pseudo sentence pair. Then, we perform sampling on $(\boldsymbol{x}, \tilde{\boldsymbol{y}})$ as what we do on $(\boldsymbol{x}, \boldsymbol{r})$ in Section 3.1. Compared to the approach of sampling on the golden parallel corpus, sampling on the pseudo parallel corpus can address the problem of distribution mismatch mentioned in Section 3.1 and it works without relying on the golden parallel corpus. However, the suggested alternatives may be in poor quality since they are parts of the translated sentences.

3.3 Extracting with Word Alignment

Considering the shortcomings of the two previous approaches, we investigate the third approach where we conduct the word alignment between the machine translation and the golden target sentence, and then extract the synthetic corpus for TS based on the alignment information. Given the sentence triple $(\boldsymbol{x}, \tilde{\boldsymbol{y}}, \boldsymbol{r})$, we perform word alignment between $\tilde{\boldsymbol{y}}$ and $\boldsymbol{r}$, and extract the aligned phrase table.[6] For phrase $\tilde{\boldsymbol{y}}^{i:j}$ in $\tilde{\boldsymbol{y}}$ and its aligned phrase $\boldsymbol{r}^{a:b}$ in $\boldsymbol{r}$, we denote $\tilde{\boldsymbol{y}}^{\backslash i:j}_{\boldsymbol{r}^{a:b}}$ as the modified version of $\tilde{\boldsymbol{y}}$ where the phrase $\tilde{\boldsymbol{y}}^{i:j}$ is replaced with $\boldsymbol{r}^{a:b}$. If $\boldsymbol{r}^{a:b}$ is not identical to $\tilde{\boldsymbol{y}}^{i:j}$ and the perplexity of $\tilde{\boldsymbol{y}}^{\backslash i:j}_{\boldsymbol{r}^{a:b}}$ is lower than that of $\tilde{\boldsymbol{y}}$ with a margin no less than β, we treat $\tilde{\boldsymbol{y}}^{\backslash \boldsymbol{i:j}}$ and $\boldsymbol{r}^{a:b}$ as the masked translation and the correct alternative respectively.[7] β is a hyper-parameter to control the threshold of the margin. While this approach has achieved much improvement compared to the previous approaches, we still notice that the errors in the extracted alignment information may introduce some noises into the constructed corpus.

4 The Model

In this section, we describe the proposed model, i.e., SA-Transformer, and the whole architecture is illustrated as Figure 2.

4.1 Inputs

Given the source sentence $\boldsymbol{x}$ and the masked translation $\boldsymbol{m}^{-\boldsymbol{w}}$, the input to the model is formatted as:

$$[\boldsymbol{x}; \langle sep \rangle; \boldsymbol{m}^{-\boldsymbol{w}}] \tag{3}$$

where $[\cdot;\cdot]$ means concatenation, and $\langle sep \rangle$ is a special token used as a delimiter.[8] The position for each segment in the input is calculated independently and we use the segment embedding to

[6] We test two different ways for performing word alignment, including fast-align (Dyer et al., 2013) and TER (Snover et al., 2006), and fast-align performs better.

[7] We use kenlm (https://github.com/kpu/kenlm), the widely used open-source tool for n-gram language model, to measure the sentence perplexity.

[8] If hints are provided, the format for the input is $[\boldsymbol{x}; \langle sep \rangle; \boldsymbol{m}^{-\boldsymbol{w}}; \langle sep \rangle; \boldsymbol{h}]$.

distinguish each segment from others. The representation for each token in the input is the sum of its token embedding, position embedding and segment embedding.

4.2 Segment-aware Self-attention

The naive Transformer applies the self-attention to extract the higher-level information from the token representations in the lower layer without distinguishing tokens in each segment from those in other segments explicitly. The attention matrix in the self-attention is typically calculated as:

$$\frac{\boldsymbol{Q}\boldsymbol{W}^{\boldsymbol{Q}}(\boldsymbol{K}\boldsymbol{W}^{K})}{\sqrt{d_x}} \tag{4}$$

where the $\boldsymbol{Q}$ and $\boldsymbol{K} \in R^{s*d_x}$ are identical in the encoder, $\boldsymbol{W}^{\boldsymbol{Q}}$ and $\boldsymbol{W}^{\boldsymbol{K}} \in R^{d_x*d_x}$ are the projection matrix, d_x is the dimension of the word embedding. However, the inputs for TS contain tokens from different segments, i.e., the source sentence, masked translation, and the hints if provided, and the tokens in each segment are expected to be distinguished from those in other segments since they provide different information for the model's prediction. While the segment embedding in token representations has played the role for distinguishing tokens from different segments, its information has been mixed with the word embedding and diluted with the information flow. With this consideration, we propose the segment-aware self-attention by further injecting the segment information into the self-attention to make it perform differently according to the segment information of the tokens. Formally, the attention matrix in the proposed segment-aware self-attention is calculated as:

$$\frac{(\boldsymbol{E}_{seg} \cdot \boldsymbol{Q})\boldsymbol{W}^{\boldsymbol{Q}}((\boldsymbol{E}_{seg} \cdot \boldsymbol{K})\boldsymbol{W}^{K})}{\sqrt{d_x}} \tag{5}$$

where $\boldsymbol{E}_{seg} \in \boldsymbol{R}^{s*d_x}$ is the segment embedding and $\cdot$ represents dot production.[9]

4.3 Two-phase Pre-training

We apply the pretraining-finetuning paradigm for training the proposed model. The pre-training process can be divided into two phases: In the first phase, we follow Lee et al. (2021) to pre-train a XLM-R model with a modified translation language model objective on the monolingual corpus, and then utilize the pre-trained parameters of XLM-R to initialize the encoder of the proposed model.[10] In the second phase, we apply the combination of all the constructed synthetic corpus to pre-train the whole model. After pre-training, we finetune the model on the golden training set of *WeTS*.

5 Experiments and Results

We first describe the experimental settings, including datasets, pre-processing, and hyper-parameters; Then we introduce the baseline systems and report the main experimental results.

5.1 Datasets and Pre-processing

To make our results reproducible, we pre-train our model on the publicly available datasets from the WMT2019 and WMT2014 shared translation tasks for Chinese-English and English-German language pairs respectively. We use the full training set of the WMT14 English-German, which contains 4.5M sentence pairs. For the WMT19 Chinese-English dataset, we remove sentences longer than 200 words and get 20M sentence pairs. The NMT models utilized for constructing synthetic corpus are identical to the ones used for constructing *WeTS*. For each translation direction, the source and target corpus are jointly tokenized into sub-word units with BPE (Sennrich et al., 2016). The source and target vocabularies are extracted from the source and target tokenized synthetic corpus respectively. During fine-tuning, we pre-process the golden corpus with the same tokenizer utilized in pre-training. For details about datasets and pre-processing, we refer the readers to Appendix A.

5.2 Hyper-parameters and Evaluation

We take the Transformer-base (Vaswani et al., 2017) as the backbone of our model, and we use beam search with a beam size of 4 for searching the results. The proposed model is implemented based on the open-source toolkit fairseq.[11] BLEU is utilized as the evaluation metric and we report the BLEU scores on the test sets of *WeTS*. For the direction of English-to-Chinese, we report the character-level BLEU. For the other three directions, we report the case-sensitive BLEU on the de-tokenized sentences. In this paper, we utilize the script of *multi-bleu.pl* as the evaluation tool. We refer the readers to the appendix B for details.

[9] We also tried to sum the $\boldsymbol{E}_{seg}$ with $\boldsymbol{Q}$ or $\boldsymbol{K}$, but we did not get any improvement.

[10] For details about the first-phase pre-training, we refer the readers to the work of Lee et al. (2021).

[11] https://github.com/pytorch/fairseq

#	Systems	*Vanilla TS*				*TS with hints*			
		Zh⇒En	En⇒Zh	De⇒En	En⇒De	Zh⇒En	En⇒Zh	De⇒En	En⇒De
1	XLM-R	21.25	32.48	27.40	25.12	57.49	80.41	65.81	60.38
2	Naive Transformer	24.20	35.01	30.08	28.15	68.31	91.49	70.15	67.40
3	Dual-source Transformer	24.29	35.10	30.23	28.09	68.42	91.72	70.18	67.51
4	SA-Transformer (ours)	**25.51***	**36.28***	**31.20***	**29.48***	**69.49***	**92.78***	**72.84***	**69.20***

Table 2: The main results on the four language pairs. The numbers with '*' indicate the significant improvement over the baseline of naive Transformer with $p < 0.01$ under t-test.

5.3 Baselines

XLM-R. The first baseline system we consider is the work of Lee et al. (2021) who propose the TS system based on XLM-R (Conneau et al., 2020). Following Lee et al. (2021), we re-implement the XLM-based TS model based on the open-source toolkit of XLM (Lample and Conneau, 2019) with slight modification.

Naive Transformer. We take the naive Transformer (Vaswani et al., 2017) as the second baseline and we directly apply the implementation of fairseq toolkit.

Dual-source Transformer. We finally consider the dual-source Transformer (Junczys-Dowmunt and Grundkiewicz, 2018) which applies two shared encoders to encode the source sentence and masked translation respectively. We re-implement the model based on the fairseq toolkit.

All of the baseline systems mentioned above are trained in the same way as our system.

5.4 Main Results

Table 2 shows the main results of our experiments. We can find that, compared to all of the baseline systems, the proposed SA-Transformer achieves SOTA results on all of the four translation directions. Compared with the XLM-based approach (comparing systems 2-4 with system 1), the Transformer-based approach can achieve substantial gains on the final performance, especially on the sub-task *TS with hints*. While the dual-source Transformer has a more complex model structure, it only achieves comparable results with the naive Transformer. We conjecture the main reason is that the dual-source Transformer does not model the interaction between the source and translation, as the source and translated sentences are encoded with two separate encoders in the dual-source Transformer. Compared to the naive Transformer, the proposed model achieves the improvement up to +1.3 BLEU points on Chinese-to-English translation direction in the sub-task *vanilla TS*. By comparing *vanilla TS* and *TS with hints*, it is easy to be noticed that *TS with hints* achieves much better performance than *vanilla TS*, and it even achieves the BLEU score over 90 on English-to-Chinese translation direction. This shows that the translators' hints are strong features for the model to predict the right suggestions. However, in the practical application scenario, the users usually tend to give partial hints for their ideal suggestions, which may introduce more challenges for the proposed model. The data for TS with partial hints can be easily constructed from the proposed *WeTS* and we leave the experiments for future work.

Systems	En⇒Zh	
	w/o finetuning	w/ finetuning
Ours	**29.76**	**36.28**
w/o on golden corpus	26.64	34.27
w/o on pseudo corpus	26.04	34.01
w/o with word alignment	21.26	28.42

Table 3: Results on the effects of synthetic corpus.

6 Analysis

We analyze the proposed approach on the sub-task *vanilla TS*. With the limitation of space, we report the performance on two directions for most of the following experiments.

6.1 Effects of Synthetic Parallel Corpus

In this paper, we propose three different ways for constructing synthetic corpus for the second-phase pre-training. A natural question is that how each of the synthetic corpora affects the performance. We investigate this problem by studying the performance on the English-to-Chinese direction with different synthetic corpus. We report both intermediate and final performances of the model, where fine-tuning is removed and applied respectively. Results are presented in Table 3. As shown in Table 3, we can find that the model trained on the combination of all three kinds of synthetic corpus achieves the best performance. The synthetic corpus constructed with word alignment contributes the most to the final performance among all the synthetic corpora.

6.2 Study the Training Procedure

We adopt the pretraining-finetuning paradigm for the model training, where the two-phase pre-training enhances the model's ability in modeling the general inputs and the fine-tuning further enhances the performance of the model on the golden test sets. In this section, we aim to investigate how the training procedure affects the final performance. Table 4 shows the experimental results. As Table 4 shows, the model achieves very low BLEU scores, i.e., 6.70 in English-to-Chinese and 5.87 in English-to-German, if pre-training is not applied. This is mainly because that the golden corpus of *WeTS* is too scarce to train a well-performed TS model. In the two-phase pre-training, the second-phase pre-training plays a more important role for the final performance, with a decrease of almost 20 BLEU score on English-to-Chinese translation direction if removed. Fine-tuning on the golden corpus of *WeTS* substantially enhances the performance, with an improvement of almost 8 BLEU score on the English-to-German translation direction.

System	En⇒Zh	En⇒De
Ours	**36.28**	**29.48**
w/o fine-tuning	29.76	21.44
w/o pre-training	6.70	5.87
w/o first-phase pre-training	34.63	28.37
w/o second-phase pre-training	16.85	14.14

Table 4: Results on the effects of training strategies.

6.3 Ablation Study on Model Structure

To understand the importance of different components of the model, we perform an ablation study by training multiple versions of the model with some components removed or degenerated into the corresponding components in the naive Transformer. We mainly test three components, including the independent position encoding, segment embedding, and the segment-aware self-attention. Experimental results are reported in Table 5. We find that the best performance is obtained with the simultaneous use of all test components. The most critical component is the segment-aware self-attention, which enables the model to perform a different calculations of self-attention according to the type of the input tokens. When we remove the segment embedding, we get 0.46 BLEU points decline on the English-to-Chinese translation direction. And when the segment-aware self-attention is removed, the decline can be as large as 0.77 BLEU points. These results indicate that the segment information is important for the proposed model, and the segment-aware self-attention can provide more useful segment information.

System	En⇒Zh	En⇒De
SA-Transformer	**36.28**	**29.48**
w/o independent position encoding	36.01	29.35
w/o segment embedding	35.82	29.01
w/o segment-aware self-attention	35.51	28.74

Table 5: Results for the ablation study. 'w/o segment embedding' means that the segment embedding is not added into the token representation, but still inserted in the segment-aware self-attention.

6.4 Case Study and Weaknesses

We present some examples in Chinese-to-English and English-to-Chinese directions, and each example includes the source sentence, translation, incorrect word span, and corresponding suggestions. For case 1 in Figure 3, the Chinese word "火了" (means getting popular) has been wrongly translated into its superficial meaning "fire", and the proposed model gives the right suggestions when the translator selects "fire" as the incorrect part. Similarly, in case 4, the English word "Thursday" has been wrongly translated into "24日", and our model provides three correct alternatives. Case 2 shows that our model can fill in the missed constituents in the translation. Case 3 demonstrates that the proposed model can generate more fluent alternatives. While achieving promising performance, the proposed model still has some weaknesses in the real application: 1) The suggestions sometimes have low diversity. This is mainly because that the search space of the beam search is too narrow to extract diverse suggestions (Wu et al., 2020; Sun et al., 2020). 2) The model tends to provide less satisfactory suggestions for the incorrect spans which include too many words. 3) The best suggestion does not always rank in the first position.

7 Related Work

Related tasks. Some similar techniques have been explored in CAT. Green et al. (2014) and Knowles and Koehn (2016) study the task of so-called translation prediction, which provides pre-

#	Inputs	Suggestions
1 (Zh => En)	**Src:** 一首被称为“神曲”的《生僻字》在网上火了。 **Src in pinyin:** yi shou bei cheng wei shen qu de sheng pi zi zai wang shang huo le. **Translation:** Asong called “shenqu” “rare words” on the internet **fire**.	1 became popular 2 has become popular 3 has been popular
2 (Zh => En)	**Src:** 今天天气很不错，想一起出去逛街么？ **Src in pinyin:** jin tian tian qi hen bu cuo, xiang yi qi chu qu guang jie me? **Translation:** Today is a beautiful day, **want to** go out shopping together?	1 do you want to 2 do you like to 3 you want to
3 (En => Zh)	**Src:** A new policy was adopted to achieve the peaceful unification of our country **Translation:** 对于和平实现祖国统一，已经采取了新的政策 **Translation in pinyin:** dui yu he ping shi xian zu guo tong yi, yi jing cai qu le xin de zheng ce	1 为实现祖国和平统一 2 对于和平实现统一 3 和平实现团结统一
4 (En => Zh)	**Src:** France would not join a US military invasion of Haiti as part of an effort to restore democratic rule, French Foreign Minister said Thursday. **Translation:** 法国外交部长 24 日表示，法国不会加入美国对海地的军事入侵，这是法国恢复民主统治努力的一部分。 **Translation in pinyin:** fa guo wai jiao bu zhang 24 ri biao shi, fa guo bu hui jia ru mei guo dui hai di de jun shi ru qin, zhe shi fa guo hui fu min zhu tong zhi nu li de yi bu fen	1 周四 2 在周四 3 本周四

Figure 3: Case study for the proposed approach. ’Src’ means the source sentence. The segment in red color represents the incorrect part in the translation, and the top-3 suggestions are provided for each incorrect part. For readability, we provide the pinyin version for each Chinese sentence.

dictions of the next word (or phrase) given a prefix. Huang et al. (2015) and Santy et al. (2019) further consider the hints of the translator in the task of translation prediction. Compared to TS, the most significant difference is the strict assumption of the translation context, i.e., the prefix context, which severely impedes the use of their methods under the scenarios of PE. Lexically constrained decoding which completes a translation based on some unordered words, relaxes the constraints provided by human translators from prefixes to general forms (Hokamp and Liu, 2017; Post and Vilar, 2018; Kajiwara, 2019; Susanto et al., 2020). Although it does not need to re-train the model, its low efficiency makes it only applicable in scenarios where only a few constraints need to be applied. Recently, Li et al. (2021) study the problem of auto-completion with different context types. However, they only focus on the word-level auto-completion, and their experiments are also conducted on the automatically constructed datasets.

Related models. Lee et al. (2021) propose to perform translation suggestion based on XLM-R, where the model is trained to predict the masked span of the translation sentence. During inference, they need to generate multiple inputs for the selected sequence of words, with each input containing a different number of the "[MASK]" token. Therefore, the inference process of XLM-R based model gets complex and time-consuming. With the success on many sequence-to-sequence tasks, Transformer can generate sequences with various lengths. The naive Transformer treats each token in the input sentence without any distinction. Based on Transformer, (Junczys-Dowmunt and Grundkiewicz, 2018) propose the dual-source encoder for the task of PE. Wang et al. (2020) also apply the dual-source encoder to the touch-editing scenario, and they also consider the translator’s actions for PE. In parallel to our work, Zhang et al. (2021) propose a domain-aware self-attention to address the domain adaptation. While their idea is similar to the proposed segment-aware self-attention, they introduce large-scale additional parameters.

8 Conclusion and Future work

In this paper, we propose a benchmark for the task of translation suggestion. We construct and share a golden dataset, named *WeTS*, for the community, and propose several ways for automatically constructing the synthetic corpus which can be used to improve the performance substantially. Additionally, we for the first time propose the segment-aware self-attention based Transformer, named SA-Transformer, which achieves the SOTA performance on all four translation directions. We hope our work will provide a new perspective and spur future researches on TS.

There are two promising directions for the future work. First, we plan to make up for the weaknesses discussed in Section 6.4. Second, we decide to consider TS from the perspective of the recommendation system, and from which we introduce new techniques to generate more diverse and accurate suggestions.

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 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 25–28.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

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*, 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 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1225–1236.

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

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*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. Ms-uedin submission to the wmt2018 ape shared task: Dual-source transformer for automatic post-editing. *arXiv preprint arXiv:1809.00188*.

Tomoyuki Kajiwara. 2019. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6047–6052.

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

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Dongjun Lee, Junhyeong Ahn, Heesoo Park, and Jaemin Jo. 2021. Intellicat: Intelligent machine translation post-editing with quality estimation and translation suggestion. *arXiv preprint arXiv:2105.12172*.

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

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. *arXiv preprint arXiv:1804.06609*.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231.

Zewei Sun, Shujian Huang, Hao-Ran Wei, Xin-yu Dai, and Jiajun Chen. 2020. Generating diverse translation by manipulating multi-head attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8976–8983.

Raymond Hendy Susanto, Shamil Chollampatt, and Liling Tan. 2020. Lexically constrained neural machine translation with levenshtein transformer. *arXiv preprint arXiv:2004.12681*.

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.

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 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 1–11.

Xuanfu Wu, Yang Feng, and Chenze Shao. 2020. Generating diverse translation from model distribution with dropout. *arXiv preprint arXiv:2010.08178*.

Shiqi Zhang, Yan Liu, Deyi Xiong, Pei Zhang, and Boxing Chen. 2021. Domain-aware self-attention for multi-domain neural machine translation. *Proc. Interspeech 2021*, pages 2047–2051.

Vilém Zouhar, Aleš Tamchyna, Martin Popel, and Ondřej Bojar. 2021. Neural machine translation quality and post-editing performance. *arXiv preprint arXiv:2109.05016*.

A Pre-processing in Detail

For learning the BPE codes on Chinese-English language pairs, the number of the merge operation is set as 64,000. For English-German language pairs, the number of merge operation is 32,000. For constructing the synthetic corpus, we perform randomly sampling on the golden and pseudo parallel corpus. The size of the constructed synthetic corpus is listed as Table 6:

Directions	on golden	on pseudo	with word alignment
En⇒De	9.0M	9.0M	5.8M
De⇒En	9.0M	9.0M	5.3M
Zh⇒En	20M	20M	19.2M
En⇒Zh	20M	20M	18.4M

Table 6: The Statistics about the constructed synthetic corpus. "on golden" indicates the method of sampling on the golden parallel corpus.

B Experimental Settings in Detail

Following the base model in Vaswani et al. (2017), we set the word embedding as 512, dropout rate as 0.1 and the head number as 8. We use beam search with a beam size of 4. The proposed model is implemented based on the open-source toolkit fairseq.[12] For generating the synthetic corpus with word alignment, we set β as 10. During pre-training, the batch size is set as 81,920 tokens, and the learning rate is set as 0.0008. During fine-tuning, the batch size and learning rate are set as 41,960 and 0.0001 respectively. For the first-phase pre-training, we stop training when the model achieves no improvements for the tenth evaluation on the development set. For the process of second-phase pre-training and fine-tuning, we train the whole model for 200,000 and 100 steps respectively. BLEU is utilized as the evaluation metirc and we report the BLEU scores on the test sets of *WeTS*. For the direction of English-to-Chinese, we report the character-level BLEU. For other three directions, we report the case-sensitive BLEU on the de-tokenized sentences. In this paper, we utilize the script of *multi-bleu.pl* as the evaluation tool.

[12] https://github.com/pytorch/fairseq