BiTIIMT: A Bilingual Text Infilling Method for Interactive Machine Translation

Yanling Xiao[1*] **Lemao Liu**[2†] **Guoping Huang**[2] **Qu Cui**[2]
Shujian Huang[1†] **Shuming Shi**[2] **Jiajun Chen**[1]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, China
[1]Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing
[2]Tencent AI Lab, China
[1]xiaoyl@smail.nju.edu.cn, {huangsj,chenjj}@nju.edu.cn
[2]{redmondliu, donkeyhuang, qucui, shumingshi}@tencent.com

Abstract

Interactive neural machine translation (INMT) is able to guarantee high-quality translations by taking human interactions into account. Existing IMT systems relying on lexical constrained decoding (LCD) enable humans to translate in a flexible translation manner beyond left-to-right. However, they typically suffer from limitations in translation efficiency and quality due to the reliance on LCD. In this work, we propose a novel **BiTIIMT** system, **Bi**lingual **T**ext-**I**nfilling for **I**nteractive Neural **M**achine **T**ranslation. The key idea to BiTIIMT is the Bilingual Text-infilling (BiTI) task which aims to fill missing segments in a manually revised translation for a given source sentence. We propose a simple yet effective solution by casting this task as a sequence-to-sequence task. The benefits of our solution are that it performs efficient decoding with the same complexity as the standard decoding in NMT and makes full use of revised words for better translation prediction. Experiment results show that BiTiIMT performs significantly better and faster than state-of-the-art LCD-based IMT on three translation tasks.

1 Introduction

Recent years have witnessed significant development in neural machine translation(NMT) (Bahdanau et al., 2015; Vaswani et al., 2017). Despite their success, their translation in quality still can not meet the requirements in industrial applications. On the other hand, interactive neural machine translation (IMT) (Foster et al., 1997; Langlais et al., 2000; Simard et al., 2007; Barrachina et al., 2009; González-Rubio et al., 2013; Cheng et al., 2016; Weng et al., 2019; Huang et al., 2021) is able to guarantee high-quality translation: it is an iterative collaboration process between human and machine that involves multiple interactive steps to obtain a satisfactory translation.

Traditional IMT generates a translation in the left-to-right completing paradigm (Langlais et al., 2000; Sanchis-Trilles et al., 2014; Peris et al., 2017a; Knowles and Koehn, 2016; Zhao et al., 2020) where human translators are required to revise words in the translation prefix. This strict left-to-right manner limits its flexibility because some human translators may enjoy their translation manners beyond left-to-right. As a result, another part of works (Weng et al., 2019; Huang et al., 2021) propose an alternative IMT paradigm under which human translators can revise words at arbitrary positions of a translation. The essential technique to this paradigm is lexically constrained decoding (LCD) (Hokamp and Liu, 2017; Post and Vilar, 2018), which extends beam search in the decoding stage to include revised words as constraints.

Unfortunately, LCD-based IMT suffers from two major shortcomings on efficiency and translation quality in practice. Firstly, LCD-based IMT usually involves multiple interactions between a human translator and machine and runs the LCD algorithm multiple times to translate a sentence. Since each LCD run takes considerable time compared with NMT decoding, the human translator will encounter severe latency, leading to poor user experience. In addition, LCD is based on the standard translation model, which is defined on top of the prefix context, and thus cannot make use of the revised words to assist the model in predicting target words to their left. Hence this characteristic limits its overall translation quality.

This paper proposes a simple yet effective IMT approach, BiTIIMT, which addresses the issues above. The core idea to BiTIIMT is the Bilingual Text-infilling (BiTI) task which extends text-infilling (Zhu et al., 2019) from monolingual setting to bilingual setting and aims to fill missing segments in a revised translation for a given source sentence. Unlike Zhu et al. (2019) carefully designing a model, we simply cast the bilingual text-infilling

[*]Work was done during internship at Tencent AI Lab.
[†]Corresponding authors.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 1958 - 1969

task as a sequence-to-sequence task and then employ the standard NMT model to perform this task. To train the model, we construct simulated data by randomly sampling revised sentences from a bilingual corpus and augment the simulated data with bilingual corpus for further improvements. In this way, our model is able to yield a valid output that can be seamlessly filled in the revised translation in an efficient way similar to the standard NMT decoding. Moreover, the proposed model makes full use of all revised words to predict a target word and thus has the potential to obtain better translation than LCD.

We conduct extensive experiments on WMT14 En-De, WMT14 En-Fr, and Zh-En tasks. Our simulated experiments demonstrate that the proposed model indeed outperforms LCD in terms of translation quality and efficiency, and the proposed BiTIIMT is better than LCDIMT according to both translation quality and human editing costs. The advantages of BiTIIMT over LCDIMT are also verified in real-world IMT experiments with human translators.

This paper makes the following contributions:

- It proposes the bilingual text-infilling task and provides a simple yet effective solution to address this task.
- It proposes a novel IMT system on top of bilingual text-infilling which empirically outperforms a strong baseline in both translation quality and efficiency.

2 Background

2.1 Neural Machine Translation

Neural machine translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017) is based on a sequence to sequence model which adopts an encoder-decoder architecture. The encoder summarizes the source sentence into an intermediate representation, and the decoder generates the target sentence.

Given a source sentence $X = \{x_1, \cdots, x_i, \cdots\}$, a NMT model factorizes the distribution over possible output $Y = \{y_1, \cdots, y_T\}$ into a chain of conditional probabilities from left to right:

$$P(Y \mid X; \theta) = \prod_{t=1}^{T+1} P(y_t \mid y_1, y_2, \cdots, y_{t-1}, X; \theta) \quad (1)$$

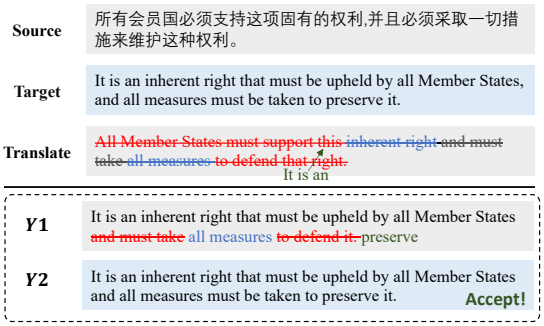

Figure 1: An example of the BiTIIMT. Words with blue fonts are chosen to keep by human translators. Those with strikethrough and red fonts are deleted, and words with green fonts are inserted by humans.

where the special tokens y_0 (e.g. <bos>) and y_{T+1} (e.g. <eos>) are used to represent the beginning and end of all target sentences.

2.2 Text Infilling

Text infilling (Zhu et al., 2019) is a task that fills missing text segments of a sentence or paragraph by a model (Berglund and Leo; Fedus et al., 2018; Zhu et al., 2019) trained on a large amount of data in a fill-in-the-blank format. The input text $X = \{x_1, \text{<blank>}, x_i, ..., \text{<blank>}, ...\}$ has an unknown number of blanks whose positions are arbitrary, and each blank has an arbitrary unknown length.

To address this task, a text infilling model fills in each blank from left to right by predicting a target word y_j at each time step j. As a solution, Zhu et al. (2019) proposes a variant model based on Transformer, whose position encoding takes both segment positions and token positions into account.

3 Proposed BiTIIMT

This section illustrates the overview of the proposed BiTIIMT system and accordingly presents its essential technique (i.e., bilingual text-infilling) to take human interactions into NMT.

3.1 Overview of BiTIIMT

In general, the proposed BiTIIMT enjoys a human-in-the-loop manner to output the final translation, similar to conventional IMT systems (Cheng et al., 2016). Specifically, for a given source sentence X, BiTIIMT iteratively performs the following two steps:

- A human translator edits a translation Y from the translation engine;

- Then the engine updates Y based on the edited translation as well as its source X.

This procedure terminates until the human translator is satisfied with the quality of Y. This procedure is illustrated in Figure 1. The key to BiTIIMT is its second step which relies on Bilingual Text-infilling to update a translation Y. In the rest of this section, the details about bilingual text-infilling will be described.

3.2 Bilingual Text-infilling

3.2.1 Problem Statement and Model

Problem Statement Generally, bilingual text-infilling extends text-infilling from the monolingual setting (Zhu et al., 2019) to the bilingual setting. Suppose $\bar{Y}$ is a template, i.e., the edited (partial) translation, which includes some blanks to be filled in; $Y^b = \{y_1^b, y_2^b, \cdots, \}$ is a sequence of segments used to fill the blanks in $\bar{Y}$. BiTI aims to generate Y^b for filling the blanks in $\bar{Y}$, to obtain a translation Y for a source sentence X.

Take Figure 2 as an example, the template $\bar{Y}$ is a partial translation edited by a translator which contains three blanks. Y^b includes three segments to fill each blank in $\bar{Y}$, and Y is the translation after filling $\bar{Y}$ with Y^b. It is worth noting that Y^b contains three special tokens "<eob>", indicating the end of a segment, which correspond to the blanks in $\bar{Y}$, respectively.

$\bar{Y}$:	It is an inherent right __ all measures __ preserve __
Y^b:	that must be upheld by all Member States, and <eob> must be taken to <eob> it. <eob>
Y:	It is an inherent right that must be upheld by all Member States, and all measures must be taken to preserve it.

Figure 2: Main notations of bilingual text-infilling where __ represent a blank based on example in Figure 1.

Model Definition Formally, bilingual text-infilling can be addressed by the following probabilistic model:

$$P(Y^b|X,\bar{Y};\theta) = \prod_t P(y_t^b \mid X, \bar{Y}, Y_{<t}^b; \theta) \quad (2)$$

X<sep>$\bar{Y}$:	所有会员国必须支持这项固有的权利,并且必须采取一切措施来维护这种权利。<sep> It is an inherent right __ all measures __ preserve __
Y^b:	that must be upheld by all Member States, and <eob> must be taken to <eob> it. <eob>

Figure 3: Bilingual text-infilling as the sequence-to-sequence task. The input is "X <sep> $\bar{Y}$" and the output is Y^b.

To implement this model, it is possible to extend the Transformer (Vaswani et al., 2017) by using two encoders (i.e., one is for X, and the other is for $\bar{Y}$) and taking both segment and token positions into account similar to Zhu et al. (2019). However, for simplicity, we instead cast this task as a standard sequence-to-sequence task by format manipulation and employ an NMT model to address it. Specifically, we treat two input sequences X and $\bar{Y}$ as one input sequence "X <sep> $\bar{Y}$", where "<sep>" is a speck token for concatenation, and Y^b as the output sequence, as shown in Figure 3. Then we employ the Transformer model to accomplish this task as the conventional NMT task.

Relation to Previous Work Our method is similar to previous works, including lexically constrained decoding (LCD) (Hokamp and Liu, 2017; Post and Vilar, 2018), MT with soft constraints (Dinu et al., 2019) and code-switch enhanced MT (Song et al., 2019; Chen et al., 2020) in the sense that all of them generate translations based on given constraints which are $\bar{Y}$ in our work. However, LCD imposes hard constraints on beam search during the decoding stage, leading to the suffering of decoding speed. In addition, the other two series of works can not guarantee the constraints $\bar{Y}$ will be in the output translation, and the code-switch method even requires word alignment information whereas human translators do not provide such alignment information in our scenario.

3.2.2 Training via Data Augmentation

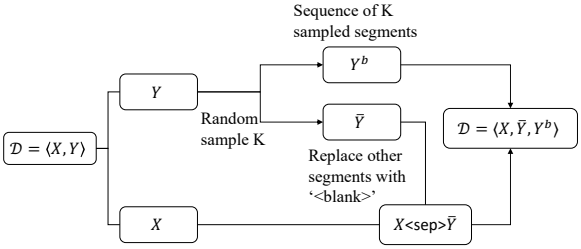

Figure 4: Sampling procedure to get synthetic bilingual text-infilling data.

Typically, to train the model in Eq.(2), one needs to obtain large amount of data consisting of triples $\mathcal{D} = \{\langle X, \bar{Y}, Y^b \rangle\}$. Unfortunately, this is impractical because both $\bar{Y}$ and Y^b are obtained by human translators. To this end, we apply a simple simulation method to obtain $\{\langle X, \bar{Y}, Y^b \rangle\}$ on top of bilingual corpus $\{\langle X, Y \rangle\}$ through random sampling. Specifically, for each oracle target sentence Y in bilingual data, we randomly sample an integer $k \in [0, 5]$ and randomly sample k non-overlapped segments in Y. Then $\bar{Y}$ can be obtained by replacing each remaining segment with "<blank>" in Y, and Y^b can be obtained by wrapping k sampled segments and then joining them together in the same order. This sampling procedure is shown in Figure 4.

Given a training data $\mathcal{D} = \{\langle X, \bar{Y}, Y^b \rangle\}$, it is straightforward to optimize the following objective function according to maximum likelihood estimation:

$$\ell = -\sum_{\langle X, \bar{Y}, Y^b \rangle} \log P(Y^b \mid X, \bar{Y}; \theta)$$

where the model P is defined in Eq. (2).

Models trained on training data $\mathcal{D} = \{\langle X, \bar{Y}, Y^b \rangle\}$ possess the ability to translate sentences with constraints in fill-in-blank format style, but it may lose its strength in translating normal sentences. As a result, for the first iteration in IMT, the initial translation maybe not good as expected, leading to more interactive iterations between humans and machines. One possible solution is to build an additional NMT model to generate the initial translation. Instead, we apply the data augmentation (DA) technique, making our model perform both tasks. For the standard bilingual parallel data $\{\langle X, Y \rangle\}$, we construct its trivial triple data $\mathcal{D}' = \{\langle X, \emptyset, Y \rangle\}$, where $\emptyset$ means that no revised words are provided in $\bar{Y}$ for each bilingual sentence. Then we combine the sampled data D and the trivial data D' as the augmented data to train the model P in our experiments.

Actually, our method is slightly different from classical data augmentation for NMT (Novak et al., 2018; Wang et al., 2018; Li et al., 2019) because the input in the augmented data is different from that for bilingual text infilling task. In addition, because the augmented data is used for bilingual text infilling task and the other data is used for translation task, our training method resembles an instance of multi-task learning (MTL) framework(Caruana, 1997; Dong et al., 2015; Liu et al., 2016; Wang et al., 2020b), where both tasks are modeled by the same Transformer architecture.

3.3 Decoding

The task of bilingual text-infilling is reduced to decode Y^b according to the model P in Eq. (2) via maximum a posteriori (MAP) estimation as follows:

$$\arg\max_{Y^b : \#_b(Y^b) = \#_b(\bar{Y})} P(Y^b \mid X, \bar{Y}; \theta) \quad (3)$$

where $\#_b$ denotes the number of blanks, i.e., $\#_b(Y^b)$ is the number of "<eob>" in Y^b and $\#_b(\bar{Y})$ counts the number of "_" in $\bar{Y}$. The constraints in the above equation are used to guarantee that all blanks in $\bar{Y}$ can be exactly filled by Y^b to obtain a valid translation Y, otherwise, Y^b would lead to an invalid Y.

Theoretically, the constrained optimization in Eq. (3) is more difficult than the unconstrained one in standard NMT decoding. In practice, since the constraint in Eq. (3) is about the number of blanks in Y^b, it is easy to satisfy by extending the standard beam search algorithm. Specifically, in the standard beam search algorithm, one only needs to maintain a number to restore the number of blanks in the partial output Y^b. If this number is equal to $\#_b(\bar{Y})$, Y^b is the final output; otherwise, Y^b should be extended until the constraint is satisfied. As a result, our decoding algorithm is very efficient, and it shares the same complexity as the standard beam search algorithm. In fact, thanks to the powerful Transformer architecture and sufficient training data in our scenario, our model is able to implicitly learn the constraint $\#_b(Y^b) = \#_b(\bar{Y})$ with about 99.39% accuracy in our preliminary experiments. In other words, for almost all sentences, the standard beam search algorithm is able to yield a valid Y, even without explicitly imposing the constraint during decoding.

4 Experiments

Following previous works (Peris et al., 2017b; Cheng et al., 2016; Weng et al., 2019; Li et al., 2020), we experiment on two simulated scenarios and a real-world scenario.

4.1 Experiment Settings

4.1.1 Dataset

We conduct experiments on the English-German dataset (En-De), English-French (En-Fr), and a

	#Raw		#Augmented	
	Train	Valid	Train	Valid
En-De	4M	2,737	8M	5,474
En-Fr	36M	3,003	72M	6,006
Zh-En	2M	2,000	4M	4,000

Table 1: Statistics on three datasets: WMT14 En-De, WMT14 En-Fr, and Zh-En. (Augmented: a combination of the raw bilingual parallel dataset and their corresponding artificial dataset with constructed templates.)

Chinese-English dataset (Zh-En), which includes about 2 million bilingual sentences from the news domain in total. For En-De and En-Fr, the datasets are from WMT14 and we use newstest13 as the valid dataset and use newstest14 filtered by Stanford (Bojar et al., 2014) as the test dataset. For Zh-En, the datasets are the same as Li et al. (2020). We utilize the approach mentioned in Section 3.2 to construct synthetic bilingual parallel sentence pairs based on all the datasets above. To set up the data augmentation strategy, we combine original training datasets and their corresponding synthetic datasets. As Table 1 shows, we get an 8M English-German dataset based on the WMT14 En-De, a 72M English-French dataset based on WMT14 En-Fr, and a 4M Chinese-English dataset based on a Zh-En dataset mentioned above. Valid sets are obtained the same as training datasets.

For all datasets mentioned above, we use Moses toolkit to tokenize and clean data. Besides, we use BPE (Sennrich et al., 2015) to process all the source and target sentences.

4.1.2 System Configurations

We train and evaluate the following systems for comparison.

- **Transformer.** We set Transformer (Vaswani et al., 2017) using fairseq (Ng et al., 2019). The Transformer model is trained on the bilingual datasets: WMT14 En-De, WMT14 En-Fr, and the Zh-En dataset.

- **LCDIMT.** Since LCD-based decoding is widely used in IMT scenario (Weng et al., 2019; Huang et al., 2021), we build an LCD-based IMT as our strong baseline. Because the original LCD algorithm is very slow, we implement its efficient version (Post and Vilar, 2018) which achieves comparable translation quality to the original version but 5x speedup in decoding.

- **Our System.** Our BiTiIMT model is based on the base Transformer architecture and trained on the synthetic datasets mentioned in Section 3.3.

All baselines (Transformer and LCDIMT) and the BiTIIMT models are based on the architecture with $d_{model} = 512$, $d_{hidden} = 2048$, $n_{heads} = 8$, $n_{layers} = 6$, and $p_{dropout} = 0.1$. We use the Adam Optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.98$ to train our models. We adopt a warm-up of 10,000 steps and set the initial learning rate to 0.0007. We set the maximum tokens in batch to 4096, and we share both source and target embeddings for all models. Training stops until the maximum update is 400,000 and the checkpoint used for testing is selected according to its performance on the valid dataset. We train all models on 16 NVIDIA V100 Tensor Core GPUs. We use a beam size of 10 throughout our experiments.

4.1.3 Evaluation Metrics

Following prior work (Cheng et al., 2016; Weng et al., 2019; Zhao et al., 2020; Huang et al., 2021), two criteria are used to evaluate INMT systems: one is translation quality and the other is efficiency to yield the translation. We employ BLEU (Papineni et al., 2002) to measure translation quality, and human editing cost to measure efficiency, which is calculated as the edit distance by counting deletions on word level and insertions on char level. In addition, we take the decoding time into account because it is directly related to the latency for human translators, which is critical for user experience.

4.2 Simulated Scenario

We conducted two different simulated experiments, including IMT with a single iteration and IMT with multiple iterations, to validate the effectiveness of our method in terms of translation quality and editing and decoding costs mentioned above.

4.2.1 IMT with a Single Iteration

Since IMT with a single iteration can be seen as machine translation with lexical constraints where human interactions are considered as constraints, we first conduct an experiment to evaluate the performance of BiTIIMT by following Hokamp and

		Number of constraint segments				
		1	2	3	4	5
En-De	Transformer			27.36 / 1×		
	LCDIMT	29.56 / 1.22×	31.33 / 1.68×	32.98 / 1.73×	36.18 / 1.79×	38.78 / 1.88×
	BiTIIMT	**32.51 / 0.93**×	**37.86 / 0.9**×	**43.01 / 0.83**×	**47.09 / 0.81**×	**52.33 / 0.73**×
En-Fr	Transformer			39.9 / 1×		
	LCDIMT	44.53 / 1.56×	47.76 / 1.82×	48.35 / 1.83×	48.64 / 1.84×	48.96 / 1.85×
	BiTIIMT	**46.02 / 0.99**×	**51.26 / 0.93**×	**51.96 / 0.91**×	**52.66 / 0.9**×	**53.18 / 0.89**×
Zh-En	Transformer			46.71 / 1×		
	LCDIMT	47.83 / 1.35×	49.18 / 1.59×	49.34 / 1.83×	49.13 / 2.02×	49.73 / 2.08×
	BiTIIMT	**49.61 / 1.08**×	**51.62 / 1.08**×	**53.86 / 0.99**×	**56.05 / 0.97**×	**56.59 / 0.95**×

Table 2: **BLEU / Relative decoding time cost w.r.t Transformer baseline** for five settings with 1 to 5 constraint segments on WMT14 En-De, WMT14 En-Fr, and Zh-En datasets. For each setting, the boldface denotes the top BLEU score and the best time cost among all systems.

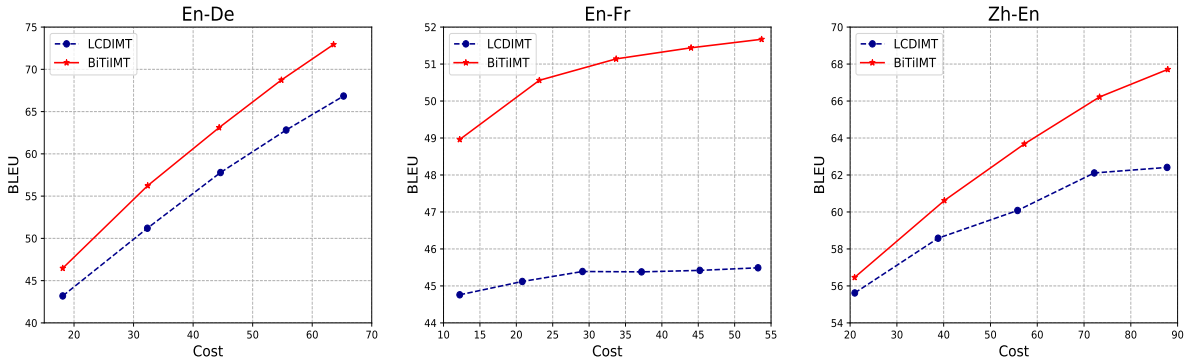

Figure 5: **BLEU** and **Interactive cost** comparison on WMT14 En-De, WMT14 En-Fr, and Zh-En datasets.

Liu (2017); Post and Vilar (2018). We use the reference as oracle to sample constraints for all datasets and we consider five settings: they include gradually increasing constraint segments from one to five. In more detail, we randomly add a constraint segment with a random length between 1 to 3 for each setting. To ensure fairness, the constraints provided to both BiTIIMT and LCDIMT are exactly the same. We measure decoding cost by using the time required to translate the whole test set with a batch size of one (excluding the time of model loading).

As shown in Table 2, with only one constraint segment, both BiTIIMT and LCDIMT obtain substantial improvements compared with Transformer and BiTIIMT significantly outperforms LCDIMT with a margin of +2.2, +4.63, and +1.12 BLEU points for En-De, En-Fr, and Zh-En tasks, respectively. This finding clearly verifies our hypothesis: BiTIIMT indeed makes full use of the constraint segments and thus yields better translations than LCDIMT.

Table 2 also reports the results of relative decoding cost with respect to the Transformer baseline. As we can see, in the first setting with one constraint segment, BiTIIMT achieves modest speedup in decoding time compared with the Transformer baseline. With the growing number of constraint segments, BiTIIMT keeps reducing the decoding time, and it gives 0.27x decrease in decoding time cost on En-De when running on five constraint segments. Meanwhile, the time cost of LCDIMT keeps growing, and it is almost twice as that of the baseline in the 5th setting. Similar results can be found on En-Fr and Zh-En datasets. It is worth noting that the decoding efficiency of LCDIMT seems not an issue for one iteration. In fact, decoding efficiency indeed is a severe issue for multiple iterations as in real-world scenarios, where more constraint segments are involved especially at the late stage of iterations.

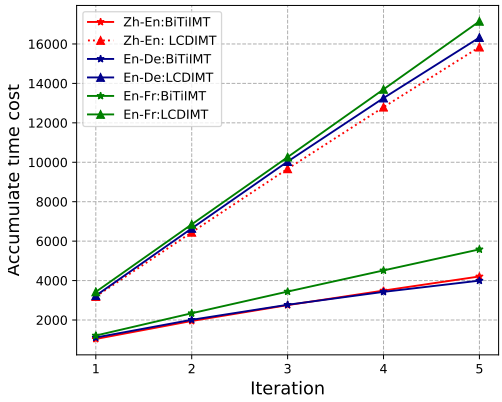

Figure 6: Accumulated decoding time cost comparision on WMT14 En-De, WMT14 En-Fr, and Zh-En datasets.

4.2.2 IMT with Multiple Iterations

We now turn to the evaluation of BiTIIMT in a simulated IMT scenario where multiple iterations of interactions are allowed. To simulate the human interactive process of IMT, at each iteration, we use the reference as oracle and match the oracle with a translation from each system to calculate $\bar{Y}$. Specifically, we delete unmatched words in the translation as simulated deletion and add a word from oracle as a simulated revision. By using the edit distance algorithm, which includes only deletion and revision operators, we can figure out $\bar{Y}$ given a translation and its reference. We use the edit cost mentioned in Section 4.1.3 to qualify human interaction cost.

Figure 5 shows BLEU scores and interaction costs along with all iterations on En-De, En-Fr, and Zh-En datasets. As expected, BLEU scores consistently increase with the increase of interaction costs. As we can see on En-De, BiTIIMT obtains improvements of about 5 BLEU points over the baseline LCDIMT when using similar human interaction costs. The BLEU gap between BiTIIMT and LCDIMT is further enlarged in the late stage of the interactive process. Results on En-Fr and Zh-En give similar conclusions. These facts show that BiTIIMT outperforms LCDIMT as it can reduce the human interaction cost to get a satisfying translation.

Figure 6 reports the accumulated decoding time cost on three datasets. As we can see, results on three datasets give similar conclusions with the conclusions in section 4.2.2 that BiTIIMT has a lower decoding time cost compared to LCDIMT for all interactive iterations. Furthermore, the accumulated decoding time cost of BiTIIMT after 5 iterations is lower than the time cost of LCDIMT with only 2 iterations. The facts indicate that BiTIIMT has its outstanding advantage in efficiency.

4.3 Real-world Scenario

We conduct two kinds of IMT experiments to validate the effectiveness of BiTIIMT in a real-world scenario. First, we use the post-editing data from a valid dataset of WMT21 Automatic Post-Editing Shared Task on En-De (Sharma et al., 2021), where the words are edited by humans are natural constraints instead of simulated constraints. By using these constraints, we compare the proposed IMT method with baselines involve a single iteration of human-machine interactions. Results in table 3 show that BiTIIMT can obtain much better translations than LCDIMT. Since in this dataset there are many constraints edited by humans and the post-edited translation is used as the reference, the improvements of BiTIIMT over LCDIMT are substantial in terms of BLEU (up to 9 BLEU points).

Furthermore, we conduct another real-world experiment that involves multiple iterations of human-machine interactions. Specifically, we randomly sample 200 sentences from the Zh-En test set and then ask two professional human translators to interact with both systems. Translators are asked to do interactions (deletions, revisions, and insertions) multiple times until they get a satisfactory translation. We compared LCD-based and BiTI-based IMT systems on averaged BLEU, averaged decoding time cost, and averaged editing cost of deletions, revisions, and insertions supplied by human translators. As shown in Table 4, BiTIIMT can reach higher BLEU points with less decoding time cost, editing cost, as well as fewer interaction rounds. Note that our BLEU gains over LCDIMT in Table 4 are relatively small compared with those in Table 3. One main reason is that the constraints edited by human translators may not appear in the reference translation.

4.4 Analysis

Effect of Data Augmentation As the description in Section 3.2.2, we train our model (#Augmented in Table 5) on the augmented data which includes synthetic bilingual data $\mathcal{D} = \{\langle X, \bar{Y}, Y^b \rangle\}$ and their corresponding bilingual parallel data $\mathcal{D}' = \{\langle X, \emptyset, Y \rangle\}$ by data augmentation. For com-

Method	BLEU
Transformer	41.92
LCDIMT	46.96
BiTIIMT	**56.02**

Table 3: Results on the real-world dataset from WMT21 En-De Automatic Post-Editing shared task.

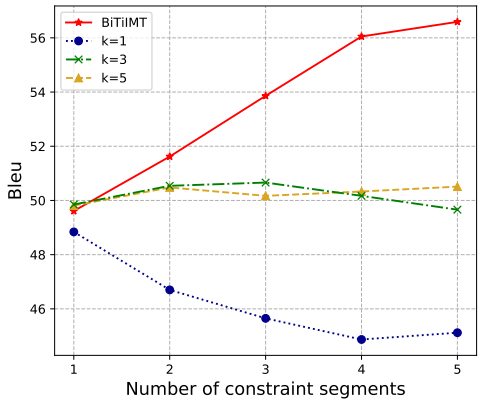

Figure 7: Effectiveness of Random Sampling Strategy. "k" denotes the models trained on augmented Zh-En datasets, which includes k constraint segments. BiTiIMT is trained on a dataset with a random number of constraint segments. The x-coordinate denotes testing with a different number of constraint segments.

parison, we train two additional models: one is on $\mathcal{D}$ only (#Synthetic Only in Table 5) and the other is trained on $\mathcal{D}'$ (#Raw in Table 5). We compare all these three models according to their BLEU on the Zh-En dataset. Note that in this experiment, we do not provide any human interactions to all three models, i.e., $\bar{Y} = \emptyset$, during testing.

Table 5 summarizes the results of the three models. The model trained on the only synthetic dataset almost collapses on automatic machine translation task: it is worse than # Raw model by a substantial margin of 18 BLEU points. In addition, it can be found that the model trained on the augmented dataset achieves +0.68 BLEU improvements over # Raw model. Although such improvements are not that large, it still shows that data augmentation or the application of multiple task learning plays a critical role in making BiTiIMT successful.

Effect of Random Sampling Strategy When training the model for BiTIMT in section 3.2.2, we employ a random strategy to sample $\bar{Y}$ such that it contains a random number of constraint segments. We are also curious about the effect of such a random strategy. In comparison, we fix the number of constraint segments and then train three models for $k = \{1, 3, 5\}$. Under all five settings as in Section 4.2.1, we compare BiTIIMT with these three models on the Zh-En task. Figure 7 shows that, by training on data with a diverse number of constraints, our model achieves increasing BLEU on all five settings while the BLEU of the model ($k = 1$) gradually decreases and another two models could not give continuous improvement. Results suggest that our random sampling strategy assists BiTIIMT to translate with a various number of constraints.

5 Related Work

Since the period of statistical machine translation (SMT), IMT has been widely exploited to reduce human effort by using human's feedback to improve translation quality (Foster et al., 1997; Langlais et al., 2000; Simard et al., 2007; Barrachina et al., 2009; González-Rubio et al., 2013; Cheng et al., 2016; Li et al., 2021). Recently, with the development of NMT (Bahdanau et al., 2015; Vaswani et al., 2017), researchers turned to employ IMT on it (Hokamp and Liu, 2017; Wang et al., 2020a).

A classical type of IMT uses a left-to-right sentence completing framework proposed in Langlais et al. (2000), in which human translators can only revise the translation generated by models from left to right. Generally, the text portion from the beginning to the current modified part is called prefix, and the system will generate a new translation based on the given prefix (Sanchis-Trilles et al., 2014; Peris et al., 2017a; Knowles and Koehn, 2016).

Cheng et al. (2016) propose a pick-revise framework that enables translators to do revisions on arbitrary positions to improve efficiency. Huang et al. (2021) allow users to make any interaction at random positions by using LCD algorithms (Hokamp and Liu, 2017; Post and Vilar, 2018) in the decoding stage which can integrate lexical constraints into translation. However, LCD can not achieve a win-win result in terms of decoding speed and translation quality. Weng et al. (2019) propose a bidirectional IMT framework on top of LCD, which could fix mistakes by using two constrained decoding procedures with opposite directions. However, it needs to train two decoders, and in each con-

Methods	BLEU	Decoding Time Cost	Editing Cost	Rounds
Transformer	45.46	0.92	-	-
LCDIMT	51.36	1.44	19.79	1.46
BiTIIMT	**53.48**	**0.27**	**15.815**	**1.27**

Table 4: Results in the real-world IMT scenario for different methods.

Dataset	BLEU
#Raw	46.71
#Synthetic Only	28.96
#Augmented	**47.39**

Table 5: Results of routine machine translation task over models trained on different dataset settings. (Raw: Zh-En dataset; Synthetic Only: artificial dataset with constraint segments based on Zh-En; Augmented: a combination of the raw dataset and synthetic dataset).

strained decoding, the model can only use part of the constraints supplied by translators, making it inefficient both in using human knowledge and decoding speed. Instead, BiTIIMT puts all constraints into a template as part of the input, which makes it possible for the model to use all human knowledge and meanwhile fix mistakes automatically in the whole sentence.

Other works (Alkhouli et al., 2019; Song et al., 2020; Chen et al., 2021) apply alignment information to improve the decoding efficiency of LCD. Alkhouli et al. (2019) use alignment extracted by vanilla transformer, which is poor as argued by Garg et al. (2019). Song et al. (2020) apply an external aligner to train the alignment module. These works can only perform constrained decoding based on a constraint pair, which means a burden for human translators.

In order to address the issue of decoding speed for LCD, some works use a non-autoregressive approach to integrate constraints. Susanto et al. (2020) propose Levenshtein Transformer (Gu et al., 2019) to inject terminology constraints at inference time. Xu and Carpuat (2021) propose a novel re-position operator to replace deletion in Levenshtein Transformer to exploit lexical constraints more effectively and efficiently. However, non-autoregressive models are still worse than autoregressive models in translation quality currently. Compared to these efforts in NAT, BiTIIMT is essentially based on an auto-regressive translation model.

6 Conclusion

Traditional IMT systems often use LCD to incorporate manually revised words into translations. In this paper, we propose BiTIIMT, a novel IMT method that outperforms LCD-based IMT in both translation quality and efficiency. The key to BiTIIMT is the bilingual text-infilling task which extends text-infilling from a monolingual setting to a bilingual one. We cast this task as a sequence-to-sequence task and propose a simple yet effective solution to address it. Experiments show that BiTIIMT achieves a significantly improved efficiency in the area of IMT.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments. Shujian Huang and Lemao Liu are the corresponding co-authors. This work is supported by National Science Foundation of China (No. U1836221, 6217020152).

References

Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. 2019. On The Alignment Problem In Multi-Head Attention-Based Neural Machine Translation. *Proceedings of the Third Conference on Machine Translation: Research Papers*, 1:177–185.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan-Miguel Vilar. 2009. Statistical Approaches to Computer-Assisted Translation. *Computational Linguistics*, 35(1):3–28.

Mathias Berglund and K Leo. Bidirectional recurrent neural networks as generative models. *Advances in Neural Information Processing Systems*, pages 1–10.

Ondřej 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.

Rich Caruana. 1997. Multitask Learning. *Machine Learning*, 28(1):41–75.

Guanhua Chen, Yun Chen, and Victor O K Li. 2021. Lexically Constrained Neural Machine Translation with Explicit Alignment Guidance. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12630–12638.

Guanhua Chen, Yun Chen, Yong Wang, and Victor O K Li. 2020. Lexical-Constraint-Aware Neural Machine Translation via Data Augmentation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, {IJCAI-20}*, pages 3587–3593. International Joint Conferences on Artificial Intelligence Organization.

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

Georgiana Dinu, Marcello Federico, and Yaser Al-onaizan. 2019. Training Neural Machine Translation To Apply Terminology Constraints. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732.

William Fedus, Ian Goodfellow, and Andrew M Dai. 2018. Maskgan: better text generation via filling in the_. *arXiv preprint arXiv:1801.07736*.

George Foster, Pierre Isabelle, and Pierre Plamondon. 1997. Target-Text Mediated Interactive Machine Translation. *Machine Translation*, 12(1):175–194.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. *arXiv preprint arXiv:1909.02074*.

Jesús González-Rubio, Daniel Ortiz-Martínez, José Miguel Benedí, and Francisco Casacuberta. 2013. Interactive machine translation using hierarchical translation models. In *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, October, pages 244–254.

Jiatao Gu, Changhan Wang, and Jake Zhao. 2019. Levenshtein transformer. *CoRR*, abs/1905.11006.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, volume 1, pages 1535–1546.

Guoping Huang, Lemao Liu, Xing Wang, Longyue Wang, Huayang Li, Zhaopeng Tu, Chengyan Huang, and Shuming Shi. 2021. TranSmart: A Practical Interactive Machine Translation System. *arXiv preprint arXiv:2105.13072*, pages 1–21.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam- A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, pages 1–15.

Rebecca Knowles and Philipp Koehn. 2016. Neural Interactive Translation Prediction. *Proceedings - AMTA 2016: 12th Conference of the Association for Machine Translation in the Americas*, 1:107–120.

Philippe Langlais, George Foster, and Guy Lapalme. 2000. Transtype: a computer-aided translation typing system. In *ANLP-NAACL 2000 Workshop: Embedded Machine Translation Systems*.

Guanlin Li, Lemao Liu, Guoping Huang, Conghui Zhu, and Tiejun Zhao. 2019. Understanding data augmentation in neural machine translation: Two perspectives towards generalization. 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)*, pages 5689–5695, Hong Kong, China. Association for Computational Linguistics.

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*, PP:1.

Huayang Li, Lemao Liu, Guoping Huang, and Shuming Shi. 2021. GWLAN: General word-level AutocompletioN for computer-aided translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4792–4802, Online. Association for Computational Linguistics.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102, Osaka, Japan. The COLING 2016 Organizing Committee.

Nathan Ng, David Grangier, Michael Auli, and Sam Gross. 2019. A Fast, Extensible Toolkit for Sequence Modeling. *arXiv preprint arXiv:1904.01038*.

Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. 2018. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*.

Kishore Papineni, Salim Roukos, Todd Ward, and Weijing Zhu. 2002. BLEU : a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, (July):311–318.

Álvaro Peris, Luis Cebrián, and Francisco Casacuberta. 2017a. Online Learning for Neural Machine Translation Post-editing. *arXiv preprint arXiv:1706.03196*, pages 1–12.

Álvaro Peris, Miguel Domingo, and Francisco Casacuberta. 2017b. 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. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1(Table 1):1314–1324.

Germán Sanchis-Trilles, Vicent Alabau, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin L Hill, Philipp Koehn, Luis A Leiva, Bartolomé Mesa-Lao, Daniel Ortiz-Martínez, Herve Saint-Amand, Chara Tsoukala, and Enrique Vidal. 2014. Interactive translation prediction versus conventional post-editing in practice: a study with the CasMaCat workbench. *Machine Translation*, 28(3):217–235.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Abhishek Sharma, Prabhakar Gupta, and Anil Nelakanti. 2021. Adapting neural machine translation for automatic post-editing. In *Proceedings of the Sixth Conference on Machine Translation*, pages 315–319, Online. Association for Computational Linguistics.

Michel Simard, Nicola Ueffing, Pierre Isabelle, and Roland Kuhn. 2007. Rule-based translation with statistical phrase-based post-editing. *Proceedings of the Second Workshop on Statistical Machine Translation*, (June):203–206.

Kai Song, Kun Wang, Heng Yu, Yue Zhang, Zhongqiang Huang, Weihua Luo, Xiangyu Duan, and Min Zhang. 2020. Alignment-enhanced transformer for constraining NMT with pre-specified translations. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pages 8886–8893.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:449–459.

Raymond Hendy Susanto, Shamil Chollampatt, and Liling Tan. 2020. Lexically constrained neural machine translation with Levenshtein transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3536–3543, Online. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *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. *Advances in Neural Information Processing Systems*, 2017-December(Nips):5999–6009.

Qian Wang, Jiajun Zhang, Lemao Liu, Guoping Huang, and Chengqing Zong. 2020a. 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, Suzhou, China. Association for Computational Linguistics.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium. Association for Computational Linguistics.

Yiren Wang, ChengXiang Zhai, and Hany Hassan Awadalla. 2020b. Multi-task learning for multilingual neural machine translation. *CoRR*, abs/2010.02523.

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

Weijia Xu and Marine Carpuat. 2021. EDITOR: An Edit-Based Transformer with Repositioning for Neural Machine Translation with Soft Lexical Constraints. *Transactions of the Association for Computational Linguistics*, 9:311–328.

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. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pages 9660–9667.

Wanrong Zhu, Zhiting Hu, and Eric P. Xing. 2019. Text infilling. *arXiv preprint arXiv:1901.00158*.