Machine Learning (2022) 111:917–935
https://doi.org/10.1007/s10994-021-06070-y

A study of BERT for context-aware neural machine translation

Xueqing Wu[1] · Yingce Xia[2] · Jinhua Zhu[1] · Lijun Wu[2] · Shufang Xie[2] · Tao Qin[2]

Received: 24 May 2021 / Revised: 13 August 2021 / Accepted: 22 September 2021 /
Published online: 9 January 2022

Abstract
Context-aware neural machine translation (NMT), which targets at translating sentences with contextual information, has attracted much attention recently. A key problem for context-aware NMT is to effectively encode and aggregate the contextual information. BERT (Devlin et al., in: NAACL, 2019) has been proven to be an effective feature extractor in natural language understanding tasks, but it has not been well studied in context-aware NMT. In this work, we conduct a study about leveraging BERT to encode the contextual information for NMT, and explore three commonly used methods to aggregate the contextual features. We conduct experiments on five translation tasks and find that concatenating all contextual sequences as a longer one and then encoding it by BERT obtains the best translation results. Specifically, we achieved state-of-the-art BLEU scores on several widely investigated tasks, including IWSLT'14 German→English, News Commentary v11 English→German translation and OpenSubtitle English→Russian translation.

Keywords Neural machine translation · BERT · Context-aware translation

This work was done when Xueqing Wu was an intern at Microsoft Research Asia.

Editors: Yu-Feng Li, Mehmet Gönen, Kee-Eung Kim.

✉ Yingce Xia
yingce.xia@microsoft.com

Xueqing Wu
shirley0@mail.ustc.edu.cn

Jinhua Zhu
teslazhu@mail.ustc.edu.cn

Lijun Wu
lijuwu@microsoft.com

Shufang Xie
shufxi@microsoft.com

Tao Qin
taoqin@microsoft.com

[1] Universit of Science and Technology of China, Hefei, Anhui, China

[2] Microsoft Research Asia, Beijing, China

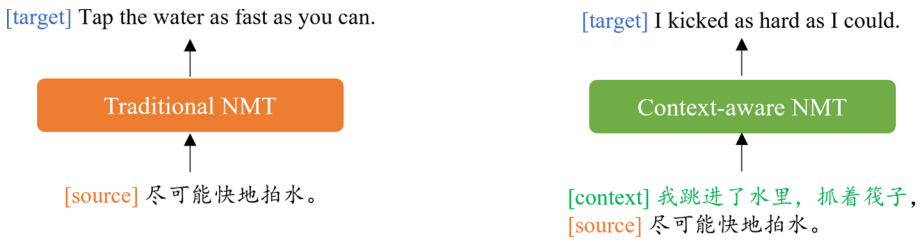

Fig. 1 An illustration of context-aware NMT. Compared to traditional NMT, context-aware NMT uses contextual information as additional inputs to assist the translation of the source sequences

1 Introduction

Neural machine translation (briefly, NMT), which aims to translating sequences from a source language to a target language, has achieved great success (Hassan et al., 2018; Ng et al., 2019). Recently, a trend is to leverage contextual information (i.e., the surrounding sentences of the one to be translated) to improve NMT (Junczys-Dowmunt, 2019; Zheng et al., 2020; Xiong et al., 2019b). Contextual information is available in many scenarios, e.g., translating news summaries, movie subtitles, dialog, etc. In other words, compared to traditional NMT, context-aware NMT takes the contextual information as additional inputs. These additional inputs are only used to assist in the translation of the source sequences and do not need to be translated. An illustration is shown in Fig. 1. With the help of contextual information, the translation coherence for adjacent sentences (i.e., discourse coherence) can be improved (Voita et al., 2019; Xiong et al., 2019a). A core problem of context-aware translation is how to encode and aggregate the contextual information effectively.

Pre-training methods like ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and Electra (Clark et al., 2020) have demonstrated great success in natural language understanding (Devlin et al., 2019). While Zhu et al. (2020) has verified the effectiveness of applying BERT into conventional NMT, to the best of our knowledge, there is no extensive study on applying pre-trained models into context-aware NMT. Therefore, we explore along this direction to use BERT to improve context-aware NMT. In particular, a pretrained BERT model is used as an additional encoder to encode contextual information (Zhang et al., 2018a; Miculicich et al., 2018; Voita et al., 2018), resulting in a group of features carrying contextual information.

We study three most common methods to aggregate the contextual features, and the illustration is shown in Fig. 2:

(1) *Concatenation mode (C-mode)* Concatenate all contextual sentences as a longer one (Agrawal et al., 2018; Zhang et al., 2018b), and then feed it into BERT to get the contextual features.
(2) *Flat mode (F-mode)* Encode each contextual sentence independently using BERT, and concatenate their outputs as the contextual features (Maruf et al., 2019a).
(3) *Hierarchical mode (H-mode)* Aggregate the contextual features hierarchically (Miculicich et al., 2018; Maruf et al., 2019a), where the features obtained by scheme (2) are further processed by two levels of attention models (a word-level attention model and a sentence-level attention model).

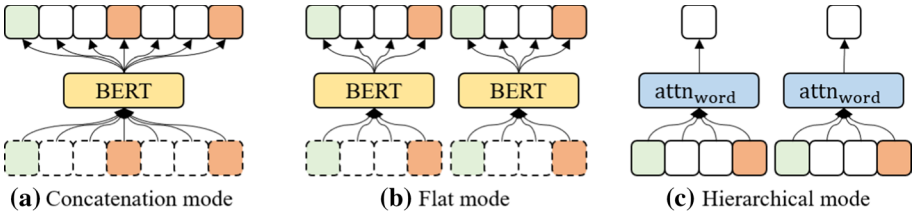

Fig. 2 Different ways to encode contextual information. We take a two-sentence case as an example. The green and orange blocks represent [CLS] and [SEP]. The bottom row and top row denote input and output respectively. The input of (**c**) is the output of (**b**) (Color figure online)

We conduct experiments on five translation tasks. We first verify that using BERT as a contextual encoder can bring promising improvements over previous methods (see Table 5). Then we find that among different contextual feature aggregation mechanisms, concatenating all contextual sequences (i.e., C-mode) is the best choice in terms of translation accuracy. F-mode can achieve comparable performance to C-mode on several tasks with lower computational complexity. Compared to these two methods, H-mode has limited advantages. With our proposed context-aware NMT with BERT, we achieve state-of-the-art BLEU scores on several widely investigated tasks, including IWSLT'14 German→English, News Commentary v11 English→German translation and OpenSubtitle English→Russian translation. We further conduct ablation study to verify that BERT is an effective contextual information encoder and can significantly help context-aware NMT (see Table 8).

The remaining part is organized as follows: Related work is introduced in Sect. 2. The network architecture and different ways to aggregate contextual information are described in Sect. 3. Experiments are summarized in Sects. 4 and 5 concludes this paper.

2 Related work

In this section, we introduce the related work on pre-training and context-aware neural machine translation.

2.1 Pre-training approaches

Pre-training has a long history in machine learning. Pre-trained models can be used to extract features for the input data, and can be finetuned for specific downstream tasks (Sermanet et al., 2013; Girshick et al., 2014). In natural language processing (NLP), a common method of pre-training is to train a language model on a large-scale unlabelled corpus (Peters et al., 2018; Radford et al., 2018; Radford et al., 2019; Devlin et al., 2019; Lewis et al., 2020). Specially, BERT (Devlin et al., 2019) is one of the most adopted approaches. It is pre-trained with Masked Language Modeling (MLM) task and Next Sentence Prediction (NSP) task, and has been applied to different downstream tasks such as open-domain question answering (Yang et al., 2019b) and document classification (Adhikari et al., 2019). Recently, the use of BERT in NMT has also been explored (Zhu et al., 2020; Yang et al., 2019a). However, there is no extensive study on applying BERT or other pre-trained models into context-aware NMT. Another related work is mBART (Liu et al.,

2020), a sequence-to-sequence model pretrained on large-scale monolingual corpora in multiple languages. It can be finetuned on downstream translation tasks and boost the performance compared with traditional NMT. mBART can also be finetuned on context-aware NMT, where it translates the entire document instead of translating individual sentences with the help of the context. We show that our method outperforms mBART.

2.2 Context-aware NMT

Network architecture The single-encoder architecture uses conventional NMT architecture and directly takes the concatenation of contextual sentences and source sentence as the input (Tiedemann and Scherrer 2017). Li et al. (2019) further used BERT to initialize the encoder, and adopted context manipulation and multi-task training to model large context. Similarly, Ma et al. (2020) adopted segment embeddings to distignuish the source sequence from contextual information, and used BERT to initialize the encoder.

To better encode the contextual information, a more common choice is the multi-encoder approach, where an additional set of encoders and attention models are introduced to encode and aggregate contextual information (Jean et al., 2017; Zhang et al., 2018b). Voita et al. (2018) and Müller et al. (2018) further employed weight sharing between the input encoder and the context encoder, and demonstrated that weight sharing can help train a stronger context encoder. Miculicich et al. (2018) proposed to use a hierarchical attention network (HAN) (Yang et al., 2016) to aggregate contextual information using word-level and sentence-level abstractions, and Yun et al. (2020) further use fully connected self attention to conduct sentence-level abstraction. Maruf et al. (2019a) used sparse attention to aggregate contextual information in order to select only the useful information. Kang et al. (2020) further proposed a context selection module trained via reinforcement learning to adaptively select the useful contextual information. Our work is based on multi-encoder approach, and we use BERT as the contextual encoder.

Another branch of context-aware NMT is to use post-processing models, where additional modules are introduced to refine the output from a standard NMT system leveraging contextual information (Voita et al., 2019; Xiong et al., 2019a; Zheng et al., 2020). A comprehensive survey is provided by Maruf et al. (2019b).

Although previous work on context-aware NMT usually translates each individual sentence with the help of its context, some recent work attempts to directly translate the entire document. For example, mBART used multi-language pretrained model to conduct document-level translation (Liu et al., 2020), and Bao et al. (2021) used group tagging and group attention to introduce locality into the encoder-decoder attention.

Evaluation The evaluation of discourse coherence of context-aware NMT has also drawn much attention. Voita et al. (2018) observed that the context-aware NMT model implicitly captures anaphora. Bawden et al. (2018) created a contrastive test set for discourse phenomena evaluation. Voita et al. (2019) conducted a human study on English-Russian translation, identified three main sources of document-level translation inconsistencies, and released a series of test sets targeting these phenomena. Wong et al. (2020) constructed a test suite targeted at cataphora translation and found that context-aware NMT outperforms traditional NMT.

Analysis Some recent work analyzed the conditions and reasons of the improvements brought by context-aware NMT. Kim et al. (2019) demonstrated that a small context and a minimal context encoder is sufficient for context-aware NMT. Li et al. (2020) showed that replacing the context sentence with an unrelated sentence does not affect the performance

of the context-aware NMT model. We reproduce their experiments with our BERT-based NMT model and get some new conclusions (see Table 8).

3 Algorithm

In this section, we will introduce notations, formulations and model architecture in Sect. 3.1, and then introduce methods to aggregate contextual information in Sect. 3.2.

3.1 Model architecture

Notations and formulations Let $\mathcal{X}$ and $\mathcal{Y}$ denote the source language space and target language space respectively. $x^{\text{curr}} \in \mathcal{X}$ is the source sentence to be translated, and $y^{\text{curr}} \in \mathcal{Y}$ is the corresponding target sentence. We denote the lengths of x^{curr} and y^{curr} as l_x and l_y respectively. The contextual information of x^{curr}, denoted as $\mathcal{C}(x^{\text{curr}})$, contains the preceding sequences and succeeding sequences of x^{curr}. Assume there are N contextual sentences in $\mathcal{C}(x^{\text{curr}})$, and let $x_k^{\text{ctx}} \in \mathcal{X}$ denote the kth one. In our work, x^{curr} is also included in $\mathcal{C}(x^{\text{curr}})$. Let $\texttt{BERT}$ denote a BERT model. Given an input x with T_x units (e.g., words, subwords), $\texttt{BERT}(x)$ outputs a representation $(h_1, h_2, \ldots, h_{T_x})$, where h_i is the representation of x_i (i.e., the ith unit in x).

Let $\texttt{attn}(q, K, V)$ denote the attention layer in Transformer, where q, K and V are query, key and value respectively. The key K and value V are two tuples with the same number of elements, where $k_i \in K$ and $v_i \in V$ are the ith key-value pair, $i \in \{1, 2, \ldots, |K|\}$. The attention layer is mathematically defined as follows:

$$\begin{aligned} \texttt{attn}(q, K, V) &= \sum_{i=1}^{|V|} \alpha_i W_v v_i, \\ \alpha_i &= \frac{\exp\left((W_q q)^T (W_k k_i)\right)}{Z}, \; Z = \sum_{i=1}^{|K|} \exp((W_q q)^T (W_k k_i)), \end{aligned} \tag{1}$$

where W_q, W_k and W_v are parameters to be trained. In Vaswani et al. (2017), $\texttt{attn}$ is implemented as a multi-head attention model, where the outputs of multiple conventional attention models are concatenated.

Let $\texttt{FFN}$ denote the feed-forward layer in Transformer, and it is defined as follows:

$$\texttt{FFN}(x) = W_2 \texttt{ReLU}(W_1 x + b_1) + b_2, \tag{2}$$

where x is the input; $\texttt{ReLU}(x) = \max(x, 0)$; and W_1, W_2, b_1, b_2 are the parameters to be learned.

Model architecture We adapt the architecture proposed by Zhu et al. (2020) into a context-aware version, which is shown in Fig. 3. C is the contextual features outputted by BERT, and the encoder and the decoder are used to encode x^{curr} and generate y^{curr} respectively. Both the encoder and decoder are L-block stacked networks. For ease of reference, briefly denote $\{1, 2, \ldots, L\}$ as $[L]$.

Integration of contextual features To integrate the contextual features C, we use two parallel attention modules: one is the original attention module (i.e., self-attention or encoder-decoder attention), and the other is used to aggregate the contextual feature C (i.e., Ctx-Enc

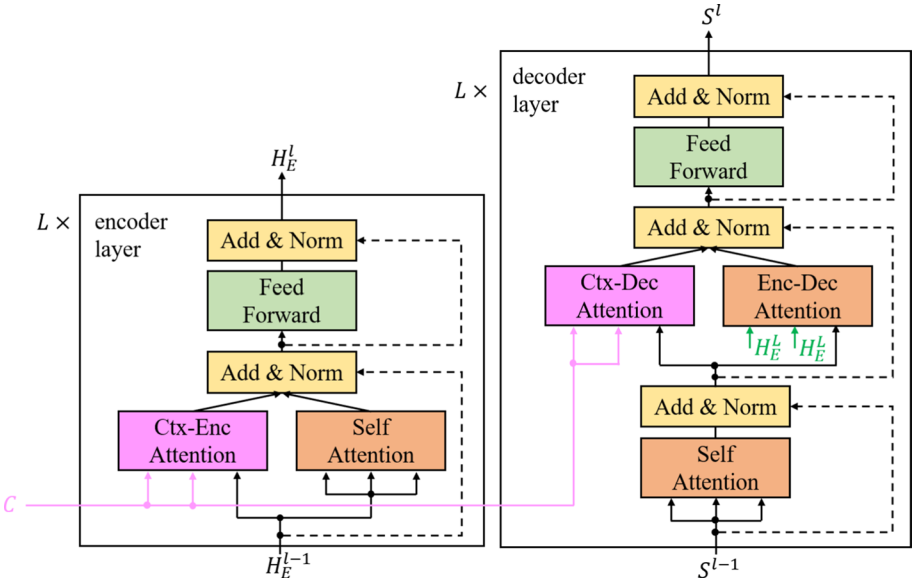

Fig. 3 Network architecture. C is the contextual features outputted by BERT, and can be calculated using different modes (i.e. C-mode, F-mode, and H-mode). The encoder and decoder are L-block stacked networks, and are used to encode the input and generate the output respectively. Both the encoder and decoder utilize the contextual information C

attention and Ctx-Dec attention). The outputs of the two parallel modules are averaged as shown in Eqs. (3) and (4).

We also use drop-net technique to train the parallel attention (Zhu et al., 2020). For either branch of attention, we pick its output as the final output with probability $p_drop - net$ $(0 \le p_drop - net \le 0.5)$; w.p. $(1 - 2p_drop - net)$, we use both the two attention outputs by averaging them. This ensures that both the contextual features C and the conventional NMT model are fully utilized.

Encoder: Let h_i^l denote the hidden representation of the i-unit in block l, $i \in [l_x]$, $l \in [L]$. Let H_E^l denote $(h_1^l, h_2^l, \dots, h_{l_x}^l)$. Specially, H_E^0 is the embedding of x^{curr}. $\texttt{attn}_S$ and $\texttt{attn}_C$ represent the self-attention layer and context-encoder attention layer (i.e., the Ctx-Enc module in Fig. 3), where the parameters of different layers are different. We then have

$$\begin{aligned} \hat{h}_i^l &= \frac{1}{2}\left(\texttt{attn}_S(h_i^{l-1}, H_E^{l-1}, H_E^{l-1}) + \texttt{attn}_C(h_i^{l-1}, C, C)\right), \forall i \in [l_x]; \\ H_E^l &= (\texttt{FFN}(\hat{h}_1^l), \dots, \texttt{FFN}(\hat{h}_{l_x}^l)). \end{aligned} \tag{3}$$

As shown in Eq. (3), the contextual features in C are fed into the Ctx-Enc attention. The output is averaged with the output of a self-attention layer, and the result is further processed by the FFN layer. We will get H_E^L from the last layer of the encoder.

Decoder: We use s_t^l to denote the hidden state of the t-unit in the lth block of the decoder, $t < l_y$, $l \in [L]$. Define $S_{<t}^l = (s_1^l, \dots, s_{t-1}^l)$. At the lth block, we have

$$\begin{aligned}\bar{s}_t^l &= \texttt{attn}_S(s_t^{l-1}, S_{<t+1}^{l-1}, S_{<t+1}^{l-1});\\ \hat{s}_t^l &= \frac{1}{2}\big(\texttt{attn}_C(\bar{s}_t^l, C, C) + \texttt{attn}_E(\bar{s}_t^l, H_E^L, H_E^L)\big),\\ s_t^l &= \texttt{FFN}(\hat{s}_t^l).\end{aligned} \tag{4}$$

Compared to the encoder, an additional encoder-decoder attention $\texttt{attn}_E$ is required. After obtaining s_t^L output from the last layer, it can be mapped to the predicted tth word.

3.2 Contextual features

We use BERT to extract contextual features. There are two special tokens in BERT: [CLS], which is padded at the start of a sentence; and [SEP], which is used to separate different sequences. We study three different ways/modes to aggregate the contextual information. The illustrations are in Fig. 2.

(1) **Concatenation mode** (C-mode): Concatenate the contextual sentences as a longer sequence, i.e.,

$$x_{\text{ctx}} = \big\langle [\texttt{CLS}], x_1^{\text{ctx}}, [\texttt{SEP}], x_2^{\text{ctx}}, [\texttt{SEP}], \ldots, [\texttt{SEP}], x_{N-1}^{\text{ctx}}, [\texttt{SEP}], x_N^{\text{ctx}}, [\texttt{SEP}] \big\rangle, \tag{5}$$

and encode x_{ctx} using BERT. $\langle \ldots \rangle$ denotes concatenating all the input tokens and sequences as a longer sequence. The contextual features extracted by BERT are $C = \texttt{BERT}(x_{\text{ctx}})$.

(2) **Flat mode** (F-mode): Each contextual sequence is independently encoded and the output features are concatenated together. For any $x \in \mathcal{C}(x^{\text{curr}})$, define

$$H_B(x) = \texttt{BERT}(\langle [\texttt{CLS}], x, [\texttt{SEP}] \rangle). \tag{6}$$

The contextual feature C is obtained by concatenating all $H_B(x)$ together, i.e.,

$$C = \big\langle H_B(x_1^{\text{ctx}}); H_B(x_2^{\text{ctx}}); \ldots; H_B(x_N^{\text{ctx}}) \big\rangle,$$

where we reuse $\langle \ldots \rangle$ to represent feature concatenation.

(3) **Hierarchical mode** (H-mode): To use hierarchical attention (Miculicich et al., 2018), we need a word-level attention model $\texttt{attn}_{\text{word}}$ and a sentence-level attention model, which is exactly what the Ctx-Enc and Ctx-Dec models do in Fig. 3. H-mode is built on top of the F-mode. Under H-mode, the contextual information C is different at each block in the encoder and decoder. At the lth block in the encoder, for any $x \in \mathcal{C}(x^{\text{curr}})$, each $H_B(x)$ is processed into a vector through the word-level attention:

$$W_B(x) = \texttt{attn}_{\text{word}}(h_i^{l-1}, H_B(x), H_B(x)). \tag{7}$$

Then the corresponding contextual information C for the lth layer in the encoder is $C = \{W_B(x) | x \in \mathcal{C}(x^{\text{curr}})\}$. Similarly, in the l-block of the decoder, the contextual information C is obtained as

$$\tilde{W}_B(x) = \texttt{attn}_{\text{word}}(s_i^{l-1}, H_B(x), H_B(x)),\ C = \{\tilde{W}_B(x) | x \in \mathcal{C}(x^{\text{curr}})\}. \tag{8}$$

In this way, the contextual features are processed in a hierarchical manner, where a word-level attention model is applied first, followed by a sentence-level attention model, i.e. the

context-encoder and the context-decoder attention models. The illustration is shown in Fig. 2c.

3.3 Discussion

With C-mode, all the contextual sentences are encoded together as a longer sequence, so that each unit can attend to units from all contextual sentences. With F-mode, each unit can only attend to units in the same contextual sequence. Therefore, C-mode can capture longer-range dependencies and is expected to achieve better translation quality than F-mode. However, C-mode requires more memory and computation than F-mode, as discussed in the next paragraph. H-mode provides a more adaptive way to aggregate contextual features, where the words within a contextual sentence are explicitly aggregated into one vector, serving as the sentence representation. The sentence-level attention module further processes the sentence representations. Compared to C-mode and F-mode, the contextual information C for each block in the encoder and decoder are different due to the existence of $\texttt{attn}_{\text{word}}$.

Let L_B/L, l_d/l_i and N denote the numbers of layers of BERT/NMT module, average sequence length of contextual sequence/input sentence to be translated, and number of contextual sequences. The time complexity of obtaining the contextual features for C-mode, F-mode and H-mode are $O(L_B N^2 l_a^2)$, $O(L_B N l_a^2)$ and $O(L_B N l_a^2 + 2 l_i N)$ respectively. That is, F-mode and H-mode are expected to be more efficient than C-mode in terms of inference speed.

4 Experiments

We conduct experiments on five translation tasks to study the effectiveness of leveraging BERT in context-aware NMT.

4.1 Settings

Dataset For ease of reference, denote English, German, Chinese and Russian as En, De, Zh and Ru respectively. In this work, we work on two types of context-aware translation tasks and five translation tasks in total. The statistics of all datasets are listed in Table 1.

For the first type, the training corpus is provided with paragraph borders. Therefore, we can extract contextual information for all sequences.

(1) IWSLT'14 En↔De: Following Edunov et al. (2018), we lowercase all words, tokenize them, and apply BPE with 10k merge operations (Sennrich et al., 2016) jointly on the source and target corpus. We randomly split 64 documents from the training corpus as the validation set. The test set is the concatenation of *tst2010, tst2011, tst2012, dev2010* and *dev2012*.
(2) OpenSubtitle2018 En↔Zh: We clean the dataset with a set of predefined rules, which is provided in the supplementary material. The Chinese sentences are segmented using

Table 1 Dataset statistics

Task	Language	Train	Valid	Test	BPE ops
IWSLT'14	En↔De	160k	7k	6.8k	10k/shared
OpenSubtitle'18	En↔Zh	2M	10.3k	10.4k	10k/not shared
News	En→De	0.4M	3k	34k	10k/shared
OpenSubtitle'18	En→Ru	6M/1.5M	10k	10k	32k/shared
WMT	En→De	5.18M/1.25M	3k	9k	32k/shared

The first two columns represent the task and language pairs. The next three columns represent the numbers of sentences in training, validation and test sets. In the last two rows of "train" column, both the number of total bilingual sequences and that of context-aware sequences are reported. The last column represents the number of BPE merge operation and whether the source and target corpus are concatenated to obtain the BPE mapping table

Jieba.[1] We apply BPE with 10k merge operations on both English and Chinese (not shared). The validation and test sets are split from the corpus without overlap to the training set.

(3) News En→De: Following Zheng et al. (2020), we choose News Commentary v11 as the training set. We use WMT newstest2015 and newstest2016 as the validation and test sets respectively. For the second type, the training corpus consists of two parts. One part consists of bilingual sentences only without contextual information, and the other contains sentences with contextual information.

(4) OpenSubtitle2018 En→Ru: The dataset is released by Voita et al. (2019), with tokenized training, validation, and test sets. The training set contains 6M sentences, with 1.5M sentences containing three previous sentences as contextual information. A held-out dataset is used to evaluate discourse phenomena.

(5) WMT'19 En→De: In WMT'19 En→De, there are 5.18M bilingual data, among which 1.25M sentences are context-aware. We choose *newstest 2015* as the validation set and *newstest 2016–2019* as test sets.

Model We choose Transformer as the backbone of the translation model. For En↔Ru and En↔Zh, the embedding dimension, feed-forward layer dimension and number of encoder/decoder blocks are 512, 2048 and 6 respectively. For IWSLT En↔De and News V11 En→De, we use the same architecture but change the feed-forward layer dimension into 1024. For WMT En→De, we use the `transformer_big` configuration, where the embedding dimension, feed-forward layer dimension, and the number of encoder/decoder blocks are 1024, 4096 and 6 respectively. The dropout rate is 0.1 for En↔Zh and 0.3 for remaining tasks.

For BERT, we use the pretrained BERT models released by `transformers` package.[2] For IWSLT De→En, OpenSubtitle Zh→En and WMT En→De translation, we use `bert-base-german-cased`, `bert-base-chinese` and `bert-large` respectively. For the remaining tasks, we use `bert-base` model.

[1] https://github.com/fxsjy/jieba.

[2] https://github.com/huggingface/transformers.

Choice of contextual information In this work, we explore several ways to select the contextual sequences: (1) Choose m sentences before x^{curr} as the contextual information (denoted as "(m prev)" in the results). (2) Choose n sentences after x^{curr} as the contextual information (denoted as "(n next)"). (3) Choose one previous sentence and one succeeding sentence of x^{curr} as the contextual information (denoted as "(1 prev, 1 next)"). x^{curr} is included in each setting.

Training strategy For all tasks, we first train a conventional sentence-level NMT model using all available data, and then finetune the context-aware NMT model warm started from the sentence-level model on the context-aware data. We use Adam optimizer (Kingma and Ba, 2015) with learning rate 5×10^{-4} and inverse square root learning rate scheduler (Vaswani et al., 2017). The IWSLT En↔De, OpenSubtitle En→Ru/En↔Zh and News En→De are trained on a single GPU with batchsize 6k. The models for WMT En→De are trained on eight GPUs with the effective batchsize as $64k$ tokens per GPU. The drop-net is fixed as 0.5. All models are trained until convergence, and we choose the best checkpoint on the validation set.

Evaluation For IWSLT'14 En↔De and OpenSubtitle En↔Zh tasks, we use beam search with beam width 5 and length penalty 1.0 to generate sentences. For En→Ru task, we use beam search with beam width 5 and length penalty 0.0 to generate sentences since the dataset contains many short sentences. For IWSLT'14 En↔De translation and Opensubtitle En→Ru, following the common practice, we use `multi-bleu.perl` to evaluate the translation quality.[3] For other tasks, we use `sacreBLEU` for evaluation.[4]

4.2 Results with contextual sequences only

The results for IWSLT'14 En↔De and OpenSubtitle En↔Zh are shown in Table 2, and the results for news En→De are reported in Table 5.

Generally, encoding contextual information with BERT can significantly boost the performances compared to the vanilla Transformer baseline (i.e., the row "Transformer"). The BERT-NMT (Zhu et al., 2020) model improves the standard Transformer on En↔De and En→Zh by large margins. Compared to BERT-NMT baseline, leveraging contextual information is helpful. We have the following observations:

(1) Among the three different contextual information encoding methods, C-mode outperforms the other two (i.e., F-mode and H-mode) in terms of BLEU score. Taking De→En translation with contextual information "m prev" (m=1, 2, 3) as an example, the best BLEU score that C-mode can obtain is 36.64, while the scores for F-mode and H-mode are 36.48 and 36.15. Compared with F-mode, in C-mode, the contextual information is attended to a longer range, so that the contextual information can be utilized more effectively. Although HAN (Miculicich et al., 2018) achieves good results on the context-aware NMT, the benefits of H-mode gradually disappears when using BERT as contextual information encoder. Comparing the results of F-mode and H-mode across different tasks in Table 2, H-mode has no significant advantages. That is, the features encoded by BERT can be directly applied to NMT model without leveraging the $\texttt{attn}_{\text{word}}$.

On the other hand, on some tasks like De→En and Zh→En, using previous contextual information only, the best BLEU scores that C-mode and F-mode can achieve are within

[3] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl.
[4] https://github.com/mjpost/sacreBLEU.

Table 2 BLEU scores on IWSLT'14 En↔De and OpenSubtitle En↔Zh datasets

Model	Context	En→De	De→En	En→Zh	Zh→En
Transformer	–	28.51	35.08	18.17	19.43
BERT-NMT	–	30.45	36.11	19.57	19.60
C-mode	1 prev	30.69	36.50	19.40	20.24
	2 prev	30.65	36.64	19.92	20.20
	3 prev	30.75	36.51	**20.06**	**20.50**
	1 next	30.66	36.57	19.51	20.14
	2 next	30.80	36.45	19.84	20.12
	3 next	**30.89**	36.63	19.67	20.44
	1 prev, 1 next	30.75	**36.84**	19.87	20.34
F-mode	1 prev	30.28	36.33	19.17	19.99
	2 prev	30.25	36.25	19.47	20.18
	3 prev	30.21	36.48	19.27	20.06
H-mode	1 prev	30.25	35.93	18.91	19.95
	2 prev	30.32	36.03	18.99	19.84
	3 prev	30.22	36.15	18.96	19.43

Bold numbers to denote the best results for each task

0.3 BLEU gap. If inference speed or memory efficiency is more important to the user, F-mode is a better choice.

(2) On English↔German translation, models using future information (i.e., the sentences after the one to be translated) generally outperform models that only use historical information (preceding sentences). On En→De translation, leveraging three next sentences as contextual information is the best strategy. On De→En, the best strategy is to choose one next sentence associated with one previous sentence as the contextual information. This shows that the future information in the source side is also important for the translation quality.[5]

(3) When using BERT, there is no consistent result on how many sequences should be selected as contextual information. A general trend shown in Table 2 is that using three contextual sequences is better than using one only, no matter whether leveraging the previous sequences or the next ones.

Computational overhead We counted the inference speed and memory consumption of each model on the IWSLT'14 De→En task, and the results are shown in Table 3. For all models, we set the batch size as 128 sentences per batch.

As expected, F-mode outperforms C-mode in terms of both inference speed and memory efficiency. Although the asymptotic space complexity of H-mode is smaller than that of F-mode, it actually requires more calculations and more memory due to the two levels of attention.

Case study To demonstrate the effectiveness and necessity of the context-aware NMT, we show three examples from OpenSubtitle Zh→En task and report them in Table 4, where *Src*, *Ref*, H_S and H_C stand for input, reference, sentence-level translation and context-aware

[5] We tried the setting (1 prev, 1 next) on IWSLT En↔De with F-mode and H-mode. The corresponding BLEU scores for En→De are 30.23 (F-mode) and 30.11 (H-mode), and those for De→En are 36.16 (F-mode) and 35.98 (H-mode), which are outperformed by the corresponding scores in C-mode. Therefore, we did not work on all the settings of F-mode and H-mode due to resource limitation.

Table 3 The inference speed (sentences per second) and memory consumption (GB) of different models on IWSLT'14 De→En

Context	Model	Sent (s)	Mem (GB)
–	Transformer	117.44	6.4
	BERT-NMT	72.11	12.3
1 prev	C-mode	64.94	13.4
	F-mode	66.72	12.8
	H-mode	45.86	17.3
2 prev	C-mode	56.41	15.2
	F-mode	63.71	14.2
	H-mode	51.67	22.2
3 prev	C-mode	49.69	16.1
	F-mode	60.22	14.8
	H-mode	40.48	27.4

Table 4 Examples of Zh→En translation

Src	我跳进了水里，抓着筏子。尽可能快地拍水。
Ref	i jumped in the water. i was actually in the water, I grabbed the rifles. I kicked as hard as I could.
H_S	I jumped into the water and grabbed a raft. Tap the water as fast as you can.
H_C	I jumped into the water, and grabbed the raft. I took as fast as I could.
Src	突然发现左边有个小男孩,他肩膀上扛着一个水罐
Ref	all of a sudden there was this small boy on our left. he was carrying a water container over his shoulder .
H_S	suddenly, there was a little boy on the left. he carried a water can on his shoulder .
H_C	suddenly, there was a little boy on the left. he was carrying a water tank on his shoulder .
Src	-哈喽丹妮尔 -你怎知道是我
Reft	- hello , daniella . - how did you know it was me ? .
H_S	- hello , danielle. - how do you know it's me?
H_C	- hello , danielle. - how did you know it was me?

translation. The contextual sequences and the input to be translated are colored by red and blue respectively. We can see that there are several advantages of context-aware NMT over the sentence-level translation model:

(1) *Subjects* In the first example, H_S gives the correct sentence-level translation, but no subject is inferred. According to the previous source sentence, we know the subject should be "I". H_C correctly recovers the "I" based on the contextual information.

Table 5 Results on news v11 En→De translation

Model	BLEU
Transformer	24.2
BERT-NMT	27.1
Ours (1 prev)	27.7
Ours (1 next)	28.2
HAN (Miculicich et al., 2018)	25.03
SAN (Maruf et al., 2019a)	24.84
Zheng et al. (2020)	24.91
Flat-Transformer (Ma et al., 2020)	24.52
HAN + DCS-PF (Kang et al., 2020)	27.61
G-Transformer (Bao et al., 2021)	26.14

The choice of contextual information is reported within brackets. C-model is used for experiments

(2) *Consistency* In the second example, we should use past progressive, which is more consistent in the context. In H_S, the usage of past tense makes it inconsistent with the complete translation, while such a drawback is overcome in H_C.
(3) *Tenses* In the third example, we know this is a conversation. It is a common practice to use "how did you know it was me" in the chat. Therefore, H_C gives a better translation result.

For news En→De translation, we work on C-mode only, since it achieves the best BLEU scores among all three modes. The results are in Table 5. The baseline BLEU scores of the standard Transformer and BERT-NMT are 24.2 and 27.1 respectively, where BERT-NMT has already significantly outperformed most of the recently proposed method (Miculicich et al., 2018; Maruf et al., 2019a; Zheng et al., 2020; Ma et al., 2020; Bao et al., 2021) for context-aware translation. Kang et al. (2020) performs better than BERT-NMT but worse than our method. By using BERT to encode contextual sentences, we can boost the scores to 27.7/28.2 with one previous/next sentence as contextual information. The results on news En→De also verify that leveraging future contextual information is helpful to boost the performances of En→De.

4.3 Results with a mixed sequences

We then analyze a more realistic setting where a large-scale context-agnostic data and a small-scale context-aware data is available.

The results for En→Ru are shown in Table 6. Following Voita et al. (2019), we reproduce the baseline and get 32.73 BLEU score, which is slightly better than that in Voita et al. (2019). Then we finetune the baseline using contextual sequences. In the test set, each sentence is provided with three previous sentences as contextual information. Therefore, we only try C-mode using preceding sequences. By using contextual information, the BLEU scores can be improved from 32.73 to 33.26, outperforming the CADec (Voita et al., 2019). We observe that in this task, the translation quality benefits from more contextual sequences.

Table 6 Results on OpenSubtitle En→Ru

Model	Context	BLEU	Ellipsis infl/VP
Results from Voita et al. (2019)			
Transformer	–	32.40	53.0/28.4
CADec	3 prev	32.38	72.2/80.0
Our results			
Transformer	–	32.73	53.0/28.6
BERT-NMT	–	32.92	54.2/29.6
Ours	1 prev	33.16	60.6/59.2
	2 prev	33.22	63.0/66.2
	3 prev	33.26	62.6/68.4

Numbers of contextual sequences are listed with brackets

We then evaluate our method on the OpenSubtitle En→Ru contrastive test sets for discourse phenomena. Each sample in the test set consists of a translation with correct discourse phenomenon and several translations with incorrect phenomena. Specifically, we evaluate our model on two tasks: Ellipsis infl and Ellipsis VP,[6] and the results are reported in the last column of Table 6. Compared to the baseline, our methods achieve promising improvements on ellipsis task. The best ellipsis inflection is achieved with two contextual sentences, and the ellipsis VP increases w.r.t. the number of contextual sequences. There is some gap between our proposed model and CADec, which leverages both source side and target side contextual sequences. We will explore how to leverage the contextual information from the target side in the future.

Finally, we conduct experiments on WMT En→De translation. Due to resource limitation, we only work on using one future sentence as contextual information and the experiments are conduted under C-mode. The results are in Table 7. Compared to the Transformer baseline, leveraging contextual information can make significant improvement, i.e., about 1 point BLEU improvement on each task. Our method is also significantly better than BERT-NMT with $p < 0.01$ on news16 and news18, and $p < 0.05$ on news17 and news19 (Koehn, 2004). The results on OpenSubtitle En→Ru and WMT En→De demonstrate that given a relatively large amount of bilingual data without contextual information and a portion of context-aware data, our method can still improve the translation quality. Our model also outperforms mBART on news19 testset, i.e., 40.0 v.s. 37.1 (Liu et al., 2020), with comparable number of parameters (600M for our model, and 610M for mBART).

4.4 Analysis

In this section, we answer the following questions:

(1) *Does our model learn to capture contextual information?* Following Li et al. (2020), we implement two variants of our method by replacing the context sentence with a

[6] We do not evaluate our model on the other tasks, because the other tasks require modeling target side context, while our model can only utilize source side context. We will explore leveraging BERT into the target side in the future.

Table 7 Results on WMT En→De translation

	news16	news17	news18	news19
Transformer	35.2	29.0	42.2	39.0
BERT-NMT	35.8	29.5	42.9	39.5
Ours	36.3	30.0	43.7	40.0

Table 8 Ablation study on IWSLT'14 En↔De dataset

Model	En→De	De→En
Transformer	28.51	35.08
BERT-NMT	30.45	36.11
Ours	30.69	36.50
Ours (random context)	30.09	36.10
Ours (fixed context)	30.29	36.12
Rand-Ctx-Enc	28.58	34.84
NMT-Ctx-Enc	28.71	34.88
WB-Ctx-Enc	28.81	34.50
MBE (Morishita et al., 2021)	28.32	34.33
Ours (no drop-net)	29.65	35.54
Ours (gating)	29.58	35.63

random sentence or a fixed sentence, which are denoted as "Ours (random context)" and "Ours (fixed context)" respectively.

(2) *How much benefit does BERT bring?* We first evaluate two simple baselines where the context encoder is initialized randomly (denoted as Rand-Ctx-Enc) and warm-started from the NMT model (denoted as NMT-Ctx-Enc). Then, We choose another baseline where the contextual encoder is the word embeddings (without positional embeddings) as in Kim et al. (2019) (denoted as WB-Ctx-Enc). Finally, we compare with the mini-batch embedding (MBE) for context-aware translation (Morishita et al., 2021), where each mini-batch is composed of sentences from the same document and thus contains contextual information.

(3) *Why do we use averaging and drop-net to integrate contextual information*, rather than the gating mechanism (Jean et al., 2017; Bawden et al., 2018; Kim et al., 2019)? To evaluate its effectiveness, we train a model without drop-net denoted as "Ours (no drop-net)", and another model with gating mechanism denoted as "Ours (gating)".

We conduct experiments on IWSLT'14 En↔De and use C-mode with one previous sentence (1 prev) as the contextual information. The results are shown in Table 8.

For the first question, we can see that both "Ours (random context)" and "Ours (fixed context)" perform worse than our method. This shows that BERT captures the relationship among different context sequences. For the second question, we can see that BERT is a strong contextual encoder and it outperforms all other contextual encoders. This shows that models pre-trained on a large corpus can extract the contextual information for context-aware NMT. Rand-Ctx-Enc performs even worse than Transformer, indicating that it is hard for the contextual encoder to learn contextual dependency from scratch. For the recently proposed MBE, its performance is far behind the pre-training based methods on

Table 9 Results of IWSLT'14 En→De with different pre-trained models

Context	BERT		RoBERTa		ELECTRA	
	0	2	0	2	0	2
BLEU	30.45	30.80	30.81	31.21	30.70	31.02

"Context" represents the number of sequences as contextual information

IWSLT En↔De translation, which shows the effectiveness of using pre-trained models in context-aware NMT. For the last question, we can see that averaging and drop-net significantly outperforms gating mechanism. A possible explanation is that BERT can produce high-quality contextual features, so that the gating mechanism is no longer necessary. Besides, drop-net is important for our model, which serves as a regularization.

4.5 Context-aware NMT with different pre-trained models

To verify the effectiveness of our proposed method, in addition to BERT (Devlin et al., 2019), we also use RoBERTa (Liu et al., 2019) and ELECTRA (Clark et al., 2020) for context-aware NMT. More specifically, we use the `roberta-base` and `google/electra-base-discriminator` from Huggingface. We conduct experiments on IWSLT'14 En→De translation. Given an input sentence, the contextual information we use is its next two sentences (i.e., the "2 next" in Table 2).

The results are in Table 9. Compared to the standard BERT, using more recent pre-trained models, RoBERTa and ELECTRA, brings more improvements. Our proposed method for context-aware NMT outperforms the corresponding baseline without contextual information on both RoBERTa and ELECTRA, which shows the effectiveness of our method.

5 Conclusions and future work

In this work, we study how to effectively leverage BERT in context-aware neural machine translation. We mainly study three approaches to extract and aggregate the contextual features, i.e., concatenation mode, flat mode and hierarchical mode. We find that in terms of accuracy, concatenation mode achieves the best results. We apply our discovery to five translation tasks in total and get promising improvements. Our work provides thorough analysis on using BERT for context-aware NMT and sets a strong baseline, which can help future work in this field.

For future work, there are many interesting directions. First, we will study how to leverage BERT into the target side of the translation model. Second, how to effectively leverage monolingual data for context-aware NMT is another topic to be explored. Third, Improving the inference efficiency is another interesting topic.

Author Contributions XW, YX, JZ, LW and SX got the basic ideas of the paper. XW and JZ conducted experiments. XW wrote the manuscript and all the authors revised the paper together. TQ supervised the project.

Funding None.

Availability of data and material All the data is publicly available. (1) IWSLT En↔De: https://github.com/pytorch/fairseq/blob/master/examples/translation/prepare-iwslt14.sh; (2) OpenSubtitle En↔Zh: https://opus.nlpl.eu/OpenSubtitles-v2018.php . Specifically, the preprocessed data is at https://github.com/bert-nmt/ctx-bert-nmt/tree/main/data/opensubtitle_enzh; (3) News Commentary v11: http://data.statmt.org/wmt16/translation-task/training-parallel-nc-v11.tgz; (4) OpenSubtitle 2018 En→Ru: https://github.com/lena-voita/good-translation-wrong-in-context. (5) WMT'19 En→De: http://www.statmt.org/wmt19/translation-task.html.

Code availability https://github.com/bert-nmt/ctx-bert-nmt.

Declarations

Conflict of interest The authors declared that they have no conflict of interest.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

Adhikari, A., Ram, A., Tang, R., & Lin, J. (2019). Docbert: Bert for document classification. arXiv preprint arXiv:1904.08398

Agrawal, R. R., Turchi, M., & Negri, M. (2018). Contextual handling in neural machine translation: Look behind, ahead and on both sides. In *21st annual conference of the European association for machine translation* (pp. 11–20).

Bao, G., Zhang, Y., Teng, Z., Chen, B., & Luo, W. (2021). G-transformer for document-level machine translation. arXiv preprint arXiv:2105.14761

Bawden, R., Sennrich, R., Birch, A., & Haddow, B. (2018). Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: Human language technologies, (long papers)* (Vol.1, pp. 1304–1313).

Clark, K., Luong, M.-T., Le, Q. V., & Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. In *International conference on learning representations*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Edunov, S., Ott, M., Auli, M., Grangier, D., & Ranzato, M. (2018). Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: Human language technologies, (long papers)* (Vol. 1, pp. 355–364). Association for Computational Linguistics.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580–587).

Hassan, H., Aue, A., Chen, C., Chowdhary, V., Clark, J., Federmann, C., Huang, X., Junczys-Dowmunt, M., Lewis, W., Li, M., & Liu, S. (2018). Achieving human parity on automatic Chinese to English news translation. arXiv preprint arXiv:1803.05567

Jean, S., Lauly, S., Firat, O., & Cho, K. (2017). Does neural machine translation benefit from larger context? arXiv preprint arXiv:1704.05135

Junczys-Dowmunt, M. (2019). Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the fourth conference on machine translation (shared task papers, Day 1)* (Vol. 2, pp. 225–233). Association for Computational Linguistics.

Kang, X., Zhao, Y., Zhang, J., & Zong, C. (2020). Dynamic context selection for document-level neural machine translation via reinforcement learning. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)* (pp. 2242–2254).

Kim, Y., Tran, D. T., & Ney, H. (2019). When and why is document-level context useful in neural machine translation? In *Proceedings of the fourth workshop on discourse in machine translation (DiscoMT 2019)* (pp. 24–34).

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 388–395). Association for Computational Linguistics.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 7871–7880).

Li, L., Jiang, X., & Liu, Q. (2019). Pretrained language models for document-level neural machine translation. arXiv preprint arXiv:1911.03110

Li, B., Liu, H., Wang, Z., Jiang, Y., Xiao, T., Zhu, J., Liu, T., & Li, C. (2020). Does multi-encoder help? A case study on context-aware neural machine translation. arXiv preprint arXiv:2005.03393

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., & Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics, 8,* 726–742.

Ma, S., Zhang, D., & Zhou, M. (2020). A simple and effective unified encoder for document-level machine translation. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 3505–3511).

Maruf, S., Martins, A. F. T., & Haffari, G. (2019a). 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 (long and short papers)* (Vol. 1, pp. 3092–3102). Association for Computational Linguistics.

Maruf, S., Saleh, F., & Haffari, G. (2019b). A survey on document-level machine translation: Methods and evaluation. arXiv preprint arXiv:1912.08494

Miculicich, L., Ram, D., Pappas, N., & Henderson, J. (2018). Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 2947–2954). Association for Computational Linguistics.

Morishita, M., Suzuki, J., Iwata, T., & Nagata, M. (2021). Context-aware neural machine translation with mini-batch embedding. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume* (pp. 2513–2521). Association for Computational Linguistics.

Müller, M., Gonzales, A. R., Voita, E., & Sennrich, R. (2018). A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation. In *Proceedings of the third conference on machine translation: research papers* (pp. 61–72).

Ng, N., Yee, K., Baevski, A., Ott, M., Auli, M., & Edunov, S. (2019). Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the fourth conference on machine translation (WMT19)* (pp. 314–319). Association for Computational Linguistics.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies (long papers)* (Vol. 1, pp. 2227–2237). Association for Computational Linguistics.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/languageunderstandingpaper.pdf

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog, 1*(8), 9.

Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (long papers)* (Vol. 1, pp. 1715–1725). Association for Computational Linguistics.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229

Tiedemann, J., & Scherrer, Y. (2017). Neural machine translation with extended context. arXiv preprint arXiv:1708.05943

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł, & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Voita, E., Sennrich, R., & Titov, I. (2019). When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 1198–1212). Association for Computational Linguistics.

Voita, E., Serdyukov, P., Sennrich, R., & Titov, I. (2018). Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th annual meeting of the association for computational linguistics (long papers)* (Vol. 1, pp. 1264–1274). Association for Computational Linguistics.

Wong, K., Maruf, S., & Haffari, G. (2020). Contextual neural machine translation improves translation of cataphoric pronouns. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 5971–5978).

Xiong, H., He, Z., Wu, H., & Wang, H. (2019a). Modeling coherence for discourse neural machine translation. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, pp. 7338–7345).

Xiong, H., Zhang, R., Zhang, C., He, Z., Wu, H., & Wang, H. (2019b). Dutongchuan: Context-aware translation model for simultaneous interpreting. arXiv preprint arXiv:1907.12984

Yang, J., Wang, M., Zhou, H., Zhao, C., Yu, Y., Zhang, W., & Li, L. (2019a). Towards making the most of bert in neural machine translation. arXiv preprint arXiv:1908.05672

Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., Li, M., & Lin, J. (2019b). End-to-end open-domain question answering with bertserini. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations)* (pp. 72–77).

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies* (pp. 1480–1489).

Yun, H., Hwang, Y., & Jung, K. (2020). Improving context-aware neural machine translation using self-attentive sentence embedding. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, pp. 9498–9506).

Zhang, J., Luan, H., Sun, M., Zhai, F., Xu, J., Zhang, M., & Liu, Y. (2018a). Improving the transformer translation model with document-level context. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 533–542). Association for Computational Linguistics.

Zhang, J., Luan, H., Sun, M., Zhai, F., Xu, J., Zhang, M., & Liu, Y. (2018b). Improving the transformer translation model with document-level context. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 533–542).

Zheng, Z., Yue, X., Huang, S., Chen, J., & Birch, A. (2020). Toward making the most of context in neural machine translation. In *IJCAI-PRICAI*.

Zhu, J., Xia, Y., Wu, L., He, D., Qin, T., Zhou, W., Li, H., & Liu, T. (2020). Incorporating bert into neural machine translation. In *International conference on learning representations*.