Easy Guided Decoding in Providing Suggestions for Interactive Machine Translation

Ke Wang[*], **Xin Ge**[*], **Jiayi Wang, Yu Zhao, Yuqi Zhang**[†]
Alibaba Group Inc.
{moyu.wk,shiyi.gx,joanne.wjy,kongyu, chenwei.zyq}@alibaba-inc.com

Abstract

Machine translation technology has made great progress in recent years, but it cannot guarantee error-free results. Human translators perform post-editing on machine translations to correct errors in the scene of computer-aided translation. In favor of expediting the post-editing process, many works have investigated machine translation in interactive modes, in which machines can automatically refine the rest of translations constrained by human's edits. Translation Suggestion (TS), as an interactive mode to assist human translators, requires machines to generate alternatives for specific incorrect words or phrases selected by human translators. In this paper, we utilize the parameterized objective function of neural machine translation (NMT) and propose a novel constrained decoding algorithm, namely Prefix-Suffix Guided Decoding (PSGD), to deal with the TS problem without additional training. Compared to the state-of-the-art lexically constrained decoding method, PSGD improves translation quality by an average of 10.87 BLEU and 8.62 BLEU on the *WeTS*[1] and the WMT 2022 Translation Suggestion datasets[2], respectively, and reduces decoding time overhead by an average of 63.4% tested on the WMT translation datasets. Furthermore, on both of the TS benchmark datasets, it is superior to other supervised learning systems trained with TS annotated data.

1 Introduction and Related Work

The emergence of machine translation technology (Lopez, 2008; Koehn, 2009) assists human translation to improve translation efficiency (Green et al., 2014, 2015; Herbig et al., 2020). Even though there is a quality gap between the outputs of machine translation (MT) systems and manual translations by professional translators, MT can still practically reduce time in comparison with translating from scratch (Läubli et al., 2013). Later, the advances at the sequence-to-sequence model (Sutskever et al., 2014; Bahdanau et al., 2014; Vaswani et al., 2017) further made a breakthrough in translation technology, inspiring the industry to transform human translation into computer-aided translation (CAT) (Knowles and Koehn, 2016; Santy et al., 2019) to a great extent. CAT usually relies on an MT engine and a platform with a user-friendly interface (Bowker, 2002; Lengyel et al., 2004; Bowker and Fisher, 2010; Bowker, 2014; Pal et al., 2016; Chatterjee, 2019), with which humans perform post-editing (PE) on machine translations to achieve final results with quality standards.

In the past, the post-editing process was typically static and machines would no longer respond to humans' modifications once humans started post-editing. Recent works (Domingo et al., 2016; González-Rubio et al., 2016; Peris et al., 2017) investigate interactive protocols and algorithms so that humans and machines can collaborate and machines automatically refine the translations according to the human's edits. One promising mode is Translation Suggestion (TS) proposed by Yang et al. (2021a) as a pioneer, which requires the machine to provide alternatives for specific spans of incorrect words or phrases selected by humans, namely making suggestions given prefix and suffix constraints. In practical applications as shown in Figure 1, it usually happens when human translators would like to edit part of the MT output. It can be easily implemented with a user interface if machines correctly provide suggestions for the selected incorrect spans. Yang et al. (2021a) has proven the significance of TS in post-editing in terms of resolving two pitfalls of earlier works. The importance has also been recognized by the Conference of Machine Translation (WMT), and they released the Naive Translation

[*] indicates equal contribution.
[†] indicates the corresponding author.
[1] https://github.com/ZhenYangIACAS/WeTS
[2] https://www.statmt.org/wmt22/translation-suggestion-task.html

Example 1	
Source	*Ressortchef Robert Gates hatte am Vortag erklärt, er nehme den kritischen Bericht des Rechnungshofes im Kongress sehr ernst und erwäge verschiedene Möglichkeiten.*
Translation	***Defense Secretary*** *Robert Gates said the day before that he took the critical report by the Court of Auditors in Congress very seriously and considered various options.*
Suggestion for the span	*Head of department*
Final Translation	*Head of department Robert Gates said the day before that he took the critical report by the Court of Auditors in Congress very seriously and considered various options.*
Example 2	
Source	*Er soll Ende 2008 für die Dauer von zweieinhalb Jahren bestimmt werden und die EU in aller Welt repräsentieren. (cp/dpa)*
Translation	*At the end of 2008,* ***it*** *is to be appointed for two and a half years and represent the EU around the world (cp / dpa).*
Suggestion for the span	*he*
Final Translation	*At the end of 2008, he is to be appointed for two and a half years and represent the EU around the world (cp / dpa).*

Figure 1: Examples of Translation Suggestions in computer-aided translation. Highlighted are spans with incorrect words selected by humans. The TS system automatically generates alternatives for the spans to obtain final translation results.

Suggestion shared task (Yang et al., 2022) in WMT 2022[3].

One of the solutions to TS can be training a supervised model with TS annotated data. Yang et al. (2021a) trained such an end-to-end Transformer-like model as a benchmark system. Ge et al. (2022) applied model fine-tuning with TS data augmentation on pre-trained NMT models. However, supervised learning, which relies on a large amount of labeled data, is too heavy to be easily adjusted to other domains. In addition, due to the complicated post-editing process, it is expensive to obtain such limited annotated data.

Our idea is to investigate inference algorithms given prefix and suffix constraints. We tested the state-of-the-art lexically constrained decoding algorithm (Post and Vilar, 2018; Hu et al., 2019) on the *WeTS* dataset, and found that omissions frequently occur in suggestion generations. There are two reasons behind: (1) the division of beams in the dynamic beam allocation narrows the search space so that there would not be enough candidates that match constraints to be picked. This is more likely to happen when constraints are much longer than the average length of the span selected, such as in TS applications; (2) the beam search stops when the probability of *eos*, the special token for the end of a sentence, appears largest in the softmax distribution, but the probability of the entire sentence generation has not been considered. In terms of efficiency, this decoding algorithm contains unnecessary calculations since it always generates suggestions step by step from the beginning of the translation to the end of the sentence, including prefix and suffix constraints.

In this paper, we propose a neat prefix-suffix guided decoding (PSGD) algorithm for the TS task. There are three main contributions: (1) PSGD emphasizes the probability of the entire generation including prefix and suffix constraints, but only decodes for the incorrect span rather than the whole translation sentence, which improves both suggestion quality and time efficiency. (2) PSGD theoretically avoids dividing beams as in Hu et al. (2019) so that the original beam search space can be used to improve the quality of generated translation suggestions. (3) PSGD does not require any additional training/fine-tuning on the original NMT model, which means it can be applied to any auto-regressive machine translation system with flexibility.

Our experimental observations show that PSGD significantly outperforms the state-of-the-art lexically constrained decoding method (Post and Vilar, 2018; Hu et al., 2019) by an average increase of 10.87 BLEU and 8.62 BLEU on the benchmark datasets *WeTS* and WMT 2022 Naive Translation Suggestion datasets (*WMT22-TS*), respectively. Experimental results also demonstrate PSGD's superiority in overall time efficiency by a 63.4% time reduction. In addition, on both the *WeTS* and *WMT22-TS* datasets, PSGD is superior over other supervised learning systems trained with TS annotated data.

[3]`https://statmt.org/wmt22/translation-suggestion-task.html`

Symbol	Definition
$\mathbf{x}$	Sentence in source language
$\mathbf{y}$	Translated sentence in target language
Θ	Model parameters
$\mathbf{p}$	A given prefix of $\mathbf{y}$
$\mathbf{s}$	A given suffix of $\mathbf{y}$
$\mathbf{r}$	The remained part of $\mathbf{y}$ after $\mathbf{p}$ and $\mathbf{s}$ are removed
t_p	The number of tokens in $\mathbf{p}$
t_s	The number of tokens in $\mathbf{s}$
t_r	The number of tokens in $\mathbf{r}$
bos	The special begin of sentence token used in machine translation
eos	The special end of sentence token used in machine translation
y_i	The i-th token of $\mathbf{y}$, similar for x_i, p_i, s_i, r_i, etc.
$\hat{\mathbf{y}}$	The estimation/prediction of $\mathbf{y}$, similar for $\hat{\Theta}$, $\hat{\mathbf{r}}$, $\hat{r}_j$, etc.
$\mathbf{v}_{<t}$	The first t tokens of sequence $\mathbf{v}$
$\langle \mathbf{v}_1, ..., \mathbf{v}_n \rangle$	Concatenation of sequences $\mathbf{v}_1, ..., \mathbf{v}_n$
P_{MT}	The probabilistic model of machine translation
$P_r^{(n)}$	The probability of the entire sequence after the n-th decoding step for the remain part r
pt	Patience for decoding early stopping
$f(y_t)$	The softmax probability distribution at the t-th decoding step

Table 1: Notations

2 Preliminary

Before introducing the PSGD, we first elaborate on the problem in the scene of TS with the mathematical notations in Table 1. The sequence-to-sequence machine translation model is generally a conditional auto-regressive language model that follows the factorized probability distribution.

$$P_{\text{MT}}(\mathbf{y}|\mathbf{x};\Theta) = \prod_i P_{\text{MT}}(y_i|\mathbf{y}_{<i}, \mathbf{x};\Theta) \quad (1)$$

Given training pairs of $(\mathbf{x}, \mathbf{y})$, maximizing the probability in Formula 1 will return an estimate of the model parameter, $\hat{\Theta}$.

During inference, we are supposed to maximize the probability of the generated sequence:

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} P_{\text{MT}}(\mathbf{y}|\mathbf{x};\hat{\Theta}) \quad (2)$$

However, the solution space $\{\mathbf{y}\}$ is infinite and impossible to search. Therefore, an auto-regressive decoding process is applied as an approximation. The decoder of the model decodes the translation step by step greedily[4] as following until *eos* is met:

$$\hat{y}_i = \operatorname*{argmax}_{y} P_{\text{MT}}(y|\hat{\mathbf{y}}_{<i}, \mathbf{x};\hat{\Theta}) \quad (3)$$

In the scene of TS, a span $\mathbf{r}$ is masked for suggestion generation, and thus we need to guarantee that $\mathbf{y}$ starts with the given prefix $\mathbf{p}$ and ends with the given suffix $\mathbf{s}$. Then, the problem becomes:

$$\hat{\mathbf{r}} = \operatorname*{argmax}_{\mathbf{r}} P_{\text{MT}}(\langle \mathbf{p}, \mathbf{r}, \mathbf{s} \rangle|\mathbf{x};\hat{\Theta}) \quad (4)$$

3 Methodology

As mentioned previously, earlier works of constrained decoding require step-by-step generations for the whole sentence even with prefix and suffix constraints. It causes efficiency issues especially when the constraints are long. What's worse, the division of beams might bring about low-quality suggestions. Accordingly, PSGD generates translation suggestions given prefix and suffix constraints through maximizing the probability of the whole generated sequence. Meanwhile, PSGD performs a step-by-step beam search generation only for the span without dividing beams. A brief example of the PSGD algorithm for TS can be found in Figure 2. With this picture in mind, we will give details of the PSGD algorithm in this Section. The pseudo codes of this algorithm are presented in Algorithm 1. And the fairseq based implementation is available on github[5].

3.1 Decoding Process for TS

Similar to the normal translation process, we apply the auto-regressive decoding process to overcome the issue of infinite solution space $\{\mathbf{r}\}$. We can factorize the entire probability in Formula 4 into three parts:

[4]We do not involve beam-search here to simplify the equations, but in practice, all our experiments are conducted with beam-search.

[5]https://github.com/wangke1996/translation-suggestion-psgd

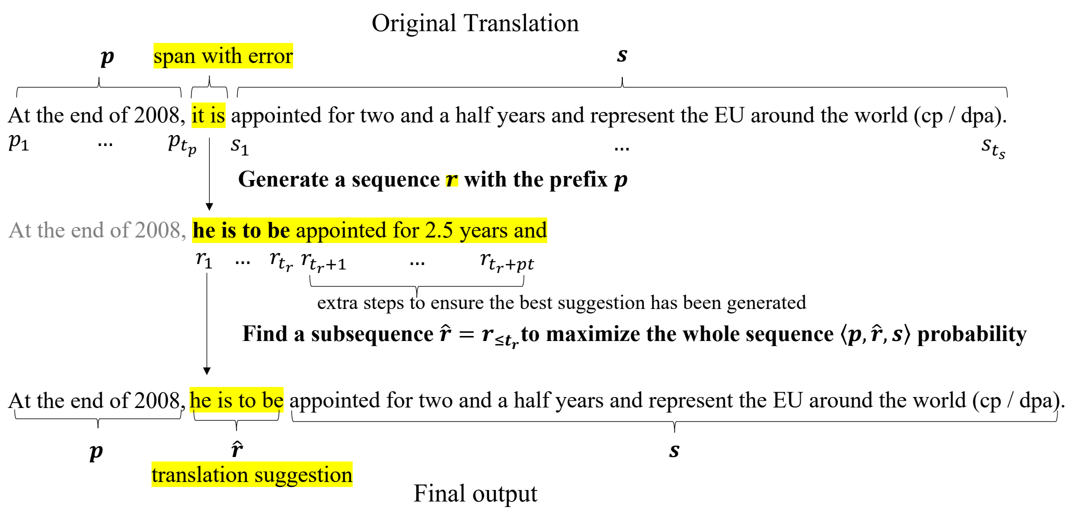

Original Translation
p
span with error
s
At the end of 2008, it is appointed for two and a half years and represent the EU around the world (cp / dpa).
p_1 ... p_{t_p} s_1 ... s_{t_s}
Generate a sequence r with the prefix p
At the end of 2008, he is to be appointed for 2.5 years and
r_1 ... r_{t_r} r_{t_r+1} ... r_{t_r+pt}
extra steps to ensure the best suggestion has been generated
Find a subsequence $\hat{r} = r_{\leq t_r}$ to maximize the whole sequence $\langle p, \hat{r}, s \rangle$ probability
At the end of 2008, he is to be appointed for two and a half years and represent the EU around the world (cp / dpa).
p
$\hat{r}$
s
translation suggestion
Final output

Figure 2: An example of applying the PSGD algorithm to generate translation suggestions

.

Algorithm 1 Algorithm of PSGD

```
Input: $\mathbf{x}$, $\mathbf{p}$, $\mathbf{s}$
Parameter: $pt$
Output: $\hat{\mathbf{r}}$
 1: Let $\hat{\mathbf{r}} = []$, $n = 0$, $maxP_r = -\infty$, $bestStep = 0$.
 2: while $n - bestStep < pt$ do
 3:     $\hat{\mathbf{y}} = \langle \mathbf{p}, \hat{\mathbf{r}}, \mathbf{s} \rangle$
 4:     $\{f(p_i)\}, \{f(\hat{r}_j)\}, \{f(s_k)\}$ = Output probability distribution of every tokens in $\hat{\mathbf{y}} = \langle \mathbf{p}, \hat{\mathbf{r}}, \mathbf{s} \rangle$
 5:     calculate $P_r^{(n)}$ with Formula 9
 6:     if $P_r^{(n)} > maxP_r$ then
 7:         $maxP_r = P_r^{(n)}$
 8:         $bestStep = n$
 9:     end if
10:     calculate $\hat{r}_{n+1}$ with Formula 10
11:     append $\hat{r}_{n+1}$ to $\hat{\mathbf{r}}$
12:     $n = n + 1$
13: end while
14: return the first $bestStep$ elements of $\hat{\mathbf{r}}$
```

$$
\begin{aligned}
&P_{\text{MT}}(\langle \mathbf{p}, \mathbf{r}, \mathbf{s} \rangle | \mathbf{x}; \Theta) \\
&= \prod_i P_{\text{MT}}(p_i | \mathbf{p}_{<i}, \mathbf{x}; \Theta) \\
&\cdot \prod_j P_{\text{MT}}(r_j | \langle \mathbf{p}, \mathbf{r}_{<j} \rangle, \mathbf{x}; \Theta) \\
&\cdot \prod_k P_{\text{MT}}(s_k | \langle \mathbf{p}, \mathbf{r}, \mathbf{s}_{<k} \rangle, \mathbf{x}; \Theta)
\end{aligned}
\tag{5}
$$

Notice that in Formula 5, as all p_i and s_k are given, only the second product requires a step-by-step auto-regressive generation. The first product, referring to the probability of the prefix sequence $\mathbf{p}$, can be calculated at the first step when r_1 is decoded. And when all of the decoding steps of $\mathbf{r}$ are completed, we can easily obtain the third product, which refers to the probability of the suffix sequence $\mathbf{s}$. Hence, the decoding process at each step is actually:

$$\hat{r}_j = \operatorname*{argmax}_r P_{\text{MT}}(r | \langle \mathbf{p}, \hat{\mathbf{r}}_{<j} \rangle, \mathbf{x}; \hat{\Theta}) \tag{6}$$

Theoretically, we only need to go forward through the model for t_r times, which is smaller than that of a normal constraint decoding (Hu et al., 2019), i.e. $t_p + t_r + t_s$. In particular, for the TS task, human translators might select a short span with incorrect words, and then $t_p + t_r + t_s$ could be much larger than t_r.

3.2 Condition for Stopping Decoding

The decoding process aforementioned is straightforward. Now, it comes to a problem: when should the inference stop? In Post and Vilar (2018) and Hu et al. (2019), the model generates the full sequence and the generation stops when the *eos* is met. However, in PSGD, only the span $\mathbf{r}$ will be generated and $\mathbf{r}$ does not end with *eos*. We cannot utilize the generation of *eos* as a signal to stop decoding.

The key to solving this problem can be aligned with the objective function of NMT training as described in Formula 1. We assume that we have finished decoding in Formula 6 for N steps, where N is large enough to obtain a hypothesis sequence

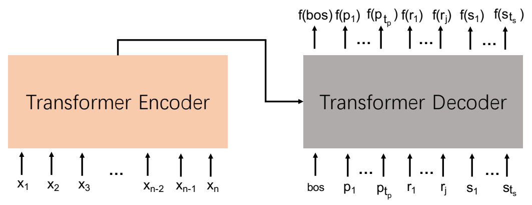

f(bos) f(p$_1$) f(p$_{t_p}$) f(r$_1$) f(r$_j$) f(s$_1$) f(s$_{t_s}$)
Transformer Encoder
Transformer Decoder
x$_1$ x$_2$ x$_3$ ... x$_{n-2}$ x$_{n-1}$ x$_n$
bos p$_1$ ... p$_{t_p}$ r$_1$ ... r$_j$ s$_1$ ... s$_{t_s}$

Figure 3: Forward path for calculating $P_r^{(n)}$

.

$\mathbf{r}^*$. The solution space $\{\mathbf{r}\}$ becomes $\{\mathbf{r}^*_{<n}\}_{n=1}^{N}$. The most optimal sequence $\hat{\mathbf{r}} = \mathbf{r}^*_{<\hat{n}}$ is obtained by calculating $\hat{n}$ as:

$$\hat{n} = \operatorname*{argmax}_{n} P_{MT}(\langle \mathbf{p}, \mathbf{r}^*_{<n}, \mathbf{s}\rangle | \mathbf{x}; \hat{\Theta}) \tag{7}$$

The normal decoding process stops when *eos* is reached, and it does not necessarily mean that the probability of the entire sequence generation is maximized. In other words, there might be a better hypothesis if the model generates fewer or more steps in the normal decoding process. Our condition for stop decoding in PSGD approaches the training objective function in Formula 1 as much as possible. This is the reason why PSGD achieves better performance in suggestions.

3.2.1 Early Stopping

Although the above workaround provides a proper condition for stopping decoding, it might cause efficiency issues since the number of decoding steps would become N rather than t_r. Obviously, N should be equal to or greater than t_r to achieve the best performance. However, we have no idea how large t_r is. We have to set N to a large enough value to cover as many cases as possible, which is unacceptable in practice.

Instead of setting a large N, we try to apply an early stopping mechanism to save unnecessary decoding steps. In Formula 6, after every decoding step, PSGD calculates and records the averaged probability of entire sequence generation with both prefix and suffix constraints concatenated, denoted as $P_r^{(n)}$ as follows:

$$P_r^{(n)} = \frac{P_{MT}(\langle \mathbf{p}, \hat{\mathbf{r}}_{<n}, \mathbf{s}\rangle | \mathbf{x}, \hat{\Theta})}{t_p + n + t_s} \tag{8}$$

PSGD stops decoding when $P_r^{(n)}$ does not increase any more within pt consecutive steps, where pt represents patience for early stopping.

3.2.2 Parallel Acceleration

Because the early stopping mechanism requires extra calculations for $P_r^{(n)}$ at each step, it seems that we would need to go forward through the model for $2 * (t_r + pt)$ times. In fact, by utilizing the future mask in Transformer's decoder, we only run $t_r + pt$ times since $P_r^{(n)}$ and $\hat{r}_{n+1}$ can be obtained at the same time. Details are explained as follows.

Figure 3 shows the forward path for calculating $P_r^{(n)}$, where $f(\hat{y}_t)$ is the softmax probability distribution at the t-th decoding step. $f(y|y_t)$ indicates the probability of token y at $(t+1)$-th step, which is short for $P_{MT}(y|\hat{\mathbf{y}}_{\le t}, \mathbf{x}; \hat{\Theta})$. We have:

$$\begin{aligned} P_r^{(n)} = & \frac{1}{t_p + n + t_s} \cdot f(p_1|bos) \cdot f(p_2|p_1) \cdot \ldots \\ & \cdot f(p_{t_p}|p_{t_p-1}) \cdot f(\hat{r}_1|p_{t_p}) \cdot f(\hat{r}_2|\hat{r}_1) \cdot \ldots \\ & \cdot f(\hat{r}_n|\hat{r}_{n-1}) \cdot f(s_1|\hat{r}_n) \cdot f(s_2|s_1) \cdot \ldots \\ & \cdot f(s_{t_s}|s_{t_s-1}) \cdot f(eos|s_{t_s}) \end{aligned} \tag{9}$$

and

$$\begin{aligned} \hat{r}_{n+1} = & \operatorname*{argmax}_{r} P_{\mathrm{MT}}(r|\langle \mathbf{p}, \hat{\mathbf{r}}_{\le n}\rangle, \mathbf{x}) \\ = & \operatorname*{argmax}_{r} f(p_1|bos) \cdot f(p_2|p_1) \cdot \ldots \\ & \cdot f(p_{t_p}|p_{t_p-1}) \cdot f(\hat{r}_1|p_{t_p}) \cdot f(\hat{r}_2|\hat{r}_1) \cdot \\ & \cdot \ldots \cdot f(\hat{r}_n|\hat{r}_{n-1}) \cdot f(r|\hat{r}_n) \\ = & \operatorname*{argmax}_{r} f(r|\hat{r}_n) \end{aligned} \tag{10}$$

At the n-th step, we can simultaneously get the softmax distributions $\{f(p_i)\}$, $\{f(\hat{r}_j)\}$, and $\{f(s_k)\}$. Then, according to Formula 9 and 10, $P_r^{(n)}$ (the probability of the entire generation for early stopping) and $\hat{r}_{n+1}$ (the next token generation) can be obtained together. Therefore, the total step of the decoding process in PSGD is $t_r + pt$.

3.3 Summary of Methodology

In this section, we have provided full details of PSGD, and the pseudo code is presented in Algorithm 1. By utilizing the parameterized objective function, PSGD maximizes the probability of the entire generation for better suggestion quality, but only generates the suggestion for the incorrect span instead of the whole sequence. Besides, it designs a proper early-stopping mechanism for efficiency improvement. Only $t_r + pt$ decoding steps are required, which is smaller than $t_p + t_r + t_s$ in Hu et al. (2019), since t_r is much smaller than $t_p + t_r + t_s$

on average in the scene of TS, and the value of the hyper-parameter pt is proven to be quite small in the subsequent experimental observations.

4 Experiments

4.1 Generation Performance on Translation Suggestion

Translation suggestion is a significant CAT task where human translators select spans with incorrect words or phrases in the MT results. It requires models to automatically provide alternatives for the spans for efficiency improvement in translation. *WeTS* (Yang et al., 2021a) and *WMT22-TS* (Yang et al., 2022) are the benchmark datasets for the TS task. Each sample consists of a source sentence, a masked MT sentence, and a golden reference corresponding to the masked span. The masked spans with potential translation errors are selected and annotated by humans. The annotations mainly focus on three types of translation errors: under (over)-translations, grammatical or syntactic errors, and semantic errors.

We compare the PSGD with three state-of-the-art systems on *WeTS* and *WMT22-TS*: SA-Transformer (Yang et al., 2021a), TSMind (Ge et al., 2022) (*without TS data augmentation*) and VDBA (Hu et al., 2019). SA-Transformer is an end-to-end transformer-like model, which initializes the encoder with weights from the pre-trained XLM-Roberta (Conneau et al., 2020) and fine-tunes it with TS labeled data. TSMind utilizes the pre-trained NMT model and fine-tunes it with TS triplets as well, and it wins in three out of four language pairs of the Naive Translation Suggestion task in WMT 2022 Yang et al. (2022). VDBA is basically an improved version of the Dynamic Beam Allocation algorithm proposed in Post and Vilar (2018), that is remarkably capable in lexically constrained decoding.

We should notice that both of SA-Transformer and TSMind perform supervised learning with human-annotated data for translation suggestion, while VDBA and our PSGD generate suggestions solely based on the original NMT model (released by Yang et al. (2021a)) without any additional training/fine-tuning on any TS-labeled data. Comparisons between SA-Transformer and TSMind have been reported in Yang et al. (2021a) and Ge et al. (2022), respectively. VDBA and PSGD are both implemented with the Fairseq toolkit (Ott et al., 2019) to generate suggestions. In evaluation,

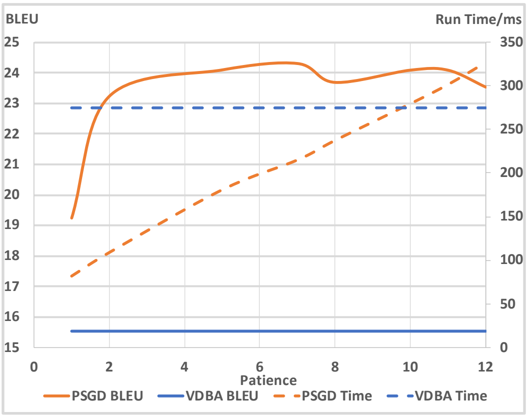


Figure 4: BLEU and decoding time cost with different early stopping patience pt on the *WeTS* German-English development set.

we calculate SacreBLEU metric (Post, 2018) between the golden references for the masked spans and the alternatives generated by each system to measure their performances.

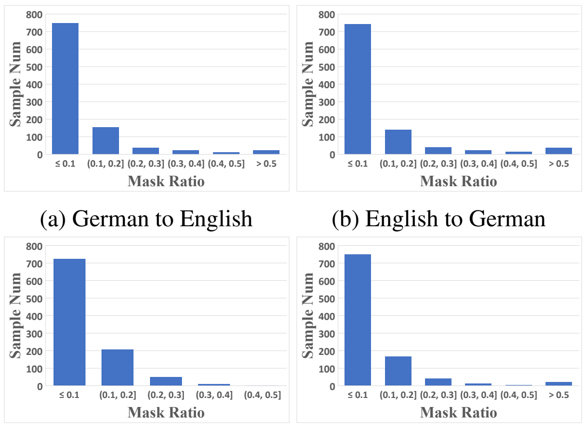


Figure 5: Histograms of the length ratios of the masked spans on the *WeTS* test sets (1000 samples each).

Results on the test set of *WeTS* are presented in Table 2. Notice that results of TSMind are not listed in Table 2. This is because TSMind is a system submitted to the WMT22-TS shared task. Its results on the WeTS dataset are those of systems trained with a large amount of augmented data. To demonstrate the performance of our designed unsupervised method, comparisons with models without data augmentation are more reasonable and much fairer. Therefore, we compared our model performances with the results of TSMind without data augmentation on the development set of *WMT22-TS* in Table 3. It can be observed that our PSGD outperforms VDBA sig-

Method	Note	De-En	En-De	Zh-En	En-Zh	Avg
SA-Transformer (Yang et al., 2021a)	Supervised	31.20	29.48	25.51	36.28	30.62
VDBA (Hu et al., 2019)	Unsupervised	23.71	31.29	12.11	12.96	20.02
PSGD (Ours)		**34.74**	**38.92**	**20.25**	**29.64**	**30.89**

Table 2: BLEU results on the *WeTS* test sets. PSGD outperforms VDBA by +10.87 BLEU on average, and it also outperforms the TS benchmark systems, SA-Transformer, trained with annotated data in German-English and English-German.

Method	Note	De-En	En-De	En-Zh	Zh-En	Avg
TSMind (Ge et al., 2022)	Supervised	33.23	37.14	21.20	16.44	27.00
VDBA (Hu et al., 2019)	Unsupervised	23.16	29.86	13.37	8.97	18.84
PSGD (Ours)		**33.87**	**35.99**	**24.64**	**15.32**	**27.46**

Table 3: BLEU results on the *WMT22-TS* test sets. PSGD outperforms VDBA by +8.62 BLEU on average, and it also slightly outperforms the winning system, TSMind, trained without pseudo corpus on average of 4 language pairs.

nificantly by an average of 10.87 BLEU and an average of 8.62 BLEU separately on the two datasets for German-English, English-German, Chinese-English, and English-Chinese. Moreover, without any extra training, PSGD is even better than the supervised SA-Transformer and TSMind systems that are trained with TS-labeled data.

4.2 Patience for Early Stopping in Decoding

In PSGD, we involve a hyper-parameter pt, the number of extra decoding steps we take for early stopping. Larger pt can usually avoid omissions since when a local optimum is reached, the model should be encouraged to generate in more steps to ensure that it is globally optimal. However, a larger pt would slow down the decoding process.

To determine a proper value for pt, we investigate the model performances in SacreBLEU and the decoding time cost on the German-English development set of $WeTS$. The results are shown in Figure 4, which indicates that when pt increases from a relatively small value, the performance of PSGD will significantly increase as well. However, there is no more much gain when pt is over 5. Taking the trade-off between time efficiency and the suggestion quality into consideration, we empirically set $pt = 5$. For other language pairs, we found $pt = 5$ is also relatively optimal.

4.3 Experiments with Different Mask Ratios

Samples in the *WeTS* dataset have a short length of masked spans on average, whose masked spans contain less than 10% words of the whole translation sentence as displayed in the histograms in Figure 5.

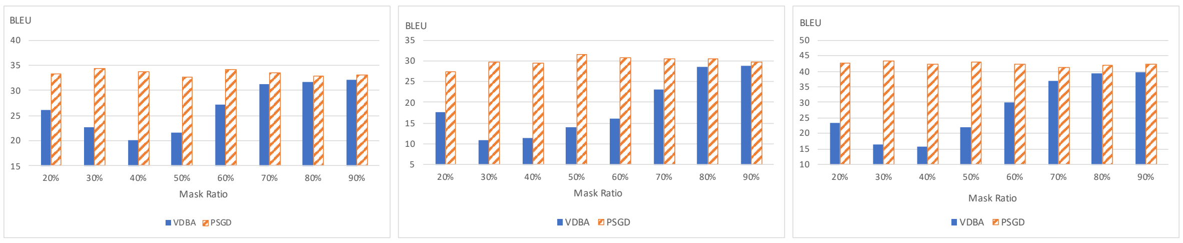
To fairly verify the performance of PSGD on longer masked spans, we consider applying various mask ratios on the translation datasets. We mask a span in the golden references with a random length ratio from 20% to 90% and obtain predictions for the spans by PSGD and VDBA separately. Then, we calculate SacreBLEU scores for comparison.

In practice, we use the WMT 2014 English-French test set (Bojar et al., 2014) and WMT 2021 English-German and German-English test sets (Akhbardeh et al., 2021) in the news translation for evaluation. PSGD and VDBA decode based on the publicly released NMT models, Ng et al. (2019) for English-German and German-English and Ott et al. (2018) for English-French.

Experimental results are shown in Figure 6. It shows that the proposed PSGD outperforms VDBA in all language pairs for each mask ratio. The performance of VBDA reaches the lowest point at the mask ratio of 40%, and then dramatically increases as the mask ratio increases. By contrast, the performance of PSGD appears more stable.

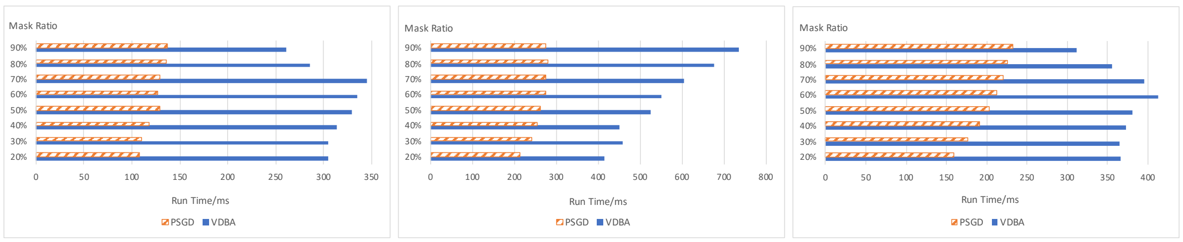
4.4 Computation Efficiency

In translation suggestion, compared with the decoding steps $t_p + t_r + t_s$ of Hu et al. (2019), PSGD is more efficient with $t_r + pt$ decoding steps when pt is smaller than $t_p + t_s$. To thoroughly evaluate the speed of PSGD's decoding process, we



(a) German to English (b) English to German (c) English to French

Figure 6: BLEU of the masked span with different mask ratios on WMT news test sets. PSGD outperforms VDBA in all language pairs with each mask ratios.



(a) German to English (b) English to German (c) English to French

Figure 7: Decoding time costs with different mask ratios on WMT news test sets. PSGD saves 63.4% of time costs compared with VDBA on average.

conduct a running-time cost analysis of translation suggestions with different mask ratios, relying on the WMT news translation test sets mentioned in Section 4.3. All experiments are run on a single Tesla P-100 GPU with CUDA V10.2. The target sequences are generated with a batch size of 10 and a beam size of 5. We obtain the time cost by averaging the running time of each sample in the test sets. Notice that times for pre-processings, such as tokenization and subwords segmentation, are not counted because these parts are identical for both methods. Hyper-parameter pt is empirically set to 5, the same value in the experiments shown in Figure 6.

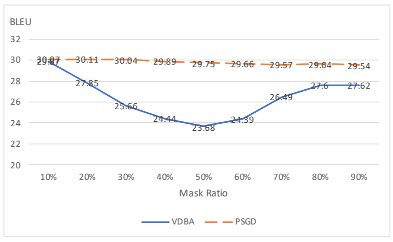
The time costs of PSGD and VDBA are displayed in Figure 7. It is clear that PSGD runs faster than VDBA in all cases. As discussed in Section 4.3, in the scene of TS, the incorrect spans are usually short, which contain less than 10% of words in the whole translation sentences on average. In such a case, referred in Figure 7, PSGD can speed up at least 2x times compared with VDBA.

4.5 Robustness Test

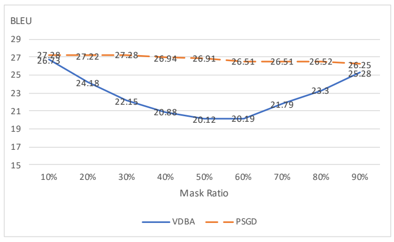
In the above experiments, the given prefix and suffix constraints come from the golden translation references. While in the applications of CAT, prefix and suffix constraints could include erroneous words even if the incorrect span has been selected. Therefore, it is necessary to carry out experiments where the prefix and suffix constraints contain potential errors. We assume that the performances of a robust decoding algorithm will not decrease too much in such a circumstance.

Again, we use the machine translation datasets mentioned in Section 4.3 and 4.4. The difference is that rather than masking golden references, we try to mask a span with random length in the MT result to get the prefix and suffix constraints. This setup is much closer to that of *WeTS*'s. In evaluation, the Sacre-BLEU score can only be calculated between the entire golden reference and the suggestions for the masked span alone with the prefix and suffix constraints, since we don't have the parallel references for the masked spans.

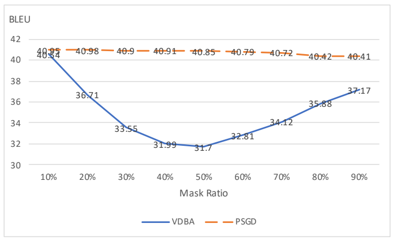
Results of PSGD and VDBA are plotted in Figure 8. It illustrates that PSGD significantly outperforms VDBA in all the cases of mask ratios. When the masked span is extremely short, both algorithms will naturally not impact much on the whole sequence evaluation. On the other hand, when the masked span is extremely long, the decoding process of PSGD or VDBA is almost equivalent to that of the original NMT model. Therefore, in both cases, it is unsurprising to see that the gap between the performances of PSGD and VDBA is small. Except for these two situations, the performance of PSGD remains at a higher level than that



(a) German to English



(b) English to German



(c) English to French

Figure 8: BLEU of whole sentences with different mask ratios when prefix and suffix constraints come from MT outputs.

of VDBA. And VDBA reaches the lowest point when nearly half of MT words are masked.

4.6 Summary of Experiments

In this section, we conduct exhaustive experiments to evaluate the suggestion quality, time efficiency, and robustness of PSGD. PSGD significantly outperforms VDBA by an average increase of 10.87 BLEU and 8.62 BLEU on the benchmark datasets, *WeTS* and *WMT22-TS*, respectively. Experiments in time efficiency show its superiority by an overall 63.4% deduction on the WMT translation datasets. In robustness tests, PSGD remains at a higher level than VDBA all the time. Finally, on the TS benchmark datasets, PSGD is superior over two supervised TS systems, SA-Transformer and TSMind, which are trained with human-annotated data.

5 Conclusion

In this paper, we propose a neat prefix-suffix guided decoding (PSGD) algorithm for the translation suggestion task in computer-aided translation. It emphasizes the probability of the entire generation including prefix and suffix constraints and decodes only for incorrect spans selected by human translators with an early stopping mechanism. Given a pre-trained auto-regressive NMT model, PSGD can be easily applied during inference without any additional training/fine-tuning. Comprehensive experimental results demonstrate that PSGD significantly outperforms the state-of-the-art constrained decoding algorithm, VDBA, in all of the translation quality, time efficiency, and robustness. Meanwhile, it is also superior to supervised TS systems trained with human-annotated data.

Limitations

PSGD is a straightforward constrained-decoding algorithm for the translation suggestion task. However, the early-stopping mechanism involves extra time costs. Though PSGD is more efficient than VDBA in the scene of TS, where only two constraints (prefix and suffix) appear, it could be slower than VDBA if there were more short constraints. Besides, even if we take both prefix and suffix constraints into consideration for emphasis on the whole translation generation, the decoding process is still auto-regressive from left to right. The algorithm could be improved if we made better use of the information of suffix constraints. For example, how to apply PSGD on non-autoregressive models, such as (Gu et al., 2019) and (Yang et al., 2021b) will be our future work.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

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

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

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

Lynne Bowker. 2014. Computer-aided translation: Translator training. In *Routledge encyclopedia of translation technology*, pages 126–142. Routledge.

Lynne Bowker and Des Fisher. 2010. Computer-aided translation. *Handbook of translation studies*, 1:60–65.

Rajen Chatterjee. 2019. Automatic post-editing for machine translation. *arXiv preprint arXiv:1910.08592*.

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.

Miguel Domingo, Alvaro Peris, and Francisco Casacuberta. 2016. Interactive-predictive translation based on multiple word-segments. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 282–291.

Xin Ge, Ke Wang, Jiayi Wang, Nini Xiao, Xiangyu Duan, Yu Zhao, and Yuqi Zhang. 2022. Tsmind: Alibaba and soochow university's submission to the wmt22 translation suggestion task. *arXiv preprint arXiv:2211.08987*.

Jesús González-Rubio, Daniel Ortiz-Martínez, Francisco Casacuberta, and José Miguel Benedi Ruiz. 2016. Beyond prefix-based interactive translation prediction. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 198–207, Berlin, Germany. Association for Computational Linguistics.

Spence Green, Jason Chuang, Jeffrey Heer, and Christopher D Manning. 2014. Predictive translation memory: A mixed-initiative system for human language translation. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 177–187.

Spence Green, Jeffrey Heer, and Christopher D Manning. 2015. Natural language translation at the intersection of ai and hci. *Communications of the ACM*, 58(9):46–53.

Jiatao Gu, Qi Liu, and Kyunghyun Cho. 2019. Insertion-based decoding with automatically inferred generation order. *Transactions of the Association for Computational Linguistics*, 7:661–676.

Nico Herbig, Tim Düwel, Santanu Pal, Kalliopi Meladaki, Mahsa Monshizadeh, Antonio Krüger, and Josef van Genabith. 2020. Mmpe: A multi-modal interface for post-editing machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1691–1702.

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, Volume 1 (Long and Short Papers)*, pages 839–850, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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

Samuel Läubli, Mark Fishel, Gary Massey, Maureen Ehrensberger-Dow, Martin Volk, Sharon O'Brien, Michel Simard, and Lucia Specia. 2013. Assessing post-editing efficiency in a realistic translation environment.

Lengyel, István, Kis, Balázs, Ugray, and Gábor. 2004. Memoq: A new approach to computer-assisted translation.

Adam Lopez. 2008. Statistical machine translation. *ACM Computing Surveys (CSUR)*, 40(3):1–49.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. *arXiv preprint arXiv:1907.06616*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. *CoRR*, abs/1806.00187.

Santanu Pal, Marcos Zampieri, Sudip Kumar Naskar, Tapas Nayak, Mihaela Vela, and Josef van Genabith. 2016. Catalog online: Porting a post-editing tool to the web. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 599–604.

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

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Sebastin Santy, Sandipan Dandapat, Monojit Choudhury, and Kalika Bali. 2019. Inmt: Interactive neural machine translation prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 103–108.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

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.

Zhen Yang, Fandong Meng, Yingxue Zhang, Ernan Li, and Jie Zhou. 2022. Findings of the wmt 2022 shared task on translation suggestion. In *Proceedings of the Seventh Conference on Machine Translation*, pages 821–829, Abu Dhabi. Association for Computational Linguistics.

Zhen Yang, Yingxue Zhang, Ernan Li, Fandong Meng, and Jie Zhou. 2021a. Wets: A benchmark for translation suggestion. *arXiv preprint arXiv:2110.05151*.

Ziyi Yang, Yinfei Yang, Daniel Cer, Jax Law, and Eric Darve. 2021b. Universal sentence representation learning with conditional masked language model. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6216–6228, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
6

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Not applicable. Left blank.

C ☑ Did you run computational experiments?

4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.