Lexical Translation Inconsistency-Aware Document-Level Translation Repair

Zhen Zhang[1], **Junhui Li**[1]*, **Shimin Tao**[2], **Hao Yang**[2]
[1]School of Computer Science and Technology, Soochow University, Suzhou, China
[2]Huawei Translation Services Center, Beijing, China
zzhang99@stu.suda.edu.cn; lijunhui@suda.edu.cn
{taoshimin,yanghao30}@huawei.com

Abstract

Following the idea of "one translation per discourse", in this paper we aim to improve translation consistency via document-level translation repair (DocRepair), i.e., automatic post-editing on translations of documents. To this end, we propose a lexical translation inconsistency-aware DocRepair to explicitly model translation inconsistency. First we locate the inconsistency in automatic translation. Then we properly provide translation candidates for those inconsistency. Finally, we propose lattice-like input to properly model inconsistent phrases and their candidates. Experimental results on three document-level translation datasets show that based on G-Transformer, a state-of-the-art document-to-document (Doc2Doc) translation model, our Doc2Doc DocRepair not only achieves improvement in translation quality in BLEU scores, but also greatly improves lexical translation consistency.

1 Introduction

Although neural machine translation (NMT) has made remarkable progress (Bahdanau et al., 2015; Vaswani et al., 2017), sentence-level NMT still suffers from the serious problem of lexical translation inconsistency due to the lack of inter-sentence context. To better model inter-sentence context, previous studies in document-level NMT propose various context-aware models which use sentences in the wider-document context, thus implicitly learning discourse correlations as a by-product of optimising an NMT model (Maruf et al., 2022). However, as these models rarely try to model discourse phenomena explicitly, there still exist much rooms for improvement on discourse phenomena. In this paper, we follow up the idea of "one translation per discourse" (Merkel, 1996; Carpuat, 2009; Türe et al., 2012; Guillou, 2013; Khotaba and Tarawneh, 2015) and focus on lexical translation consistency, which is one of the most serious issues in document-level (Chinese-to-English) translation (Kang et al., 2021; Lyu et al., 2021b). Our goal is to improve translation consistency via document-level translation repair (DocRepair for short (Voita et al., 2019)), i.e., automatic post-editing on translations of documents.

Figure 1 shows an example of an input document and its translation from both state-of-the-art sentence-level and document-level NMT models.

Source
#1: 孙燕姿 非洲 送 关爱 遇 沙暴 挨 石头 被 划伤 #13: ... 孙燕姿 经历 当地 难得一见 的 沙尘暴 ... #17: 之后 听 当地人 解释 沙尘暴 是 好运 的 象征 ... #20: ... 请 孙燕姿 坐 上 骆驼 接受 当地人 欢呼
Sentence-level NMT
#1: sun yen-tzu in africa ... and gets injured by stones ... #13: ... sun yanzi experienced rare sandstorms in the area ... #17: after hearing the locals explain that dust storms are a symbol of good luck ... #20: ... invite sun yanzi to sit on the camel and receive cheers from the local people ...
Document-level NMT
#1: sun yen-tzu sent ... she was hit by sand and rocks ... #13: ... sun yanzi experienced the rare dust storms in the area ... #17: after hearing the locals explain n that sand storms were symbols of good fortune ... #20: ... asking sun yantzu to sit on the camel and receive a cheer from the local people ...
Reference
#1: sun yanzi delivers love to africa ... #13: ... sun yanzi experienced a sandstorm that was rarely seen in the area #17: she later heard the locals explaining that the sandstorm is a sign of good luck ... #20: ... and invited sun yanzi to sit on the camel to receive the cheers of the locals

Figure 1: An example of document-level Chinese-to-English translation from the test set NIST 2008, where the source words like 孙燕姿/*sun_yan_zi*, 沙尘暴/*sha_chen_bao* and 当地人/*dang_di_ren* are inconsistent in the sentence-level and document-level NMT systems but tend to be consistent in the reference.

*Corresponding author: Junhui Li.

Findings of the Association for Computational Linguistics: ACL 2023, pages 12492–12505
July 9-14, 2023

The source words like 孙燕姿*/sun_yan_zi*, 沙尘暴*/sha_chen_bao* and 当地人*/dang_di_ren*, occurring two or more times within the source document, unexpectedly get different translations while they are translated consistently in its reference (human translation). For example, person name 孙燕姿*/sun_yan_zi* is translated into *sun yen-tzu* and *sun yanzi* by sentence-level NMT. Such inconsistent translations, however, tend to confuse readers. Moreover, even some context-aware document-level NMT models like G-Transformer (Bao et al., 2021) could not well alleviate this phenomenon as shown in the figure.

Very few studies in document-level NMT explicitly encourage lexical translation consistency. Lyu et al. (2021b) obtain a word link for each source word in a document and exchange their context information in encoding by using an auxiliary loss to constrain their translation being consistent. Kang et al. (2021) and Lyu et al. (2022) both construct source-side lexical chains, and use different approaches to learn (or model) translations for tokens within the same lexical chain. Different from above studies which encourage translation consistency in the translation process, in this paper we aim to improve translation consistency via DocRepair. Different from Voita et al. (2019) which implicitly learns inconsistency within document translation, we propose a lexical translation inconsistency-aware DocRepair model to explicitly correct translation inconsistency. Given automatic translation $\mathcal{T}$ of a document $\mathcal{S}$, either from sentence-level NMT or document-level NMT, this is done by the following steps. First, in translation $\mathcal{T}$ we locate inconsistent phrases, each of which consists of one or more consecutive tokens. Then, we provide translation candidates for those inconsistent phrases. Finally, we adapt G-Transformer, a state-of-the-art document-to-document translation model, to repair document-level translation $\mathcal{T}$ equipped with inconsistent phrases and their candidates.

Overall, we make the following contributions.

- Based on G-Transformer (Bao et al., 2021), a state-of-the-art document-to-document (Doc2Doc) NMT model, we extend Voita et al. (2019) and build a strong Doc2Doc DocRepair baseline model.

- We propose a novel approach to repair translation of documents with explicit aim of correcting translation inconsistency. In this approach, we use lattice-like input to model inconsistent phrases and their candidate translations.

- Experimental results in three document-level translation datasets show that given translation from either sentence-level or document-level NMT models, our DocRepair approach not only improves translation performance in BLEU, but also greatly improves lexical translation consistency.

2 Approach

2.1 Problem Statement

Formally, we use $\mathcal{S} = \{S^{(k)}\}|_{k=1}^{K}$ to denote a source-side document composed of K source sentences, and assume each source-side sentence $S^{(k)} = \{s_i^{(k)}\}|_{i=1}^{I}$ consists of I words. Likewise, we use $\mathcal{T} = \{T^{(k)}\}|_{k=1}^{K}$ to denote its automatic translation and $T^{(k)} = \{t_j^{(k)}\}|_{j=1}^{J}$ to represent the automatic translation of the k-th sentence in $\mathcal{S}$. Finally, we use $\mathcal{Y} = \{Y^{(k)}\}|_{k=1}^{K}$ and $Y^{(k)} = \{y_m^{(k)}\}|_{m=1}^{M}$ to denote the corresponding target-side gold document and the gold translation of the k-th sentence, respectively.

Therefore, assuming that the repair is done in a left-to-right way, we can decompose the document-level repair probability as

$$\mathrm{P}\left(\mathcal{Y}|\mathcal{T},\mathcal{S}\right) = \prod_{k=1}^{K} \mathrm{P}\left(Y^{(k)}|T^{(k)}, S^{(k)}, Y^{(<k)}, \mathcal{T}^{-k}, \mathcal{S}^{-k}\right), \tag{1}$$

where k is the index of the current sentence, $\mathcal{T}^{-k}$ (or $\mathcal{S}^{-k}$) represents all other sentences in $\mathcal{T}$ (or $\mathcal{S}$), and $Y^{(<k)}$ represents the translations ahead of the current sentence.

If the source document S is totally ignored in the repair, then the task could be viewed as monolingual DocRepair (Voita et al., 2019) and Eq. 1 can be simplified as

$$\mathrm{P}\left(\mathcal{Y}|\mathcal{T}\right) = \prod_{k=1}^{K} \mathrm{P}\left(Y^{(k)}|T^{(k)}, Y^{(<k)}, \mathcal{T}^{-k}\right), \tag{2}$$

which *translates* a document $\mathcal{T}$ in target-side language into another document $\mathcal{Y}$ in the same language. However, totally ignoring source-side knowledge from $\mathcal{S}$ would make it hard for a monolingual DocRepair model to implicitly detect the inconsistency inside $\mathcal{T}$. By only looking the sentence-level NMT output in Figure 1, for example, it is hard to tell that *sun yen-tzu* and *sun yanzi* are inconsistent phrases.

Therefore, we make use of source-side document $\mathcal{S}$ to locate the inconsistency in $\mathcal{T}$ (Section 2.2). For

each inconsistent phrase, we provide a translation candidate list (Section 2.3), which is extracted from $\mathcal{T}$. Being aware of inconsistent phrases, we adapt G-Transformer (Bao et al., 2021) with lattice-like input (Lai et al., 2021) as our Doc2Doc DocRepair model (Section 2.4). Overall, in this paper we approximate the DocRepair probability as

$$\mathrm{P}\left(\mathcal{Y}|\mathcal{T},\mathcal{S}\right)=\prod_{k=1}^{K}\mathrm{P}\left(Y^{(k)}|T^{(k)},Y^{(<k)},\mathcal{T}^{-k},\mathrm{ctx}\left(\mathcal{S},\mathcal{T}\right)\right), \tag{3}$$

where $\mathrm{ctx}\left(\mathcal{S},\mathcal{T}\right)$ returns the inconsistent phrases in $T^{(k)}$ and their respective candidate list.

2.2 Locating Inconsistency in Translation

In translation $\mathcal{T}$, we say a phrase is inconsistent if its counterpart in the source side repeats two or more times in $\mathcal{S}$ and has different translations in $\mathcal{T}$.

Given a source document $\mathcal{S}$, we follow Lyu et al. (2022) and extract N lexical chains $\mathcal{C}=\{C^i\}|_{i=1}^{N}$. Each lexical chain $C^i=\{w^i,\left(a_l^i,b_l^i\right)|_{l=1}^{L}\}$ records all positions of word w^i repeated L times ($L\geq 2$) in document $\mathcal{S}$, where a and b indicate the sentence index and word index of a position, respectively. Then we obtain C^i's translation $CT^i=\left(ct_1^i,\cdots,ct_L^i\right)$ according to word alignment between sentence pairs in $(\mathcal{S},\mathcal{T})$, where ct_l^i could be a phrase.[1] Therefore, if there exist two entries in CT^i which are not consistent, then we say source word w_i is an inconsistency trigger and $ct_l^i\in CT^i$ is an inconsistent phrase in translation $\mathcal{T}$.[2] We traverse all lexical chains to obtain all inconsistency phrases in $\mathcal{T}$.

Taking the sentence-level NMT output in Figure 1 as an example, we extract a lexical chain for source word 孙燕姿*/sun_yan_zi* as it appears three times in the document.[3] Then according to the result of word alignment, we obtain its translation CT = (*sun yen-tzu*, *sun yanzi*, *sun yanzi*). Since there exist inconsistency between phrases *sun yen-tzu* and *sun yanzi*, both *sun yen-tzu* and *sun yanzi* in the 1st, 13th, and 20th sentences are inconsistency phrases. Similarly, *sandstorms* and *dust storms* in the 13th and the 17th sentences, *locals* and *local people* in the 17th and 20th sentences are inconsistency phrases, which are related to source-side inconsistency triggers 沙尘暴*/sha_chen_bao* and 当地人*/dang_di_ren*, respectively.

2.3 Obtaining Candidates for Inconsistency

Once we have located inconsistency in translation $\mathcal{T}$, we further explicitly provide a candidate set of other possible translations in $\mathcal{T}$ for the inconsistency. Here we hope that the candidate set would provide a resolution to the inconsistency.

If source word w^i of the i-th lexical chains C^i is an inconsistency trigger, we provide a translation candidate set from its translation CT^i. Each entry in the set is associated with a weight indicating the translation probability from w_i. As in sentence-level NMT output of Figure 1, the translation candidate set of inconsistency trigger 孙燕姿*/sun_yan_zi* is {*sun yen-tzu*: 1/3, *sun yanzi: 2/3*}, where 1/3 and 2/3 are translation probability. Likewise, the translation candidate sets of 沙尘暴*/sha_chen_bao* and 当地人*/dang_di_ren* are {*sandstorms*: 1/2, *dust storms*: 1/2} and {*locals*: 1/2, *local people*: 1/2}, respectively.

2.4 Lexical Translation Inconsistency-Aware DocRepair

2.4.1 Sentence To Word Lattice

So far, we provide target-side translation $\mathcal{T}$ with inconsistent phrases and their corresponding translation candidate set. To let the DocRepair model be aware of inconsistency and potential resolution, we follow Lai et al. (2021) and propose word lattice-like input for DocRepair.

As shown in the bottom-right corner of Figure 2, a word lattice is a directed acyclic graph, where the nodes are positions in the sentence, and each directed edge represents a word. In particular, we replace inconsistent phrases with their corresponding candidate sets. As shown, word lattice-like input consumes all entries in the candidate set and even the source-side trigger word so that models could explicitly exploit the potential resolutions to the inconsistency. For those words without consistency issue, such as *experienced* and *rare* in the figure, they are essentially on the path from the beginning word *[BOS]* to the end word *[EOS]*. The challenges to model the lattice-like inputs include: 1) encoding the lattice tokens while preserving lattice structures (Lai et al., 2021); and 2) differentiating translation candidates with different quality. Next we present our solutions to the two challenges.

[1] We constrain the target-side aligned words to be continuous.

[2] When extracting lexical chains, we use the SnowballStemmer package in NLTK toolkit to stem the source words for eliminating morphological differences if necessary. We also stem target-side words and filter out function words in CT^i.

[3] Here we simply assume the example in Figure 1 as a full document for better readability.

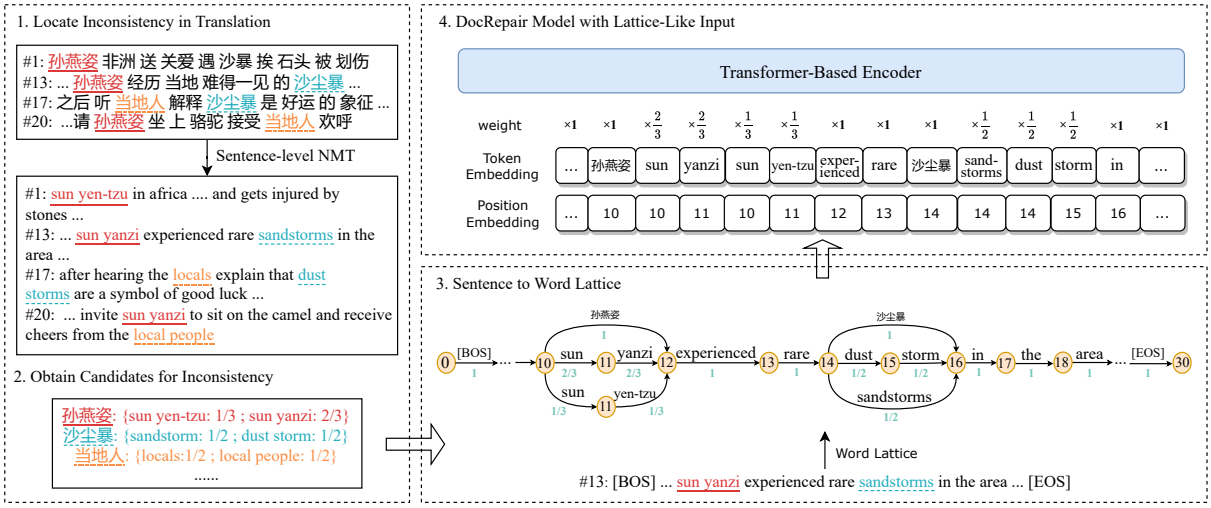

Figure 2: Illustration of our proposed approach.

Token Lattice Position. We assign each node in the lattice graph with a lattice position, whose value is its longest distance from the beginning word *[BOS]*, i.e., the number of nodes in between. Then we set the position of a token as the position of its preceding node. For example, the position values for *dust* and *storm* are 14 and 15, respectively.

Token Weight. According to the type of token, we set token weight differently.

- For those tokens without inconsistency issue, we set their weight as 1.0.
- For tokens of source-side trigger words, like 孙燕姿*/sun_yan_zi* and 沙尘暴*/sha_chen_bao*, we set their weight as 1.0, too.
- For tokens in candidate sets, we set their value as its corresponding translation candidate's probability. For example, in the translation candidate set of the trigger word 孙燕姿*/sun_yan_zi*, {*sun yen-tzu*: 1/3, *sun yanzi: 2/3*}, we set the weight for tokens in *sun yen-tzu* as 1/3 while tokens in *sun yanzi* as 2/3.

2.4.2 DocRepair Model with Lattice-Like Input

As shown in the up-right corner of Figure 2, we linearize a lattice graph into a sequence with pre-prepared lattice position. The input to the encoder is

$$H^0 = [\,\mathrm{WE}(X) + \mathrm{PE}(X)\,] \odot \mathrm{Weight}(X), \qquad (4)$$

where X is the lattice-like input, $\mathrm{WE}(\cdot)$ and $\mathrm{PE}(\cdot)$ return word embedding and sinusoidal positional embedding, respectively. $\mathrm{Weight}(\cdot)$ returns a weight vector for tokens in X.

Different from Voita et al. (2019) which use vanilla Transformer as the DocRepair model, we alternatively choose G-Transformer (Bao et al., 2021) as the base model. G-Transformer is a Doc2Doc translation model which views the source document and target document as long sequences. It uses combined attention, i.e., local attention and global attention to both focus on current sentence and extract contextual information from other sentences. More importantly, it could recover sentence-level translation from the long output. It achieves state-of-the-art performance in document-level translation. For more details, please refer to Bao et al. (2021).

3 Training and Evaluation Metric

3.1 Training

The training consists of two stages: we first pre-train our Doc2Doc DocRepair model on pseudo document-level instances; then fine-tune the pre-trained model on document-level instances.

Pre-training on Pseudo Doc2Doc Instances. Due to the limited size of document-level parallel data, we make use of sentence-level parallel dataset $\left(\mathcal{SL}^{(\mathcal{S})}, \mathcal{SL}^{(\mathcal{Y})}\right)$. On the one hand, we translate source sentences $\mathcal{SL}^{(\mathcal{S})}$ by a sentence-level NMT trained on the dataset and get automatic translation $\mathcal{SL}^{(\mathcal{T})}$. On the other hand, we extract phrase translation table after doing word alignment (Dou and Neubig, 2021)[4] between sentence pairs in $\left(\mathcal{SL}^{(\mathcal{S})}, \mathcal{SL}^{(\mathcal{Y})}\right)$. Given a sentence-level triple $(S, T, Y) \in \left(\mathcal{SL}^{(\mathcal{S})}, \mathcal{SL}^{(\mathcal{T})}, \mathcal{SL}^{(\mathcal{Y})}\right)$, where

[4]https://github.com/neulab/awesome-align

S is the source-side sentence while T and Y are its automatic and reference translation, respectively. So (T, Y) is a sentence-level translation repair instance.

To construct lattice-like input, we need to *locate* inconsistency phases in T, and properly provide their candidate set. Given a source sentence $S = \{s_i\}|_{i=1}^{I}$ with I words, we simply view word s_i is an inconsistency trigger if it 1) is neither a stop word nor a high frequency word; and 2) has two or more translations in phrase translation table. Then for trigger s_i, we randomly select 1 (or 2 or 3) different translations from the phrase translation table and together with s_i's translation in T, and construct its translation candidate set. Finally, we shuffle all (T, Y) pairs and merge neighbouring pairs as a document-level DocRepair instance with max length of 512 on both input and output.

Fine-Tuning on Doc2Doc Instances. In the fine-tuning stage, we only use document-level parallel dataset $\left(\mathcal{DL}^{(\mathcal{S})}, \mathcal{DL}^{(\mathcal{Y})}\right)$. Given a document-level parallel pair $(\mathcal{S}, \mathcal{Y})$, we get its automatic translation $\mathcal{T}$ by above sentence-level NMT. Then for a document-level triple $(\mathcal{S}, \mathcal{T}, \mathcal{Y})$, we get a Doc2Doc training instance according to Section 2.

3.2 Reference-based Lexical Translation Consistency Metric

Lyu et al. (2021b) propose a metric to evaluate lexical translation consistency, named *lexical translation consistency ratio* (LTCR), which is based on whether translations of repeated words are consistent. However, it does not take the reference into account and ignores the correctness of these translations. Therefore, we extend LTCR and propose *ref-LTCR* by comparing the consistency between automatic and reference translations.

Given a document-level triple $(\mathcal{S}, \mathcal{T}, \mathcal{Y})$, let us assume that source word w appears k times in $\mathcal{S}$. Based on word alignment between $\mathcal{S}$ and $\mathcal{T}$, we could get its k automatic translations, i.e., $(t_1, \cdots, t_k)$, where t_i may consist of zero, one or more words. Similarly, we could get its k reference translations $(y_1, \cdots, y_k)$. For a pair of two automatic translations (t_i, t_j), the basic idea of ref-LTCR is that we encourage translation consistency between them only if their reference counterparts (y_i, y_j) are consistent. Specifically, we define the precision and recall values for word w as:

$$\begin{aligned} \text{Pre}(w) &= \frac{\sum_{i=1}^{k}\sum_{j=i+1}^{k}\mathbb{1}(t_i = t_j \;\&\&\; y_i = y_j)}{\sum_{i=1}^{k}\sum_{j=i+1}^{k}\mathbb{1}(t_i = t_j)}, \\ \text{Rec}(w) &= \frac{\sum_{i=1}^{k}\sum_{j=i+1}^{k}\mathbb{1}(t_i = t_j \;\&\&\; y_i = y_j)}{\sum_{i=1}^{k}\sum_{j=i+1}^{k}\mathbb{1}(y_i = y_j)}, \end{aligned} \tag{5}$$

where function $\mathbb{1}(condition)$ returns 1 if the condition is satisfied, otherwise 0; $t_i = t_j$ returns *true* if they are consistent, otherwise *false*.

In above it calculates ref-LTCR for a single word in a document. Likewise, we could apply the metric to all source words in a document-level parallel dataset by summing up all these words' corresponding numerators and denominators, respectively. After calculating the values of precision and recall, we report their F1 score, which is the harmonic mean of the two.

In brief, besides illustrating how frequent translation pairs of w is consistent within a document, ref-LTCR also measures how similar the consistency is compared against the reference translation. The higher ref-LTCR is, the more likely w is translated as in reference. See Appendix A for the computation of ref-LTCR when there exist multiple reference translations.

4 Experimentation

To verify the effectiveness of our proposed approach, we conduct experiments on three datasets with three language pairs, i.e., Chinese-to-English (ZH→EN), English-to-Chinese (EN→ZH) and German-to-English (DE→EN).

4.1 Experimental Setup

Datasets. For NIST (ZH↔EN), the pre-training data is from LDC and contains 2.0M sentence pairs. The document-level fine-tuning data is a subset of the pre-training set, including 66.4K documents with 0.83M sentence pairs. We use NIST 2006 as the development set and combine NIST 2002, 2003, 2004, 2005 and 2008 as the test set.

For PDC (ZH→EN), the document-level fine-tuning dataset is from Sun et al. (2022), which contains 10K documents with 1.39M sentence pairs. We combine the 1.39M sentence pairs and above NIST (ZH→EN) 2.0M sentence pairs as the pre-training data.

For Europarl (DE→EN), the document-level fine-tuning training set, and the development and test sets are from Maruf et al. (2019). We also use

Model	NIST (ZH→EN)				NIST (EN→ZH)			
	s-BLEU	d-BLEU	LTCR	ref-LTCR	s-BLEU	d-BLEU	LTCR	ref-LTCR
Sent-level NMT	48.45	50.70	65.25	78.61	25.82	27.24	64.59	67.87
SentRepair (Trans.)	48.49	50.76	64.89	78.03	25.71	27.12	64.39	67.69
DocRepair (Trans.)	-	51.12	-	-	-	27.01	-	-
DocRepair (G-Trans.)	49.25	51.54	65.39	78.11	26.31	27.76	64.66	67.92
DocRepair (Ours)	**50.28**	**52.28**	**69.51**	**80.74**	**26.66**	**28.11**	**67.11**	**70.37**

Table 1: Experimental results on the test sets of NIST ZH→EN and EN→ZH translations when repairing sentence-level NMT translation.

the sentence pairs from the fine-tuning training set as the pre-training data.

See Appendix B for detailed statistics and preprocessing of the experimental datasets.

Model Settings. For DocRepair models, we use G-Transformer (Bao et al., 2021) as the implementation of Transformer and extend it, which enlarges the translation unit to a whole document. See Appendix C for more details of the model settings.

Evaluation. To evaluate the overall repair performance, we report both sentence-level BLEU (s-BLEU) and document-level BLEU (d-BLEU) (Papineni et al., 2002). All BLEU scores calculated by the *multi-bleu.perl* script and are case-insensitive. To evaluate lexical translation consistency, we report both LTCR (Lyu et al., 2021b) and ref-LTCR.

Baselines. We compare our DocRepair approach against three baselines.

- SentRepair (Transformer): We train vanilla Transformer on sentence-level repair instances. All the instances are without word lattice-like input.
- DocRepair (Transformer): We pre-train vanilla Transformer on sentence-level translation repair instances of the same pre-training dataset and then fine tune it on document-level translation repair instances. All the instances are without word lattice-like input. Since we may not be able to recover sentence-level repair result from the output, we only report d-BLEU score for this baseline.
- DocRepair (G-Transformer): The pre-training and fine-tuning datasets are same as our approach except that this baseline does not use word lattice-like input.

4.2 Experimental Results

In inference, the trained DocRepair models can repair translation from both sentence-level NMT and document-level NMT. Here we again use G-Transformer as a representative of document-level NMT model. See Appendix D for more details about both the sentence-level and document-level NMT models.

4.2.1 Results of Repairing from Sentence-level NMT Translation

Results on NIST ZH↔EN Translation. Table 1 lists the performance on the test sets of the NIST ZH↔EN translation. From the table, we have the following observations.

- Baseline SentRepair (Transformer) has very limited effect on the four metrics. Baseline DocRepair (Transformer) improves performance in BLEU for ZH→EN translation while it slightly hurts performance for EN→ZH translation. Thanks to the group attention mechanism, DocRepair (G-Transformer) is a strong baseline which achieves significant improvement in BLEU for both ZH↔EN translations. Not surprisingly, DocRepair (G-Transformer) has very limited effect in terms of LTCR and ref-LTCR, indicating that it fails to improve lexical translation consistency.
- Our approach achieves best performance in terms of all metrics. With explicitly modeling inconsistency, it significantly improves LTCR and ref-LTCR, indicating that the repaired translation is improved in lexical translation consistency.

Results on PDC ZH→EN and Europarl DE→EN Translation. Table 2 shows the performance of PDC ZH→EN and Europarl DE→EN Translation. From the table, we observe a similar performance trend as on the NIST ZH↔EN translation. Overall, after repair our approach achieves 0.70 and 0.63 s-BLEU gains for PDC ZH→EN and Europarl DE→EN translation, respectively, while more importantly it obtains 1.82 and 0.64 ref-LTCR gains, respectively.

Model	PDC (ZH→EN)				Europarl (DE→EN)			
	s-BLEU	d-BLEU	LTCR	ref-LTCR	s-BLEU	d-BLEU	LTCR	ref-LTCR
Sent-level NMT	27.49	30.23	74.48	71.84	38.44	40.94	68.81	81.51
SentRepair (Trans.)	27.31	30.08	73.64	71.44	38.66	41.20	69.27	79.33
DocRepair (Trans.)	-	30.57	-	-	-	41.23	-	
DocRepair (G-Trans.)	27.94	30.82	72.68	70.45	38.79	41.30	69.57	81.71
DocRepair (Ours)	**28.19**	**31.05**	**77.51**	**73.66**	**39.07**	**41.56**	**74.02**	**82.15**

Table 2: Experimental results on the test sets of PDC ZH→EN and Europarl DE→EN translations when repairing sentence-level NMT translation.

Model	s-BLEU	d-BLEU	LTCR	ref-L.
NIST ZH→EN				
Doc-level NMT	48.77	**51.11**	65.89	78.06
DocRep. (Ours)	**48.86**	51.00	**69.75**	**80.52**
NIST EN→ZH				
Doc-level NMT	26.19	27.61	64.39	72.45
DocRep. (Ours)	**26.50**	**27.94**	**67.74**	**73.81**
PDC ZH→EN				
Doc-level NMT	28.48	31.33	74.73	72.53
DocRep. (Ours)	**28.68**	**31.54**	**79.92**	**74.30**
Europarl DE→EN				
Doc-level NMT	39.64	42.16	74.47	**82.80**
DocRep. (Ours)	**39.82**	**42.36**	**76.92**	82.71

Table 3: Experimental results on the test sets when repairing document-level NMT translation.

Ablation	s-BLEU	Δ	LTCR	Δ	ref-L.	Δ
Lattice-Input	**50.28**	-	**69.51**	-	**80.74**	-
w/o lat. pos.	49.04	-1.24	67.94	-1.57	79.13	-1.61
w/o tri. word	49.70	-0.58	68.68	-0.83	79.98	-0.76
w/o weights	49.84	-0.45	69.41	-0.10	80.40	-0.34

Table 4: Ablation study results.

Number	Count	%	Number	Count	%
2	567541	78.51	3	117061	16.19
4	28693	4.01	5	6945	0.96
>6	2355	0.33	All	722865	100

Table 5: Number of translation candidates.

We note that over the baseline of DocRepair (G-Transformer), the averaged improvement our approach achieved in s-BLEU/d-BLEU is 0.48/0.40, which is much less than the improvement of 3.96/2.18 in LTCR/ref-LTCR. This is because that BLEU is not sensitive to improvement in consistency in document-level translations. As shown in case study (Appendix F), though our approach improves translation readability and achieves consistent translations for the source words appearing multiple times, it has limited effect in BLEU.

4.2.2 Results of Repairing Document-level NMT Translation

Moving to translations of document-level NMT models, Table 3 compares the performance before and after repair for the four translation tasks. It shows that though document-level NMT achieves higher performance in s-BLEU/d-BLEU than sentence-level NMT, except on Europarl (DE→EN) it has very limited effect in terms of LTCR and ref-LTCR. Based on the improved translation, our approach further significantly improves lexical translation consistency while it slight improves performance in BLEU.

5 Analysis

Next, we take NIST ZH→EN translation as a representative to discuss how our proposed approach improves performance.

5.1 Ablation Study

We further conduct ablation study to investigate the contributions of the three components in our model: 1) token lattice position; 2) source-side trigger words; and 3) token weights. From Table 4, we first observe that token lattice position contributes most as it is essential to preserve lattice structure. Second, additionally including source-side trigger word is also helpful as the DocRepair model could translate them under the document-level context.

5.2 Statistics about Inconsistency

In the fine-tuning dataset, on average each document has 10.89 inconsistent phrases while each sentence has 0.87 ones. These inconsistency phrases account for 9.19% of all tokens in the translation.

For inconsistency phrases, the number of their translation candidates differ greatly. As shown in Table 5, about 98.71% of our interested words have 4 or less candidates. This is the reason that we randomly choose 2∼4 translation candidates for each inconsistency when pre-training models on pseudo Doc2Doc instances.

5.3 Effect of Different Pre-training Strategies

In the pre-training stage, we pre-train the model on pseudo document-level dataset which originates from a large sentence-level parallel dataset. Here,

Model	s-BLEU	LTCR	ref-L.
Sent-NMT	48.45	65.25	78.61
DocRepair (w/o pre-training)	47.94	69.19	79.74
DocRepair (w/ 0.83M)	49.06	69.24	79.93
DocRepair (w/ 2.0M)	**50.28**	**69.51**	**80.74**

Table 6: Experimental results with different pre-training strategies.

Annotator	Equal	Better	Worse
1	44%	36%	20%
2	49%	33%	18%
Average	46%	35%	19%

Table 7: Human evaluation results on 200 sentence groups from our test set.

we further investigate two other variants about pre-training: 1) we directly fine-tune the DocRepair model from scratch, i.e., without pre-training; and 2) we pre-train the model only on the sentence-level parallel dataset (i.e., 0.83M sentence pairs) from the document-level dataset used in fine-tuning. That is to say, the datasets for pre-training and fine-tuning are same, but with different training instances. From Table 6, we observe that pre-training on pseudo document-level dataset is helpful to improve repair performance in all metrics, especially BLEU. Moreover, the larger sentence-level dataset used in pre-training is, the higher repair performance is achieved. Finally, no matter how much sentence-level dataset is used in pre-training, explicitly modeling inconsistency can significantly improve translation consistency.

5.4 Human Evaluation

We randomly select 200 groups from the test set and conduct human evaluation on them. For each group, it contains four consecutive source-side sentences, and their two translations, i.e., the sentence-level NMT output and its repaired version by our DocRepair model. The two translations are presented with no indication which one is repaired. Following Voita et al. (2019) and Lyu et al. (2021b), the task is to choose one of three options: (1) the first translation is better, (2) the second translation is better, and (3) the translations are of equal quality. Two annotators are asked to avoid the third option if they are able to give preference to one of the translations.

Table 7 shows the results of human evaluation. On average the annotators mark 46% cases as having equal quality. Among the others, our approach outperforms Transformer in 65% cases, suggesting that overall the annotators have a strong preference for our repaired translation.

6 Related Work

The idea of "one translation per discourse" has been studied in both document-level translation and repair (i.e., post-editing).

Encouraging Lexical Translation Consistency in Translation. There exist many studies in MT that explicitly encourage lexical translation consistency. In statistical machine translation (SMT), for example, Gong et al. (2011) use cache to store recent translation and Türe et al. (2012) design a few consistency features to improve translation consistency in document-level translation. Moving to NMT, both Kang et al. (2021) and Lyu et al. (2021b) perform corpus study and observe that document-level translation of NMT suffers seriously from translation consistency. Lyu et al. (2021a) constrain repeated words in a document having similar hidden states, thus encourage their translations being consistent. Both Kang et al. (2021) and Lyu et al. (2022) construct lexical chains which consist of repeated words in a document. They use different approaches to learn (or model) each chain's translation.

Encouraging Lexical Translation Consistency in Post-Editing. In SMT, Carpuat (2009), Xiao et al. (2011) and Garcia et al. (2014, 2017) propose different post-editing approaches to re-translate those repeated source words which have been translated differently. Pu et al. (2017) aim to improve translation consistency for repeated nouns. They design a classifier to predict whether a pair of repeated nouns in a text should be translated by the same noun in target-language. Moving to NMT, to our best knowledge, this is the first work that explicitly focuses on document-level lexical translation consistency in post-editing. The most related work to ours is Voita et al. (2019), who propose a context-aware model that performs post-editing on four-sentence fragment of translations and correct the inconsistencies among individual translations in context. Different from them, we extend the local context from four sentences into a document. More importantly, our DocRepair model is inconsistency-aware with lattice-like input which consumes inconsistency translation.

7 Conclusion

In this paper, we have proposed an inconsistency-aware DocRepair approach to improve document-level translation consistency via automatic post-editing. We first locate inconsistency in text translation and provide translation candidates for each inconsistency. Then we use lattice-like input to properly model inconsistency and their candidates in a document-level repair model. Experimental results on three document-level translation datasets show that our approach not only achieves improvement on translation quality in BLEU, but also greatly improves lexical translation consistency.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive feedback. This work was supported by the National Natural Science Foundation of China (Grant No. 61876120).

Limitations

In this paper, we locate inconsistency in automatic translation by looking for inconsistent translations of source-side repeated words. Sometimes such inconsistency is allowed and even encouraged to increase diversity. Without explicitly estimating whether a repeated word needs to be translated consistently, our approach will hinder translation diversity. Modeling confidence score of a repeated word being translate consistently will be explored in future work.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

Guangsheng Bao, Yue Zhang, Zhiyang Teng, Boxing Chen, and Weihua Luo. 2021. G-transformer for document-level machine 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 3442–3455, Online. Association for Computational Linguistics.

Marine Carpuat. 2009. One translation per discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 19–27.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of EACL*, pages 2112–2128.

Eva Martínez Garcia, Carles Creus, Cristina Espana-Bonet, and Lluís Màrquez. 2017. Using word embeddings to enforce document-level lexical consistency in machine translation. *Prague Bulletin of Mathematical Linguistics*, 108:85–96.

Eva Martínez Garcia, Cristina Espana-Bonet, and Lluís Màrquez. 2014. Document-level machine translation as a re-translation process. *Procesamiento del Lenguaje Natural*, 53:103–110.

Zhengxian Gong, Min Zhang, and Guodong Zhou. 2011. Cache-based document-level statistical machine translation. In *Proceedings of EMNLP*, pages 909–919.

Liane Guillou. 2013. Analysing lexical consistency in translation. In *Proceedings of DiscoMT*, pages 10–18.

Xiaomian Kang, Yang Zhao, Jiajun Zhang, and Chengqing Zong. 2021. Enhancing lexical translation consistency for document-level neural machine translation. *Transactions on Asian and Low-Resource Language Information Processing*, 21:59:1–59:21.

Eissa Al Khotaba and Khaled Al Tarawneh. 2015. Lexical discourse analysis in translation. *Education and Practice*, 6(3):106–112.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Yuxuan Lai, Yijia Liu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2021. Lattice-BERT: Leveraging multi-granularity representations in Chinese pre-trained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1716–1731, Online. Association for Computational Linguistics.

Chenyang Lyu, Lifeng Shang, Yvette Graham, Jennifer Foster, Xin Jiang, and Qun Liu. 2021a. Improving unsupervised question answering via summarization-informed question generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4134–4148, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xinglin Lyu, Junhui Li, Zhengxian Gong, and Min Zhang. 2021b. Encouraging lexical translation consistency for document-level neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3265–3277, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xinglin Lyu, Junhui Li, Shimin Tao, Hao Yang, Ying Qin, and Min Zhang. 2022. Modeling consistency preference via lexical chains for document-level neural machine translation. In *Proceedings of EMNLP*, pages 6312–6326.

Sameen Maruf, André F. T. Martins, and Gholamreza Haffari. 2019. Selective attention for context-aware neural machine translation. 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 3092–3102, Minneapolis, Minnesota. Association for Computational Linguistics.

Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. 2022. A survey on document-level neural machine translation: Methods and evaluation. *ACM Computing Surveys*, 54:45:1–45:36.

Magnus Merkel. 1996. Consistency and variation in technical translation: a study of translators ' attitudes. In *Proceedings of Unity in Diversity, Translation Studies Conference*, pages 137–149.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Xiao Pu, Laura Mascarell, and Andrei Popescu-Belis. 2017. Consistent translation of repeated nouns using syntactic and semantic cues. In *Proceedings of EACL*, pages 948–957.

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

Zewei Sun, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Shujian Huang, Jiajun Chen, and Lei Li. 2022. Rethinking document-level neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3537–3548, Dublin, Ireland. Association for Computational Linguistics.

Ferhan Türe, Douglas W Oard, and Philip Resnik. 2012. Encouraging consistent translation choices. In *Proceedings of NAACL*, pages 417–426.

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 *Proceedings of NIPS*, pages 5998–6008.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. Context-aware monolingual repair for neural machine translation. In *Proceedings of EMNLP-IJCNLP*, pages 877–886.

Tong Xiao, Jingbo Zhu, Shujie Yao, and Hao Zhang. 2011. Document-level consistency verification in machine translation. In *Proceedings of Machine Translation Summit XIII: Papers*, Xiamen, China.

A *ref-LTCR* for Multiple Reference Translations

In Section 3.2, we present the *ref-LTCR* calculation method for a single reference. When it comes to multiple references, we need to modify the Eq. 5.

Suppose that there are M references for a document S. For a source word w which appears k times in S, we could get its k reference translations $(y_1^1, \cdots, y_1^k), \cdots, (y_M^1, \cdots, y_M^k)$ for M references respectively. Then we define $C(i,j)$ as:

$$C(i,j) = \mathbb{1}(y_1^i = y_1^j || \cdots || y_M^i = y_M^j), \tag{6}$$

which $C(i,j)$ denotes whether the reference translations in index i and index j should be consistent. So we can update Eq. 5 as:

$$\begin{aligned} \text{Pre}(w) &= \frac{\sum_{i=1}^{k}\sum_{j=i+1}^{k}\mathbb{1}(t_i = t_j \wedge C(i,j))}{\sum_{i=1}^{k}\sum_{j=i+1}^{k}\mathbb{1}(t_i = t_j)}, \\ \text{Rec}(w) &= \frac{\sum_{i=1}^{k}\sum_{j=i+1}^{k}\mathbb{1}(t_i = t_j \wedge C(i,j))}{\sum_{i=1}^{k}\sum_{j=i+1}^{k}C(i,j)}. \end{aligned} \tag{7}$$

When one of the reference translations for a pair of (y^i, y^j) is consistent, we can assume that it should be consistent when translating.

B Experimental Datasets and Preprocessing

For ZH↔EN (NIST), the sentence-level training set consists of LDC2002E18, LDC2003E07, LDC2003E14, news part of LDC2004T08 and the document-level training set consists of LDC2002T01, LDC2004T07, LDC2005T06, LDC2005T10, LDC2009T02, LDC2009T15, LDC2010T03. The pre-training data contains both above sentence-level and document-level sets while only the document-level sets are used for document-level fine-tuning. In the development and test sets every Chinese document has four aligned English documents, thus for

Set	NIST		PDC		Europarl	
	#Doc	#Sent	#Doc	#Sent	#Doc	#Sent
Pre-Training	-	2M	-	3.39M	-	1.67M
Fine-Tuning	66,396	0.83M	59,384	1.39M	117,855	1.67M
Dev	100	1664	100	2320	240	3587
Test	580	5833	148	4858	360	5134

Table 8: Dataset Statistics of the number of Documents and Sentences.

ZH→EN translation one Chinese sentence has four references. In turn for EN→ZH translation each English sentence has one reference, and the numbers of sentences in development and test sets are four times those of ZH→EN translation, e.g., 4×1664 and 4×5833, respectively.

Detailed statistics for all the datasets is in Table 8. Note that the pre-training dataset shown in the table is sentence-level and we need to shuffle and merge into pseudo document-level dataset as described in Section 3.1. The number of documents shown in Table 8 is the number of complete documents. In all experiments, we split them into sub-documents with the max length of 512 on both input and output. For d-BLEU, we restore the output translations to complete documents and calculate the BLEU score.

For all tasks, the English and German sentences are tokenized and lowercased by Moses toolkits (Koehn et al., 2007)[5] while the Chinese sentences are segmented by Jieba.[6] In all experiments, we segment words into subwords with 32K merge operations (Sennrich et al., 2016).

C Model Setting and Training

Following the standard Transformer base model (Vaswani et al., 2017), we use 6 layers for both encoders and decoders, 512 dimensions for model, 2048 dimensions for ffn layers, 8 heads for attention. The parameter settings in G-Transformer are same as Bao et al. (2021). In the pre-training stage, we only use the group attention to make model focus on the current sentence and exclude all tokens outside the sentence. In the fine-tuning stage, we use the combined attention to help model focus on both target sentence and contextual information. We train the models on 4 V100 GPUs with batch-size 8192 and use Adam with $\beta 1$ = 0.9, $\beta 2$ = 0.98 for optimization (Kingma and Ba, 2015). We set dropout as 0.3 for all experiments and run our models once with a fixed seed. In both pre-training and fine-tuning stage, we use early-stopping strategy with the patience as 10 and choose the best checkpoint according to the valid loss. The whole training process takes approximately 40 hours. In inference, we set the beam size to 5.

D Details of Sentence-Level and Document-level NMT Models

For the sentence-level NMT model, we use G-transformer (Bao et al., 2021) as the implementation of the Transformer-base with full mode to generate sentence-level translations.[7] The training datasets for the sentence-level NMT models are same as the pre-training datasets in Table 8.

For the document-level NMT model, we also use G-transformer with partial mode to generate document-level translations. We fine-tune document-level NMT on sentence-level Transformer described above using a document-level dataset, same as the fine-tuning datasets in Table 8.

For both sentence-level and document-level NMT models, we use the same parameter settings as in G-Transformer (Bao et al., 2021) with dropout as 0.3.

E Model Parameter

Table 9 shows the number of parameters used in our systems. Except the system without trigger words, the parameters of other systems are exactly the same. Adding trigger words increases the size of parameter since it introduces source-side vocabulary. It is also feasible not to include trigger words (i.e., w/o tri. word) in practice with a slight performance drop.

F Case Study

To better illustrate how our model improves lexical consistency, we provide an example from NIST 2004 test set. As shown in Figure 3, we observe

[5]https://github.com/moses-smt/mosesdecoder
[6]https://github.com/fxsjy/jieba
[7]https://github.com/baoguangsheng/g-transformer

Source	<#1> … 瑞典 大使 毁损 艺术品 引发 … <#2> … 大使 马兹尔 破坏 而 引发 的 争议 … <#3> 马兹尔 大使 … 自杀 炸弹 客 照片 的 艺术品 时 <#8> … 今天 出面 挺 马兹尔 … <#9> … 微笑 的 照片 插 在 船头 ， 作为 船帆
Sentence-Level NMT	<#1> ... triggered by destruction of art by ... <#2> ... was undermined by israeli ambassador to sweden marzir ... <#3> ... ambassador marshall saw an artwork showing pictures of a suicide bomber ... <#8> ... came out today to support mahathir ... <#9> ... with a smiling photo of ... hanging on the top of the ship ... BLEU: 43.12
DocRepair (G-Trans)	<#1> … by israeli ambassador to sweden 's destruction of art <#2> … was destroyed by israeli ambassador to sweden marzir ... <#3> ... ambassador marshall saw an artwork showing pictures of a suicide bomber ... <#8> ... came out today to support mahathir ... <#9> ... with a smiling photo of ... hanging on the top ... BLEU: 44.07
Our Approach	<#1> … triggered by destruction of artwork by … <#2> … was damaged by israeli ambassador to sweden marzir ... <#3> ... ambassador marzir saw an artwork showing pictures of a suicide bomber ... <#8> ... came out today to support marzir ... <#9> ... with a smiling picture of ... hanging on the top ... BLEU: 44.30
Reference	<#1> israel 's ambassador to sweden vandalizes artwork ... <#2> ... museum of national history by mazel , israeli ambassador to sweden … <#3> ambassador mazel visited … artwork featuring a photo of the suicide bomber … <#8> ... expressed his support for mazel today ... <#9> ... with a photo of a smiling hanadi jaradat placed on the ...

Figure 3: An example of document-level Chinese-to-English translation from our test set.

Model	s-BLEU	#Params (M)
Lattice-Input	**50.28**	74.77
w/o lat. pos.	49.04	74.77
w/o tri. word	49.70	**70.34**
w/o weights	49.84	74.77

Table 9: Parameter (in millions) comparison of our different DocRepair systems.

that in this example, the sentence-level NMT model translates source-side repeated words into different translations. For example, person name 马兹尔/*ma_zi_er* maps into three different translations, i.e., *marzir*, *marshall* and *mahathir* while DocRepair (G-Transformer) could not fix such inconsistency. By contrast, our approach consistently repairs the translation of 马兹尔/*ma_zi_er* into *marzir*. Compared to the reference translation *mazel*, thought not correct, the translation *marzir* would not confuse readers. This explains that BLEU is not sensitive to improvement in translation consistency.

ACL 2023 Responsible NLP Checklist

A For every submission:

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

☐ 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?
Section 1

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

B ☑ Did you use or create scientific artifacts?

Section 3.1, Section 4.1, Appendix D

☑ B1. Did you cite the creators of artifacts you used?
Section 3.1, Section 4.1, Appendix D

☐ 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.?
Section 4.1, Appendix B

☑ 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.
Appendix B

C ☑ Did you run computational experiments?

Section 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?
Appendix C, Appendix E

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?
Section 4.1, Appendix C, Appendix D

☑ 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?
Appendix C

☑ 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.)?
Appendix B

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

Section 5.4

☐ 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.?
Not applicable. Left blank.

☐ 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)?
Not applicable. Left blank.

☐ 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?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.