Breaking Corpus Bottleneck for Context-Aware Neural Machine Translation with Cross-Task Pre-training

[1]Linqing Chen [1]Junhui Li[*] [1]Zhengxian Gong [2]Boxing Chen
[2]Weihua Luo [1]Min Zhang [1]Guodong Zhou
[1]School of Computer Science and Technology, Soochow University, Suzhou, China
[2]Alibaba DAMO Academy
{lijunhui, zhxgong, minzhang, gdzhou}@suda.edu.cn,
lqchen21@gmail.com, {boxing.cbx, weihua.luowh}@alibaba-inc.com

Abstract

Context-aware neural machine translation (NMT) remains challenging due to the lack of large-scale document-level parallel dataset. To break the corpus bottleneck, in this paper we aim to improve context-aware NMT by taking the advantage of the availability of both large-scale sentence-level parallel dataset and source-side monolingual documents.[1] To this end, we propose two pre-training tasks. One learns to translate a sentence from source language to target language on the sentence-level parallel dataset while the other learns to *translate* a document from deliberately noised to original on the monolingual documents. Importantly, the two pre-training tasks are jointly and simultaneously learned via the same model, thereafter fine-tuned on scale-limited parallel documents from both sentence-level and document-level perspectives. Experimental results on four translation tasks show that our approach significantly improves translation performance. One nice property of our approach is that the fine-tuned model can be used to translate both sentences and documents.

1 Introduction

Document-level context-aware neural machine translation (NMT) aims to translate sentences in a document under the guidance of document-level context. Recent years have witnessed great improvement in context-aware NMT with extensive attempts at effectively leveraging document-level context ((Tiedemann and Scherrer, 2017; Maruf and Haffari, 2018; Maruf et al., 2019), to name a few). However, the performance of context-aware NMT still suffers from the size of parallel document dataset. On the one hand, unlike sentence-level translation models which could be well trained on large-scale sentence-level parallel datasets, the translation models of context-aware NMT may result in insufficient training. On the other hand, with only scale-limited source-side documents, the context encoders may fail to effectively extract useful context from the whole document.[2] On the contrary, large-scale of parallel sentence corpora, and especially monolingual document corpora are much easier to find. In this paper, our goal is to break the corpus bottleneck for context-aware NMT by leveraging both large-scale sentence-level parallel dataset and monolingual documents. Specifically, we aim to use the former to boost the performance of translation models while employ the latter to enhance the context encoders' capability of capturing useful context information.

There have been several attempts to boost context-aware NMT performance in the scenarios where the document-level parallel dataset is scale-limited, or even not available. On the one hand, sentence-level parallel dataset is a natural resource to use. For example, Zhang et al. (2018) propose a two-stage training strategy for context-aware NMT by pre-training the model on a sentence-level parallel dataset. On the other hand, Junczys-Dowmunt (2019) leverage large-scale source-side monolingual documents, in which they simply concatenate sentences within a document into a long sequence and explore multi-task training via the BERT-objective (Devlin et al., 2019) on the encoder. Due to that different models are usually required to model sentences and documents, however, it is challenging to effectively take them both in a single model.

In order to effectively and simultaneously model

[*] Corresponding Author: Junhui Li.

[1] If not specified, monolingual documents are all for source-side through this paper.

[2] We note that not all, but many context-aware NMT models contain a context encoder to extract global context information from the document.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics
and the 11th International Joint Conference on Natural Language Processing, pages 2851–2861
August 1–6, 2021. ©2021 Association for Computational Linguistics

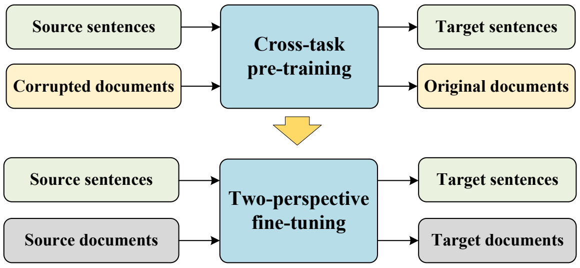

Figure 1: Illustration of the proposed cross-task pre-training (upper) and fine-tuning with two perspectives (below).

both sentence-level parallel dataset and monolingual documents, in this paper we propose a novel cross-task pre-training approach. As shown in Figure 1, we define two pre-training tasks. One learns to translate a sentence from source language to target language while the other learns to *translate* a document from deliberately noised to original. Importantly, the two pre-training tasks are jointly learned via the same model synchronously. Then we use document-level parallel dataset to fine-tune the properly pre-trained models. Similarly to the pre-training, we can fine-tune the models from both sentence-level and document-level perspectives. Experimental results on four document-level translation tasks show that our approach significantly improves translation performance, suggesting the effectiveness of our approach in modeling both sentence-level parallel dataset and monolingual documents. One nice property of our approach is that the fine-tuned models can be used to translate both sentences and documents.

2 Cross-Task Pre-training

In the following, we first describe our pre-training tasks defined upon sentence-level parallel dataset and large-scale monolingual documents (Section 2.1). Then we detail our model which caters such pre-training tasks (Section 2.2). Finally, we present our joint pre-training (Section 2.3).

2.1 Pre-training Tasks

We define two pre-training tasks in our pre-training. One is on sentence-level parallel dataset while the other is on monolingual documents.

Sentence-level Translation Given large-scale sentence-level parallel dataset, our pre-training task is quite straight, i.e., sentence-level translation.

Document-level Restoration Given monolingual documents, our pre-training task is to restore a document from a noised version. To this end, we deliberately corrupt documents by following the two pre-training objectives, which are inspired by both gap sentence objective (Zhang et al., 2020) and masked language model objective (Devlin et al., 2019).

- Context-Aware Gap Sentence Restoration (CA-GSR). Given a document S with N sentences, we randomly select M sentences as gap sentences and replace them with a mask token [MASK1] to inform the model. The gap sentence ratio is, therefore M/N. For each selected gap sentence, we use its left and right neighbours as input while the gap sentence serves as output. To mimic document-level translation task, in the selection the first and the last sentences are always not selected while any two consequent sentences are not both selected.

- Context-Aware Masked Sentence Restoration (CA-MSR). Given a sentence X, we follow BERT and randomly select 15% tokens in it. The selected tokens are (1) 80% of time replaced by a mask token [MASK2], or (2) 10% of time replaced by a random token, or (3) 10% of time unchanged. For a sentence, we use its masked $\widehat{X}$ as input while the original X serves as output.

Both CA-GSR and CA-MSR are applied simultaneously with the noised document as context. For convenience of presentation, we use a concrete example to illustrate the input and output of our document-level restoration task. As shown in Figure 2, let assume that a document $\mathcal{X}$ contains 6 sentences and the third and fifth sentences (i.e., X_3 and X_5) are selected as gap sentences while the others are not. On the one hand, for a sentence which is not selected as gap sentence, e.g., X_1, we use its masked version (e.g., $\widehat{X_1}$) as input while try to predict its original sentence (e.g., X_1). On the other hand, for a gap sentence, e.g., X_3, we concatenate its left and right neighbouring sentences with separator [MASK1] and try to predict the gap sentence (e.g., X_3). As shown in Figure 2, sentences from S_1 to S_6 constitute document-level input $\mathcal{S}$ while sentences from T_1 to T_6 make up output $\mathcal{T}$. Note that we do not include either gap sentences themselves or their masked version in $\mathcal{S}$, in case the document

$\mathcal{S}$	$\mathcal{T}$
S_1: $\widehat{X_1}$	T_1: X_1
S_2: $\widehat{X_2}$	T_2: X_2
S_3: X_2 [MASK1] X_4	T_3: X_3
S_4: $\widehat{X_4}$	T_4: X_4
S_5: X_4 [MASK1] X_6	T_5: X_5
S_6: $\widehat{X_6}$	T_6: X_6
(a) Document-Level input $\mathcal{S}$;	(b) Document-Level output $\mathcal{T}$;

Figure 2: Illustration of the proposed document-level restoration task.

context contains obvious hints for generating gap sentences.

Overall, the pre-training task of document-level restoration is to predict target output $\mathcal{T}$ by giving source input $\mathcal{S}$, which is the same as the task of document-level translation, except that in the restoration $\mathcal{S}$ and $\mathcal{T}$ are in the same language while in the latter the two are in different languages.

2.2 Joint Modeling of Pre-training Tasks

We use the same model to cater the above two pre-training tasks. Since the task of document-level restoration is more complicated than the task of sentence-level translation, we first describe the model for document-level restoration (Section 2.2.1). Then we apply the model for sentence-level translation (Section 2.2.2).

2.2.1 Context-Aware Modeling for Document-Level Restoration

We define some notations before describing our model. Given a document-level source input $\mathcal{S} = (S_1, \cdots, S_N)$ and target output $\mathcal{T} = (T_1, \cdots, T_N)$ with N sentence pairs, we assume each source sentence $S_i = (s_{i,1}, \cdots, s_{i,n})$ consists of n words. We use d_m as the size of embedding and hidden state throughout the entire model.

Figure 3 shows our context-aware model. It contains two parts, namely a global context encoder and a seq2seq model augmented by context representation. Note that for document-level restoration, we take documents as input units.

Global Context Encoder For the i-th input sentence S_i in document $\mathcal{S}$, the global context encoder aims to extract useful global context for every word $s_{i,j}$ in it. As shown in Figure 3(a), the encoder consists of a stack of N_g identical encoder layers. Each encoder layer consists of four major sub-layers: a self-attention sub-layer, a sentence representation sub-layer, a global context attention sub-layer and a feed-forward sub-layer.

In the k-th encoder layer, the **self-attention sub-layer** takes $A_i^{(k)} \in \mathbb{R}^{n \times d_m}$ as input and computes a new sequence $B_i^{(k)}$ with the same length via multi-head attention function:

$$B_i^{(k)} = \text{MultiHead}\left(q = A_i^{(k)}, k = A_i^{(k)}, v = A_i^{(k)}\right), \quad (1)$$

where the output $B_i^{(k)}$ is in the shape of $\mathbb{R}^{n \times d_m}$,[3] and q, k, v represent the query and key-value pairs in attention mechanism respectively. For the first encoder layer, $A_i^{(1)}$ is the addition of S_i's word embedding and its position embedding while for other layers, $A_i^{(k)}$ is the output of the proceeding encoder layer.

In the k-th encoder layer, the **sentence representation sub-layer** takes $B_i^{(k)}$ as input and computes a vector to represent the sentence through a linear combination with a vector of weights as:

$$\alpha_i^{(k)} = \text{softmax}\left(W^2 \tanh\left(W^1 \left(B_i^{(k)}\right)^T\right)\right) \quad (2)$$

where $W^1 \in \mathbb{R}^{d_m \times d_m}$ and $W^2 \in \mathbb{R}^{d_m}$ are model parameters. The output $\alpha_i^{(k)}$ is a n-sized vector. Then the representation vector of sentence S_i is the weighted sum of its hidden states:

$$C_i^{(k)} = \alpha_i^{(k)} B_i^{(k)}, \quad (3)$$

where $C_i^{(k)}$ is a d_m-sized vector. We then stack vectors of all sentences in $\mathcal{S}$ into $\mathcal{C}^{(k)}$, i.e., $\mathcal{C}^{(k)} = \left[C_1^{(k)}, \cdots, C_N^{(k)}\right]$. Note that $\mathcal{C}^{(k)} \in \mathbb{R}^{N \times d_m}$ is at document-level and represents the global context.

In the k-th encoder layer, the **global context attention sub-layer** extracts useful global context for $s_{i,j}$ in S_i. This is also done via multi-head attention function:

$$D_i^{(k)} = \text{MultiHead}\left(q = B_i^{(k)}, k = \mathcal{C}^{(k)}, v = \mathcal{C}^{(k)}\right), \quad (4)$$

where the output $D_i^{(k)}$ is in the shape of $\mathbb{R}^{n \times d_m}$.

In the k-th encoder layer, the **Feed forward sub-layer** is applied to each position separately and

[3] The actual output of this sub-layer is LayerNorm$(B_i^{(k)} + A_i^{(k)})$, where LayerNorm is the layer normalization function. For simplicity, we do not include the residual addition and layer normalization functions in our sub-layers. Note that the sentence representation sub-layer is the only exception which does not have residual addition and layer normalization.

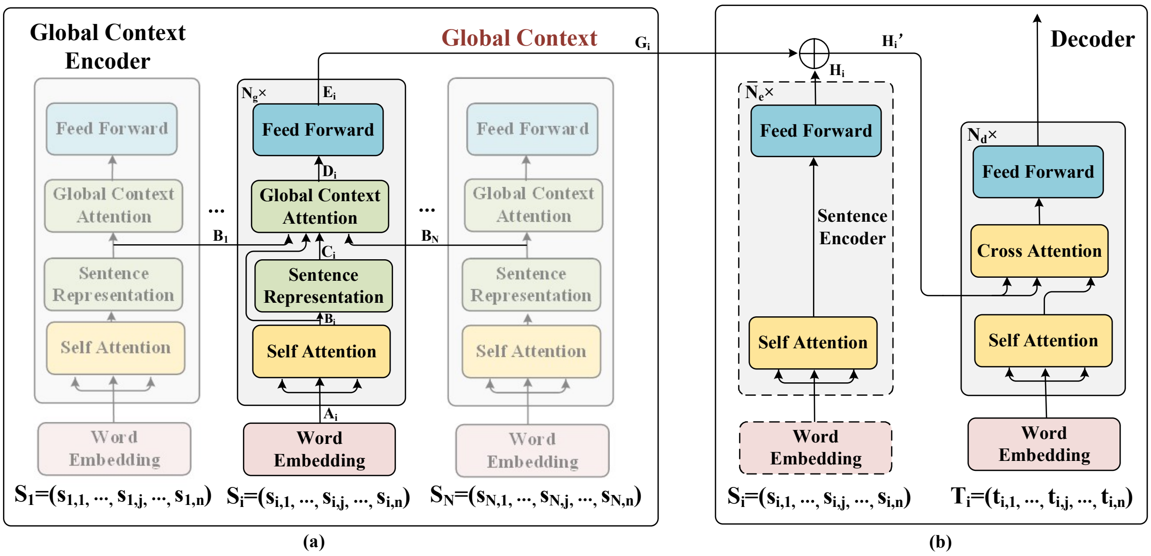

Figure 3: Illustration of the proposed context-aware model. Note that 1) we share the two sub-layers of self-attention and feed forward between the global context encoder and the sentence encoder; 2) the model uses the same vocabulary for the tasks in pre-training and fine-tuning since we share vocabulary for the source and target languages; 3) we use (b) for sentence-level translation and turn off the gate mechanism.

identically by two linear transformations with a ReLU activation in between.

$$E_i^{(k)} = \max\left(0, D_i^{(k)} W^{F1} + b^{F1}\right) W^{F2} + b^{F2}, \quad (5)$$

where $W^{F1}, W^{F2} \in \mathbb{R}^{d_m \times d_m}$, and $b^{F1}, b^{F2} \in \mathbb{R}^{d_m}$ are model parameters.

We denote $G_i \in \mathbb{R}^{n \times d_m}$ as the final output of the global context encoder, i.e., $G_i = E_i^{(N_g)}$. That is to say, G_i represents the **context representation** for sentence S_i.

Context-Aware Model As shown in Figure 3 (b), the seq2seq model is very similar to the standard Transformer, except that it is now equipped with context representation obtained by the global context encoder. For sentence S_i, we denote the sentence encoder output as $H_i \in \mathbb{R}^{n \times d_m}$. To leverage its context representation G_i, we define a gate to linearly combine the two kinds of representation via:

$$H_i' = \lambda H_i + (1 - \lambda) G_i, \quad (6)$$

where the gating weight is computed by

$$\lambda = \text{sigmoid}\left([H_i; G_i] W^G\right), \quad (7)$$

where $W^G \in \mathbb{R}^{2d_m \times d_m}$ are model parameters.

Then we use H_i' to replace H_i as the input to the decoder. We point out that in the global context encoder and sentence encoder, we share the self-attention sub-layer and the feed forward sub-layer. That is to say, compared to the standard Transformer, we introduce new parameters to cater the sentence representation sub-layers, the global context sub-layers, and the gate mechanism to combine the two kinds of representation in Eq. 6.

2.2.2 Adapting Context-Aware Model to Sentence-Level Translation

In the first pre-training task, sentence-level translation is context-agnostic and does not require the global context encoder. Therefore, it only uses the sentence encoder and decoder, as shown in Figure 3 (b). Moreover, we turn off the gate mechanism by setting $H_i' = H_i$. Since we share the two sub-layers of self-attention and feed forward between the sentence encoder and the global context encoder, updating the model by sentence-level translation will have direct impact on the global context encoder too.

2.3 Joint Pre-training Process

As shown in our experimentation, we share the same vocabulary for pre-training tasks. To train the above two pre-training tasks with a single model, we follow the strategy used in Johnson et al. (2017) and add a preceding language tag to each source and target sentence.

Our joint pre-training on two tasks falls into the paradigm of multi-task learning (MTL). In training stage, we take turns to load the training data of these pre-training tasks. For example, we update model parameters on a batch of training instances from the first task, and then update parameters on a batch of training instances of the other, and the process repeats.

3 Fine-tuning on Document-Level Parallel Dataset

3.1 Fine-tuning Tasks

Similar to pre-training tasks, we define the following two different fine-tuning tasks from both sentence-level and document-level.

Sentence-level Translation We first extract sentence-level parallel sentence pairs from the document-level parallel dataset for fine-tuning. This fine-tuning task enables the fine-tuned model to translate sentences. In fine-tuning, this task is processed as same as the sentence-level translation task in pre-training.

Document-level Translation Given a parallel document $(\mathcal{X}, \mathcal{Y})$ with N sentence pairs $(X_i, Y_i)\,|_1^N$. This fine-tune task is to translate source document $\mathcal{X}$ into target document $\mathcal{Y}$. In fine-tuning, this task takes parallel documents as input units and is processed as same as the document-level restoration task in pre-training.

3.2 Fine-tuning Process

The fine-tuning process is quite similar as the pre-training process in Section 2.3. Specifically, we add a preceding language tag to each sentence. Meanwhile in fine-tuning, we alternatively load batches of the two fine-tuning tasks.

4 Experimentation

To test the effect of our approach in leveraging sentence-level parallel dataset and monolingual documents, we carry out experiments on Chinese-to-English (ZH-EN) and English-to-German (EN-DE) translation.

4.1 Experimental Settings

Pre-training data settings. The ZH-EN sentence-level parallel dataset contains 2.0M sentence pairs with 54.8M Chinese words and 60.8M English words.[4] We use WMT14 EN-DE translation dataset as the EN-DE sentence-level parallel dataset which consists of 4.4M sentence pairs.[5]

We use Chinese Gigaword (LDC2009T27) and English Gigaword (LDC2012T21) as monolingual document dataset for ZH-EN and En-DE translation, respectively. For efficient training, we split long documents into sub-documents with at most 30 sentences. We have 2.6M (7.3M) sub-documents with 24M (102M) sentences in total for Chinese (English). Upon the monolingual documents, we prepare training instances for the document-level restoration task and set gap sentence ratio to 20%.

All Chinese sentences are segmented by Jieba[6] while all English and German sentences are tokenized by Moses scripts (Koehn et al., 2007).[7] For ZH-EN (EN-DE) translation, we merge the source and target sentences of the parallel dataset and the monolingual document and segment words into sub-words by a BPE model with 30K (25K) operations (Sennrich et al., 2016).

Fine-tuning data settings. For ZH-EN, we have one translation task on news domain. The document-level parallel corpus of training set include 41K documents with 780K sentence pairs.[8] We use the NIST MT 2006 dataset as the development set, and combine the NIST MT 2002, 2003, 2004, 2005, 2008 datasets as test set..

For EN-DE, we test three translation tasks in domains of TED talks, News-Commentary and Europarl.

- TED, which is from IWSLT 2017 MT track (Cettolo et al., 2012). We combine test2016 and test2017 as our test set while the rest as the development set.

- News, which is from News Commentary v11 corpus.[9] We use news-test2015 and news-test2016 as the development set and test set, respectively.

[4] It consists of LDC2002E18, LDC2003E07, LDC2003E14, news part of LDC2004T08, LDC2002T01, LDC2004T07, LDC2005T06, LDC2005T10, LDC2009T02, LDC2009T15, LDC2010T03.

[5] https://www.statmt.org/wmt14/translation-task.html

[6] https://github.com/messense/jieba-rs

[7] As related studies, we lowercase English sentences in ZH-EN while truecase English and German sentences in EN-DE.

[8] It consists of LDC2002T01, LDC2004T07, LDC2005T06, LDC2005T10, LDC2009T02, LDC2009T15, LDC2010T03. Note that they are also included in ZH-EN parallel dataset.

[9] http://www.casmacat.eu/corpus/news-commentary.html

#	Model	Bi-sent	Mo-doc	ZH-EN		EN-DE (TED)		EN-DE (News)		EN-DE (Europarl)		Avg.	
				BLEU	Meteor	BLEU	Meteor	BLEU	Meteor	BLEU	Meteor	BLEU	Meteor
DocT (Zhang et al., 2018)		✗	✗	40.32	27.93	24.00	44.69	23.08	42.40	29.32	46.72	29.18	40.43
HAN (Miculicich et al., 2018)		✗	✗	40.83	28.19	24.58	45.48	25.03	44.02	28.60	46.09	29.76	40.94
SAN (Maruf et al., 2019)		✗	✗	41.01	28.37	24.42	45.26	24.84	44.17	29.75	47.22	30.00	41.26
QCN (Yang et al., 2019)		✗	✗	-	-	25.19	46.09	22.37	41.88	29.82	47.86	-	-
MCN (Zheng et al., 2020)		✗	✗	40.92	28.25	25.10	-	24.91	-	30.40	-	30.33	-
#1	Transformer	✗	✗	39.64	27.56	23.02	43.66	22.03	41.37	28.65	45.83	28.33	39.61
#2	Ours-sent	✗	✗	40.73	27.97	24.75	45.83	24.19	43.96	29.10	47.55	29.69	41.33
#3	Ours-doc	✗	✗	41.27	28.46	25.31	46.30	24.70	44.38	30.07	47.93	30.34	41.76
#4	Transformer	✓	✓	46.30	32.91	26.94	47.06	26.80	46.99	29.90	47.50	32.48	43.62
#5	Ours-sent	✓	✓	49.58	35.97	28.73	48.80	28.41	48.52	30.61	48.29	34.33	45.40
#6	Ours-doc	✓	✓	50.03	36.50	29.31	49.40	29.01	48.83	31.52	49.02	34.97	45.94

Table 1: Performance (BLEU and Meteor scores) on test sets. Bi-sent/Mo-doc indicates if the models are pre-trained on sentence-level parallel dataset or monolingual documents (✗ for no and ✓ for yes). Ours-sent/Ours-doc indicates that we use sentences or documents as input units, i.e., performing sentence-level NMT or context-aware NMT. Scores are obtained by running their source code with our model settings.

- Europarl, which is extracted from the Europarl v7. The training, development and test sets are obtained through randomly splitting the corpus.

All above EN-DE document-level parallel datasets are downloaded from Maruf et al. (2019).[10] Similar to fine-tuning datasets, the pre-processing steps consist of word segmentation, tokenization, long document split. Then we segment the words into subwords using the BPE models trained on pre-training datasets. See Appendix A for more statistics of the fine-tuning datasets.

Model settings. We use OpenNMT (Klein et al., 2017) as the implementation of Transformer and implement our models based on it.[11] For all translation models, the numbers of layers in the context encoder, sentence encoder and decoder (i.e., N_g, N_e, and N_d in Fig 3) are set to 6. The hidden size and the filter size are set to 512 and 2048, respectively. The number of heads in multi-head attention is 8 and the dropout rate is 0.1. In pre-training, we train the models for 500K steps on four V100 GPUs with batch-size 8192. We use Adam (Kingma and Ba, 2015) with $\beta 1$ = 0.9, $\beta 2$ = 0.98 for optimization, and learning rate as 1, the warm-up step as 16K. In fine-tuning, we fine-tune the models for 200K steps on a single V100 GPU with batch-size 8192, learning rate 0.3, and warm-up step 4K. In inferring, we set the beam size to 5.

Evaluation. For evaluation, we use two metrics: BLEU (Papineni et al., 2002) and Meteor (Lavie and Agarwal, 2007) to evaluate translation quality.

4.2 Experimental Results

Main results. Table 1 shows the performance of our approach, where `Ours-sent` and `Ours-doc` indicate the performance achieved by our approach when we use sentences or documents as input units, respectively. In the scenario where both sentence-level parallel dataset and monolingual documents are not used, we directly train our models from scratch with the two fine-tuning tasks on the fine-tuning datasets. #2 and #3 in the table show that our model is capable of translating both sentences and documents. Interestingly, when we use sentences as translation units, our models (i.e., #2 `Ours-sent`) outperform sentence-level Transformer baseline (i.e., #1 who uses sentences as input units in both training and inferring) over all translation tasks with improvement of averaged 1.36 BLEU and 1.72 Meteor. Moreover, when we use documents as translation units, our models (i.e., #3 `Ours-doc`) achieve further improvement by modeling document-level context. Compared to previous studies, it also shows that our approach surpasses all context-aware baselines on ZH-EN and EN-DE (TED) tasks and achieves the state-of-the-art on average.

In the scenario where both sentence-level parallel dataset and monolingual documents are used,[12] similar performance trends also hold. For example, #5 `Ours-sent` significantly exceeds Transformer

[10] https://github.com/sameenmaruf/selective-attn/tree/master/data

[11] Our code is available at https://github.com/strawberry116/Breaking-Corpus-Bottleneck-for-Context-Aware-NMT

[12] For Transformer baseline (i.e., #4 in the table), the two pre-training objectives in document-level restoration are context-agnostic.

Model	Bi-sent	Mo-doc	ZH-EN		EN-DE (News)	
			BLEU	Meteor	BLEU	Meteor
Trans.	✗	✗	39.64	27.56	22.03	41.37
Ours	✗	✗	41.27	28.46	24.70	44.38
Trans.	✓	✗	46.99	33.46	26.89	47.01
Ours	✓	✗	48.03	34.27	28.32	48.16
Trans.	✗	✓	40.32	28.64	24.62	44.83
Ours	✗	✓	42.64	30.19	25.30	45.60
Trans.	✓	✓	46.30	32.91	26.80	46.99
Ours	✓	✓	50.03	36.50	29.01	48.83

Table 2: Ablation studies on ZH-EN and EN-DE (News) translation tasks. Hereafter, we use `Ours` for `Ours-doc`, i.e., using documents as input units.

baseline with 1.85 BLEU and 1.78 Meteor on average while #6 `Outs-doc` further achieves the best performance.

Ablation study. We take ZH-EN and EN-DE (News) translations as representatives to study the effect of leveraging sentence-level parallel dataset and monolingual documents.

Table 2 compares the performance on the the test sets of ZH-EN and EN-DE (News) translations in different scenarios. From it, we have the following observations.

- Using either sentence-level parallel dataset or monolingual documents helps translation for both Transformer baselines and our context-aware models. However, in the presence of sentence-level parallel dataset, the Transformer baselines fail to achieve higher performance with monolingual documents, as we observe performance drops from 46.99 BLEU to 46.30 on Zh-EN, and from 26.89 to 26.80 on EN-DE. In contrary, our models achieve the highest performance by leveraging the two resources. This suggests the effectiveness of our approach in employing the two resources.

- It is not surprising to find out that the improvement is mainly contributed by using sentence-level parallel dataset, as translation model is more important than context encoder

- Finally, our approach consistently outperforms sentence-level Transformer in all scenarios. Encouraging, the performance gap becomes even larger on ZH-EN when more resources are used.

Fine-Tuning	Inferring-Input	BLEU
w/ sentence-level	document	50.03
	sentence	49.58
w/o sentence-level	document	50.10
	sentence	48.33

Table 3: Performance on ZH-EN translation with respect to different fine-tuning strategies and different input units in inferring.

Model	Bi-sent	Mo-doc	deixis	lex.c	ell.infl.	ell.VP
Trans.	✗	✗	50.0	45.3	52.0	27.3
Ours	✗	✗	62.3	47.9	64.9	36.0
Trans.	✓	✓	50.9	46.4	67.2	75.6
Ours	✓	✓	81.9	61.7	70.6	80.5

Table 4: Accuracy (%) of discourse phenomena.

5 Discussion

Next we use ZH-EN translation to analyze more on how our approach affects translation performance. See Appendix B for parameter analysis and statistics of the pre-trained models.

5.1 Effect of Joint Fine-tuning

In Section 3 we alternate sentence-level translation and document-level translation in fine-tuning. We investigate the effect of including sentence-level translation as a fine-tuning task. Table 3 compares the performance with respect to different fine-tuning strategies and different input units in inferring. When we use documents as input units in inferring, the joint fine-tuning strategy provides no advantage. However, when the input units are sentences, the joint fine-tuning strategy outperforms the one not including sentence-level translation in fine-tuning.

5.2 Analysis of Discourse Phenomena

We also want to examine whether the proposed approach actually learns to utilize document context to resolve discourse inconsistencies. Following Voita et al. (2019b) and Zheng et al. (2020), we use the same datasets to train model and contrastive test set for the evaluation of discourse phenomena for English-Russian by Voita et al. (2019b). There are four test sets in the suite regarding deixis, lexicon consistency, ellipsis (inflection and verb phrase). Each testset contains groups of contrastive examples consisting of a positive translation with correct discourse phenomenon and negative translations with incorrect phenomena. The goal is to figure out if a model is more likely to generate a cor-

Model	Bi-sent	Mo-doc	Dev	Test
Trans.	✗	✗	67.30	68.60
Ours	✗	✗	68.33	69.73
Trans.	✓	✓	71.02	70.51
Ours	✓	✓	72.11	70.89

Table 5: Evaluation on pronoun translations of ZH-EN.

Ratio (%)	Dev	Test
10	50.64	49.89
20	50.90	50.03
30	50.59	49.70

Table 6: Performance (BLEU scores) on dev and test sets of ZH-EN translation with respect to different gap sentence ratios in pre-training task of document-level restoration.

rect translation compared to the incorrect variation. We summarize the results in Table 4, which shows that in different scenarios our models are better at resolving discourse consistencies than context-agnostic baselines.

5.3 Pronoun Translation

We follow Miculicich et al. (2018) and Tan et al. (2019) to evaluate coreference and anaphora using the reference-based metric: accuracy of pronoun translation (Werlen and Popescu-Belis, 2017).

Table 5 lists the performance of pronoun translation. From it we observe that our proposed approach can well improve the performance of pronoun translations.

5.4 Effect of Gap Sentence Ratio

A significant hyper-parameter in the pre-training task of document-level restoration is the gap sentence ratio. A low ratio makes the document-level restoration less challenging while choosing gap sentences at a high ratio makes the global context have more overlapped. Table 6 shows that we achieve the best performance when the ratio is set as 20%.

5.5 Effect of Pre-training Objectives

As shown in Figure 2, we include two pre-training objectives in document-level restoration, i.e, CA-GSR and CA-MSR. To investigate the effect of CA-GSR, we use CA-MSR as the only objective in this pre-training task. In this way, the S_3 and S_5 in Figure 2 (a), for example, will be $\widehat{X_3}$ and $\widehat{X_5}$, respectively. Table 7 compares the performance when the pre-training task is of CA-MSR objective or combination of CA-GSR and CA-MSR.It shows the combining objective achieves better performance than using CA-MSR alone.

Pre-training Objective	Dev	Test
CA-GSR + CA-MSR	50.90	50.03
CA-MSR	50.61	49.73

Table 7: Performance (BLEU scores) on dev and test sets of ZH-EN translation with respect to different pre-training objectives in document-level restoration.

6 Related Work

We describe related studies in the following two perspectives.

6.1 Context-Aware NMT

Cache/Memory-based approaches (Tu et al., 2018; Kuang et al., 2018; Maruf and Haffari, 2018; Wang et al., 2017) store word/sentence translation in previous sentences for future sentence translation. Various approaches with an extra context encoders are proposed to model either local context, e.g., previous sentences (Jean et al., 2017; Wang et al., 2017; Zhang et al., 2018; Bawden et al., 2018; Voita et al., 2018, 2019b; Yang et al., 2019; Huo et al., 2020), or entire document (Maruf and Haffari, 2018; Mace and Servan, 2019; Maruf et al., 2019; Tan et al., 2019; Xiong et al., 2019; Zheng et al., 2020; Kang et al., 2020).

Besides, there have been several attempts to improve context-aware NMT with monolingual document data. To make translations more coherent within a document, Voita et al. (2019a) propose DocRepair trained on monolingual target language documents to correct the inconsistencies in sentence-level translation while Yu et al. (2020) train a context-aware language model to rerank sentence-level translations. Finally, Junczys-Dowmunt (2019) use source-side monolingual documents to explore multi-task training via the BERT-objective on the encoder. They simply concatenate sentences within a document into a long sequence, which is different from our approach.

6.2 Pre-training for Document-Level NMT

While there are substantial studies on improving sentence-level NMT with pre-training, we limit ourselves here to pre-training for document-level (context-aware) NMT. BART (Lewis et al., 2020) is a denoising auto-encoder model which learns to reconstruct the original document from a noised version. Inspired by BART, mBART (Liu et al.,

2020) is a model trained on a mixed corpus containing monolingual documents of different languages. Both BART and mBART concatenate sentences in one document into a long sequence, and thus fall into a standard sequence-to-sequence (seq2seq) framework. This is very different from our cross-task pre-training, in which we combine both context-agnostic learning and context-aware learning in a single model.

7 Conclusion

In order to leverage both large-scale sentence-level parallel dataset and source-side monolingual documents for context-aware NMT, in this paper, we have proposed a novel cross-task pre-training approach, which simultaneously learns to translate a sentence from source language to target language while denoising a document from deliberately noised to original. Upon the pre-trained models, we fine-tune them with document-level parallel dataset from both sentence-level and document-level perspectives. Experimental results on multiple document-level translation tasks have demonstrate the effectiveness of our approach. Finally, we also provide insights on how context-aware NMT benefits from our approach.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62036004 and 61876120).

References

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of NAACL*, pages 1304–1313.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit3: Web inventory of transcribed and translated talks. In *Proceedings of EAMT*, pages 261–268.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, pages 4171–4186.

Jingjing Huo, Christian Herold, Yingbo Gao, Leonard Dahlmann, Shahram Khadivi, and Hermann Ney. 2020. Diving deep into context-aware neural machine translation. In *Proceedings of WMT*, pages 604–616.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machinetranslation benefit from larger context? *Computing Research Repository*, arXiv:1704.05135.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, and Fernanda Viégas. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *TACL*, 5:339–351.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at wmt 2019: Towards large-scale document-level neural machine translation. In *Proceedings of WMT*, pages 225–233.

Xiaomian Kang, Yang Zhao, Jiajun Zhang, and Chengqing Zong. 2020. Dynamic context selection for document-level neural machine translation via reinforcement learning. In *Proceedings of EMNLP*, pages 2242–2254.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.

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 ACL 2007, System Demonstrations*, pages 177–180.

Shaohui Kuang, Deyi Xiong, Weihua Luo, and Guodong Zhou. 2018. Modeling coherence for neural machine translation with dynamic and topic caches. In *Proceedings of COLING*, pages 596–606.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of WMT*, pages 228–231.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL*, pages 7871–7880.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *TACL*, 8:726–742.

Valentin Mace and Christophe Servan. 2019. Using whole document context in neural machine translation. In *Proceedings of IWSLT*.

Sameen Maruf and Gholamreza Haffari. 2018. Document context neural machine translation with memory networks. In *Proceedings of ACL*, pages 1275–1284.

Sameen Maruf, André F. T. Martins, and Gholamreza Haffari. 2019. Selective attention for context-aware neural machine translation. In *Proceedings of NAACL*, pages 3092–3102.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of EMNLP*, pages 2947–2954.

Kishore Papineni, Salim Roukos, Ward Todd, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of ACL*, pages 1715–1725.

Xin Tan, Longyin Zhang, Deyi Xiong, and Guodong Zhou. 2019. Hierarchical modeling of global context for document-level neural machine translation. In *Proceedings of EMNLP-IJCNLP*, pages 1576–1585.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92.

Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. 2018. Learning to remember translation history with a continuous cache. *TACL*, 6:407–420.

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

Elena Voita, Rico Sennrich, and Ivan Titov. 2019b. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of ACL*, pages 1198–1212.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of ACL*, pages 1264–1274.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. In *Proceedings of EMNLP*, pages 2826–2831.

Lesly Miculicich Werlen and Andrei Popescu-Belis. 2017. Validation of an automatic metric for the accuracy of pronoun translation (apt). In *Proceedings of Workshop on Discourse in Machine Translation*, pages 17–25.

Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Modeling coherence for discourse neural machine translation. In *Proceedings of AAAI*, pages 7338–7345.

Zhengxin Yang, Jinchao Zhang, Fandong Meng, Shuhao Gu, Yang Feng, and Jie Zhou. 2019. Enhancing context modeling with a query-guided capsule network for document-level translation. In *Proceedings of EMNLP-IJCNLP*, pages 1527–1537.

Lei Yu, Laurent Sartran, Wojciech Stokowiec, Wang Ling, Lingpeng Kong, Phil Blunsom, and Chris Dyer. 2020. Better document-level machine translation with bayes rule. *TACL*, 8:346–360.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of EMNLP*, pages 533–542.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of ICML*.

Zaixiang Zheng, Xiang Yue, Shujian Huang, Jiajun Chen, and Alexandra Birch. 2020. Towards making the most of context in neural machine translation. In *Proceedings of IJCAI*, pages 3983–3989.

A Experimental Datasets

Table 8 summarizes statistics of the four translation tasks. Note that we split long documents into sub-documents with at most 30 sentences for efficient training.

Set	ZH-EN		EN-DE (Europarl)	
	#SubDoc	#Sent	#SubDoc	#Sent
Training	47,758	781,524	132,721	1,666,904
Dev	82	1,664	273	3,587
Test	627	5,833	415	5,134
Set	**EN-DE (TED)**		**EN-DE (News)**	
	#SubDoc	#Sent	#SubDoc	#Sent
Training	7,491	206,126	10,552	236,287
Dev	326	8,967	112	2,169
Test	87	2,271	184	2,999

Table 8: Statistics of the training, development, and test sets of the four translation tasks.

B More Result Analysis

B.1 Model Parameters

Table 9 presents the numbers of parameters for ZH-EN and EN-DE translations. Note that for all EN-DE translation tasks, the numbers of parameters are same as the vocabulary for them are shared. The table shows that our models introduce very limited parameters to encode document-level context.

Model	ZH-EN	EN-DE
Transformer	80.6M	61.4M
Ours	86.2M	64.0 M

Table 9: Model parameters for ZH-EN and EN-DE translations.

B.2 Statistics on Our Pre-trained models

Table 10 presents statistics on our two pre-trained models for ZH-EN and EN-DE translations. With 500K training steps, and within 120 (130) hours we complete 3.0 (1.2) and 35 (20) passes over the sentence-level parallel dataset and monolingual document dataset for Chinese (English), respectively.

	#Epoch	#Epoch	
Translation	**on Bi-sent**	**on Mo-doc**	**Time**
ZH-EN	35	3.0	120h
EN-DE	20	1.2	130h

Table 10: Statistics on our two pre-trained models for ZH-EN and EN-DE translations.