Title	Incorporating BERT into Document-Level Neural Machine Translation
Author(s)	郭，志宇
Citation	
Issue Date	2021-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/17145
Rights	
Description	Supervisor: Nguyen Minh Le，先端科学技術研究科，修士(情報科学)

Master's Thesis

Incorporating BERT into Document-Level Neural Machine Translation

1810409 GUO Zhiyu

Supervisor Professor Nguyen Minh Le

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
Information Science

February, 2021

Abstract

In recent years, we have witnessed the rapid development of deep learning technology, and the application of deep learning in the field of machine translation has continued to be deepening. Among them, the attention-based encoder-decoder Neural Machine Translation (NMT) framework (Bahdanau et al., 2014) surpassed the traditional statistical machine translation framework in performance significantly. Further, the Transformer (Vaswani et al., 2017) framework has improved the performance of neural machine translation to a new level.

Due to the limitation of training methods, these advanced frameworks consider one sentence as a whole in the process of translation. In the actual translation process, the text we use is often composed of multiple sentences. As the document has special characteristics, the translation of these sentence-level models is often lacking coherence and cohesiveness when translating documents.

Since late 2018, large-scale pre-trained representations such as BERT (Devlin et al., 2019) have been widely used in many natural language understanding tasks, including machine reading comprehension, text classification. The methods of incorporating BERT into document-level machine translation are still being explored. BERT is able to understand sentence relationship since one of the BERT pre-training task is the next sentence prediction task, the sentence relationship information is very important for document-level machine translation. Therefore, in our work, we leverage pre-trained BERT to improve the performance of document-level machine translation.

In this research, we propose a novel method to incorporate pre-trained BERT into document-level NMT. The BERT model performs as a context encoder to model the document-level contextual information. We concatenate the document-level context and the current sentence as the input for the BERT context encoder. The contextual-representation encoded by BERT is then integrated into both the encoder and the decoder of the Transformer NMT model using the multi-head attention mechanism. The attention mechanism can also deal with the case that BERT module and Transformer NMT module might use different word segmentation rules. Given the fact that translating different sentences may require a different amount of contextual information, we propose to use context gates to integrate the output of the multi-head attention mechanism.

The parameter size of our model is very huge, to save training time, we propose a two-step training strategy for our model. Firstly, we split the document-level training data into separate sentences, we train a sentence-level Transformer NMT model. After that, we use the sentence-level Transformer NMT model to initialize the parameter of the Transformer NMT module in our model, and we train the document-level NMT model with the parameter of the BERT module fixed.

We tested our model on English-German and Chinese-English datasets. The results showed huge improvements over the sentence-level Transformer model, and our proposed model outperformed several strong document-level NMT baselines. Especially, our model achieved new state-of-the-art performance on the English-German News Commentary dataset. The effectiveness of our model has been proved.

We tried to integrate the contextual representation encoded by BERT into a different part of the Transformer NMT model. The results showed integrating contextual representation into the encoder can achieve more improvements than integrating into the decoder. Integrate the contextual representation into both the encoder and the decoder of the NMT model can achieve the best results.

Regrading Li et al. (2020) argue that the context encoder in document-level NMT can not capture contextual information, we follow their experimental setting presenting three inputs for BERT context encoder. The results showed that the BERT context encoder in our model can capture contextual information to improve translation performance.

In future work, we would like to compress our model into a light version. Also, we would like to use more than one context sentences. Furthermore, we would like to test the performance of our model in some low-resource languages.

Keywords: *Deep Learning, Neural Machine Translation, Pre-trained Model, Document-level, Attention Mechanism, Context Gate*

Acknowledgement

I would first like to thank my supervisor, Professor NGUYEN LE MINH, he gave me an opportunity to study at JAIST, his expertise was invaluable in formulating the research issues and methodologies. His insightful feedback prompted me to think more acutely and raised my work to a higher level.

Also, I would like to thank Professor Tojo Satoshi for his constructive suggestions on my research proposal, and for showing me the interesting things about school life at JAIST.

I would like to thank Assistant Professor RACHARAK Teeradaj for his valuable guidance throughout my studies.

I would like to thank Associate Professor SHIRAI Kiyoaki for his supervision in my minor research, which gave me a chance to explore a new area in NLP.

Special thanks to all members of Nguyen's Laboratory. I spend a very happy time in Nguyen lab and learned a lot of things from all the members.

At last, I would like to thank my family for supporting me to study in Japan. You are always there for me.

Guo Zhiyu
January 2021

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we will make a brief introduction about the background of this research, the motivation which inspired us to do this work, and the objectives of this research.

1.1 Background

Machine translation (MT), which converts speech or text from one natural language to another natural language, is one of the most challenging and popular objectives of computers. Traditional MT mainly relies on statistical technology, so it is called statistical machine translation (SMT), it needs to elaborately craft features to extract implicit information from the bilingual sentence-pairs corpora (Brown et al., 1993). The hand-designed features are one of the reasons for their inflexibility. With the development of deep learning technology, Neural Machine Translation (NMT) systems were proposed (Sutskever et al., 2014; Bahdanau et al., 2014; Vaswani et al., 2017), which are mainly based on neural network architectures, have the potential to overcome the complicated feature engineering problem in SMT systems. NMT systems are based on the encoder-decoder architecture, given a source sentence as input, the encoder encodes it into a fixed-length embedding vector, according to the embedding vector, the decoder generates the translation results. NMT systems have even reached human parity on some resource-rich language pairs (Hassan et al., 2018). However, most models have adopted standard assumptions to translate each sentence independently, without taking advantages of the document-level contextual information in the translation process.

In actual translation tasks, the object of translation is often a language unit composed of multiple sentences, a complete document, or a discourse. Discourse as a size larger language units have their phenomena, Justice (1983) indicates discourse have the following 7 phenomena: cohesion, coherence, intentionality, informativity, acceptability, situational, intertextuality. Among those phenomena, cohesion and coherence are two basic phenomena that have an important impact on the entire text (Scherrer et al., 2019). Cohesion is the surface attribute of text, which refers to the way text units are linked together in morphology or grammar. Coherence refers to the potential meaning relationship between a text unit and its continuity.

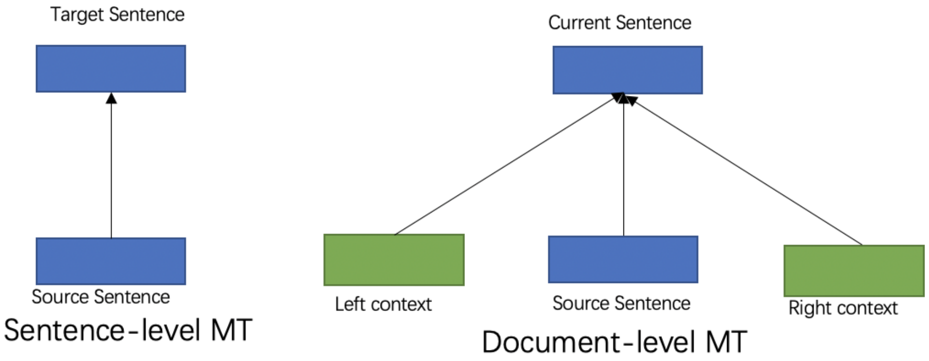


Figure 1.1: Differences between the sentence-level NMT task and the document-level NMT task. Given a source sentence, the sentence-level MT task generates a target sentence, while the document-level MT task generates a target sentence given the surrounding context. Every sentence in a document is thought to be relative with other sentences.

The two basic phenomena of the text have brought great challenges to the research of machine translation. Since the traditional machine translation framework only translates at the sentence level or even smaller granularity, the context of the sentence is not considered in the process of translating the text, which makes the translation result lacking cohesion and coherence. How to model inter-sentence information in the machine translation system to improve the performance of text translation results has always been a very important research topic.

As early as in the related research of statistical machine translation, there are some methods to improve the MT model using context information, such as maximum entropy-based phrase reordering model (Xiong et al., 2006), combining rich context information to select translation rules in the decoding process (He et al., 2008). With the development of NMT technique, various document-level NMT models, have been proposed to extract context information from the surrounding sentences and have achieved substantial improvements for generating consistent translations (Voita et al., 2018; Zhang et al., 2018; Werlen et al., 2018; Maruf et al., 2019b; Ma et al., 2020).

Large-scale pre-trained text representations like GPT-2 (Radford et al., 2018, 2019), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2019), have been widely used in many natural language understanding (NLU) tasks. Among them, BERT is one of the most effective representations that has inspired many other representations such as RoBERTa, ALBERT. It significantly boosts the performance of many NLU tasks, including question answering, text classification, etc (Devlin et al., 2019). The architecture of BERT is a Transformer-based encoder, how to incorporate BERT into natural language generation tasks such as NMT that need to use encoder-decoder architecture is a challenging problem. There are few recent works using BERT to improve the performance of NMT models (Xiong et al., 2019; Zhu et al., 2020; Weng et al., 2019; Chen et al., 2020).

1.2 Motivation

BERT is pre-trained using the Masked Language Model (MLM) task and the Next Sentence Prediction (NSP) task. For MLM task, the BERT is forced to predict the masked part of the sentence. By MLM task, a bidirectional pre-trained model is obtained. In the NSP task, the model should predict whether the two sentences are adjacent. Intuitively, a pre-training task of BERT is the binarized NSP task, a natural assumption is that the NSP task has enabled the BERT to understand the relationship information between two sentences, the relationship information is helpful to model the context information for document-level machine translation. Inspired by this, we use BERT to encode the contextual representation to improve the translation quality of document-level NMT.

Transformer model (Vaswani et al., 2017) utilizes attention mechanism to build multi-head attention structure and achieve important improvement in the NMT field. Zhang et al. (2018) use multi-head self-attention mechanism to achieve the document-level context representation, and then use multi-head attention to incorporate document-level representation into the encoder and the decoder of the NMT model. Inspired by this, we leverage multi-head attention mechanism to incorporate BERT context representation into the NMT model. The multi-head attention can also solve the problem that the NMT module and the BERT module may use distinct word segmentation rules.

In addition, based on our experience, different source sentences require a different amount of context information for translation. Inspired by context gate in Werlen et al. (2018); Zhang et al. (2018), we propose to leverage context gate to combine the output of BERT context attention and self-attention.

We propose to extend the Transformer model to take advantage of BERT document-level contextual representation. We take the BERT as a context encoder to achieve document-level representation, then the representation is integrated into both the encoder and the decoder of the Transformer NMT model. We leverage the multi-head attention and the context gate to control how each layer interacts with BERT context representations adaptively.

Li et al. (2020) claims that the improvements of the multi-encoder document-level NMT approach is not from the leverage of contextual information, it is from noise generated by the context encoder, which can provide additional supervision signals for training the sentence-level NMT models. Because our proposed document-level NMT model is a multi-encoder based model, we need to show whether our model can really capture the context information.

1.3 Objectives

In this research, the main goal is to incorporate pre-trained BERT into the document-level NMT model. Our main works are as following:

- First, we would like to implement a strong sentence-level NMT model as the baseline model, which is the widely used Tansformer (Vaswani et al., 2017) model based on attention mechanisms. Also, we implement several document-level NMT models. The goal is to build some baseline model for document-level NMT, and to get a better understanding of leveraging document-level contextual information to improve machine translation performance.

- We implement our document-level NMT model which uses the pre-trained BERT model as context encoder, the document-level contextual information encoded by BERT is integrated into the Transformer NMT model using multi-head attention mechanism and context gate. We propose a two-step training method for our model. We test the translation results of our model on English-German and Chinese-English language pair datasets, and we compare the results with the state-of-the-art document-level NMT models.

- We take ablation study about our model. We try to integrate document-level contextual information into different parts of Transformer NMT model, in this way, we investigate the effectiveness of three integration ways. Also, we try to present three kinds of input for the BERT context encoder to investigate whether the BERT context encoder can really capture the contextual information to improve translation performance.

- Finally, we would like to have an analysis of the limitation of this work and propose several potential further research directions.

1.4 Thesis Outline

The specific content of this thesis is as follows:

- **Chapter 2: Literature Review** We introduce an overview of the development of NMT technology, some state-of-the-art model in document-level MT, and some method for making use of BERT in the NMT model.
- **Chapter 3: Methodology** We introduce our document-level NMT model that using pre-trained BERT as context encoder. We also show the training strategy for our model.
- **Chapter 4: Experimentation** We introduce the datasets , the implementation details, and the evaluation metrics we.
- **Chapter 5: Evaluation** We compare the performance of our model with several state-of-the-art document-level NMT models, and we take the ablation study for our approach.
- **Chapter 6: Conclusion** We give the conclusion of this work and some directions for future research.

Chapter 2

Literature Review

In this chapter, we first give a brief review of the methods that are widely used in NMT and then we will introduce the related work in large-scale pre-trained representation and document-level NMT.

2.1 Sentence-level Neural Machine Translation

With the rise of neural network methods and their application in machine translation (Sutskever et al., 2014; Bahdanau et al., 2014; Vaswani et al., 2017), the field of Neural Machine Translation (NMT) has developed by leaps and bounds, opening a new era of machine translation for research and industry purposes. The main advantage of NMT over its predecessor is that it has an end-to-end model whose parameters can be jointly optimized for training objectives.

Given a source sentence, the goal of sentence-level NMT is to search the most probable target sequence, that is:

$$\hat{y} = \arg\max_{y} P(y \mid x) \tag{2.1}$$

The neural networks are used to model conditional probability $P(y \mid x)$, where $\mathrm{x} = (x_1, ..., x_M)$ is the source sentence and $y = (y_1, ..., y_M)$ is the target sentence. Given the source sentence x, the conditional probability of a target sentence x is decomposed as:

$$P_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{n=1}^{N} P_{\boldsymbol{\theta}}\left(y_n \mid \boldsymbol{y}_{<n}, \boldsymbol{x}\right) \tag{2.2}$$

θ denote the learnable parameters of the neural network, y_n is the current target word that needs to be generated, the previously generated words is $y_{<n}$.

2.2 Sequence-to-Sequence Model

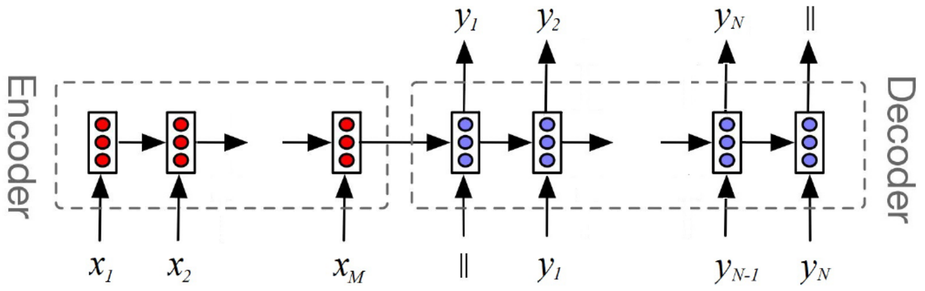


Figure 2.1: The overview of an encoder-decoder model for sentence-level NMT. The figure is from Maruf et al. (2019a).

The Sequence-to-Sequence model is widely used in NMT, and it is an encoder-decoder based architecture as shown in Figure 2.1. The source sentence is the input for the encoder, it is then encoded into a real-valued representation. Given the previously computed source sentence representation, then the decoder generates the target word at a time. Sutskever et al. (2014) use a fixed representation of the source sentence to produce the target sentence. Bahdanau et al. (2014) propose a novel encoder-decoder architecture that can generate a dynamic context representation, they have leveraged attention mechanism in their model. These architectures can be categorised as Recurrent Neural Networks (RNNs) architecture which exhibit temporal dynamic behavior over time using recurrent connections, and are significantly suitable for modeling sequence representation. However, the main disadvantage of RNN architecture is that it lacks parallelization ability during training process, when processing long sentences, this becomes a bottleneck. More recently, a new seq2seq model architecture, the Transformer (Vaswani et al., 2017), was introduced which dispense with the recurrence entirely, is based solely on attention mechanisms. It has achieved state-of-the-art performance in many language pairs.

2.3 Recurrent Neural Networks

The most widely used RNN architecture in NMT is (Bahdanau et al., 2014)'s attention-based model. The encoder of this model is a bidirectional (forward and backward) RNN, and its hidden state represents each word of the source sentence. The forward and backward RNNs runs on the source sentence along the left-to-right and right-to-left directions, and then each word in the source sentence is

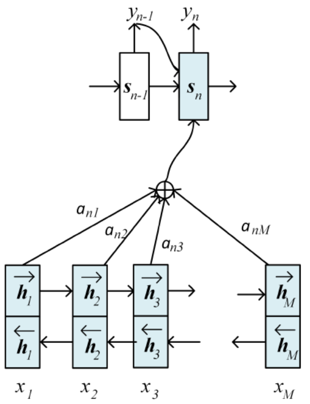


Figure 2.2: Attentional RNN-based architecture (Bahdanau et al., 2014)

represented by the corresponding two-way hidden state concatenation (as shown in Figure 2.2). These representations not only capture information about the corresponding word, but also information about other words in the sentence.

The attention mechanism is an indispensable part of the RNN-based NMT architecture. This enables the decoder to dynamically concentrates on the relevant parts of the source sentence in each step of generating the target sentence. The dynamic context vector c_n (also referred to as the attentional vector) is computed as a weighted linear combination of the hidden states produced by the bidirectional RNNs in the encoder, where the weights (α in Figure 2.2) can be regarded as the alignment probability between a target symbol at position n and a source symbol at position m.

2.4 Transformer

There are two limitations in the RNN based NMT models. The first is when processing each input token, the model has to wait until all previous input tokens have been processed, when turns to long sequences, this is a huge bottleneck. The second is about learning long-range dependencies among the tokens within a sequence. As the distance increases, so does the number of operands required to correlate signals from two arbitrary input or output positions, it is hard to learn complex dependencies between distant positions. To solve the above challenges, Transformer model Vaswani et al. (2017) was proposed. It totally discards the recurrent or convolutional structure, and only utilize attention mechanism for sequence transduction.

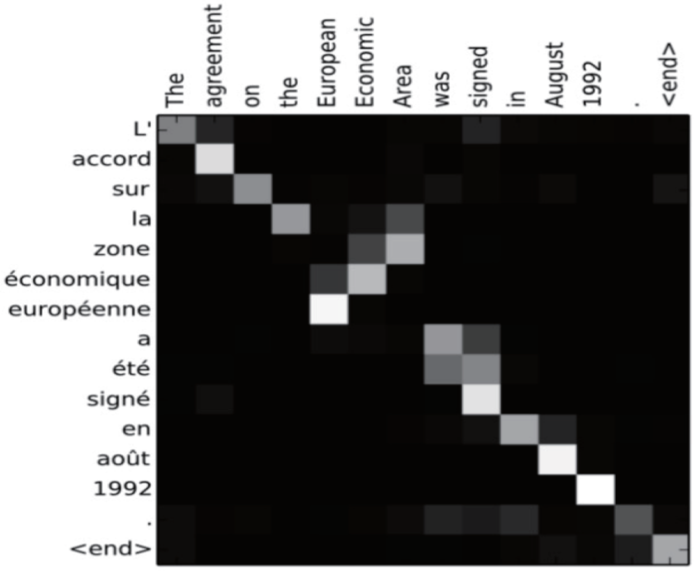


Figure 2.3: The visual alignment example of the attention mechanism (Bahdanau et al., 2014)

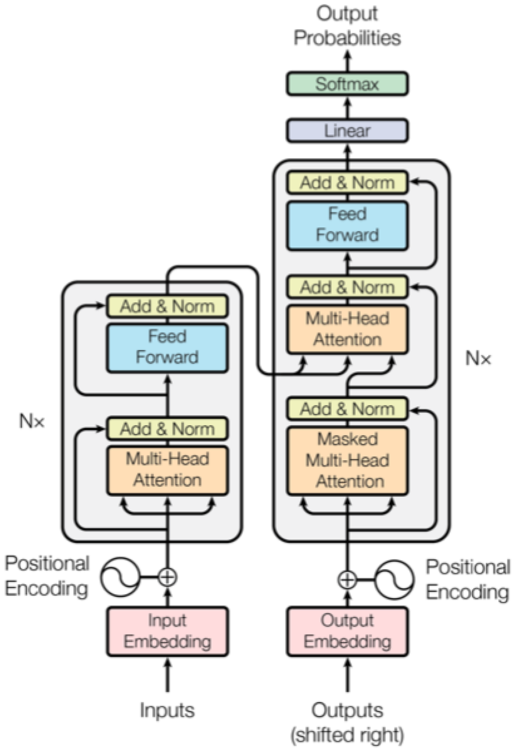


Figure 2.4: The model overview of the Transformer (Vaswani et al., 2017)

2.4.1 Encoder and Decoder

As shown in Figure 2.4, in the encoder, there are N identical layers. Each layer is composed of two sub-layers. A multi-head self-attention mechanism is the first sub-layer, and a position-wise fully connected feed-forward network is the second sub-layer. For the decoder, there are also N identical layers. Unlike the encoder which has sub-layers in each layer, there is a third sub-layer in the decoder that calculates multi-head attention on the output from the encoder.

2.4.2 Scaled Dot-Product Attention

There is a special attention called ”Scaled Dot-Product Attention” in Transformer. The input is composed of queries and keys with the dimension d_k, and values with the dimension d_v. In practice, we calculate the attention function on a set of queries at the same time, then pack them together to form a matrix Q. The keys and values are also packed together to form the matrix K and V. We calculate the output matrix as:

$$\text{Attention}\,(Q, K, V) = \operatorname{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{2.3}$$

2.4.3 Multi-Head Attention

It is beneficial to learn different linear projection linear projections h times in the dimensions of d_k, d_k and d_v for query, key and value. Then, on each version of these predictions of query, key, and value, we execute the attention function at the same time to produce the output value with the dimension of d_v. Finally, we concatenate them and project again to achieve the final value. The multi-head attention enables the model to jointly attend to information from different representation subspaces at different positions. The calculation is:

$$\begin{aligned}&\text{MultiHead (Q, K, V)= Concat (head }_1, \ldots, \text{ head}_{\text{h}})\, W^O\\ &\text{where head}_i = \text{ Attention } \left(QW_i^Q, KW_i^K, VW_i^V\right) \qquad (2.4)\end{aligned}$$

Where W_i^Q, W_i^K, W_i^V and W^O are the projections parameter matrices.

2.4.4 Positional encoding

Since there is no convolution or no recurrence in their model, to make use of the sequence order , some information about the relative or absolute position of the tokens in the sequence must be injected to the model. For this purpose, the ”positional encodings” are added to the input embeddings at the bottoms of the encoder and decoder stacks. The position encoding has the same dimension as the embeddings and is calculated as:

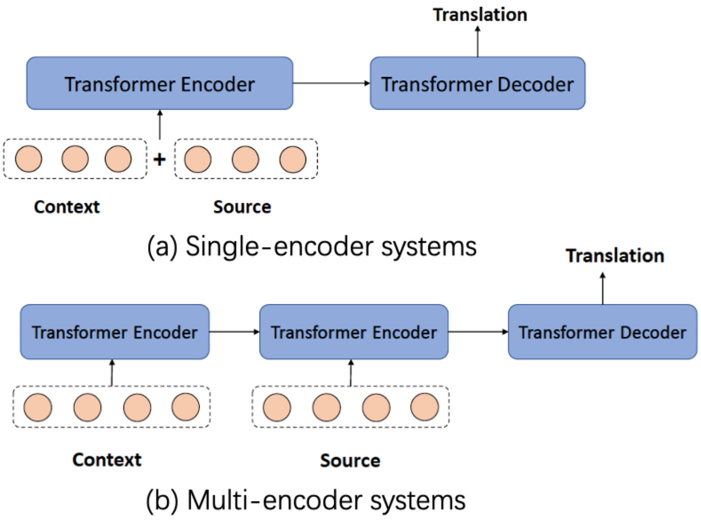
$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{\text{model}}}\right) \tag{2.5}$$

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{\text{model}}}\right) \tag{2.6}$$

where pos is the position and i is the dimension.

2.5 Related work in Document-level NMT

Most of the current document-level NMT systems that model the inter-dependencies among the sentences in a document can be broadly divided into two classes, single-encoder systems, and multi-encoder systems. This is determined by whether the model has leveraged an additional encoder to model the inter-dependencies among the sentences in the document. The overview of the single-encoder systems and the multi-encoder systems are shown in Figure 2.5.



(a) Single-encoder systems

(b) Multi-encoder systems

Figure 2.5: The overview of the single-encoder systems and the multi-encoder systems for document-level NMT. (Ma et al., 2020)

2.5.1 Single-encoder Systems

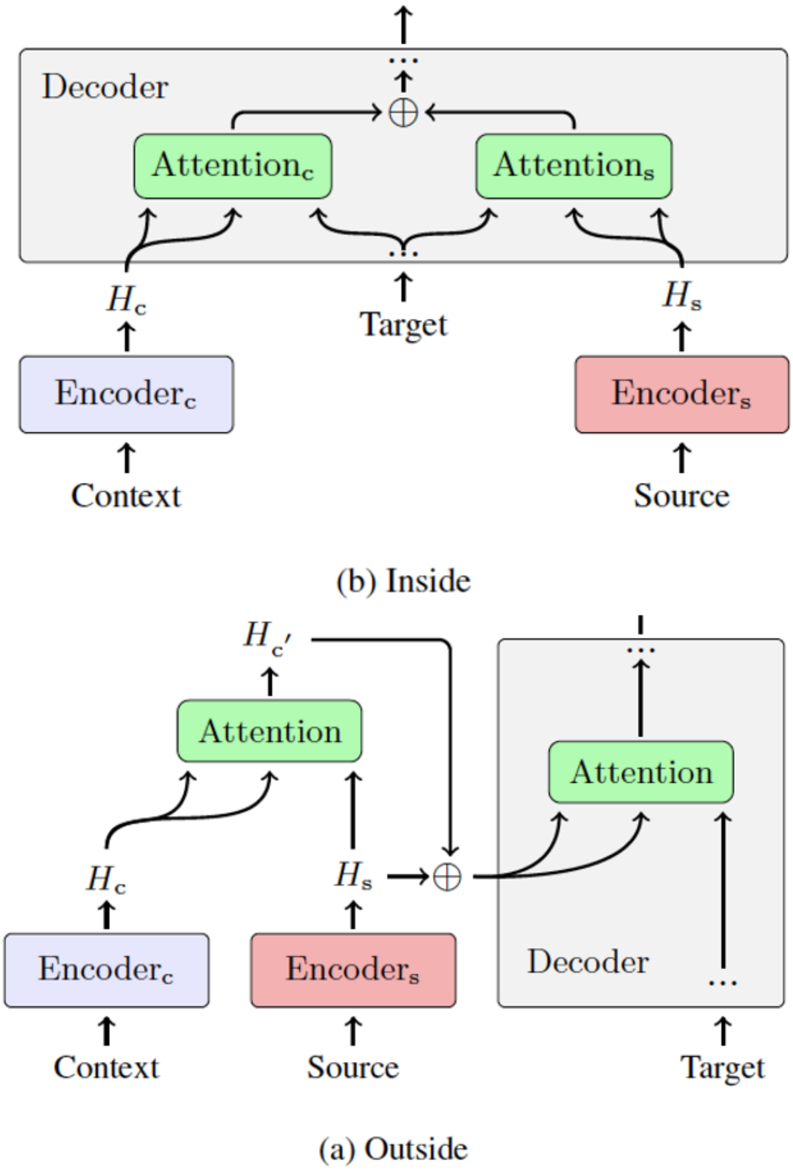
Based on the RNN NMT model, Tiedemann and Scherrer (2017) tried to extend the translation units in two ways, the first way is adding one previous sentence into the source sentence, the second way is adding one context sentence into both the source sentence and the target sentence. Ma et al. (2020) concatenate the context and source embeddings as inputs for the NMT model. In addition to the standard word embedding matrix, they proposed a novel segmented embedding matrix. After achieving the concatenation of source and context sentences, the word and segment embeddings are then added and put into the first layer of the encoder. The rest layers just use the current sentence embedding as input. In this way, the higher layer can focus more on the current sentence, the context sentence can be treated as supplemental information. Since their model is identical to the recent pre-training models like BERT, they have leveraged BERT to further improve the translation quality of their proposed approach.

2.5.2 Multi-encoder Systems

From the state-of-the-art Transformer architecture, Voita et al. (2018) changed the encoder into a context-aware encoder. The proposed context-aware encoder is composed of a context encoder and a source sentence encoder, they share the first $L - 1$ layers' parameter. The previous source sentence is the input for the context encoder. The output for the context encoder is attended to the L^{th} layer of the source sentence encoder using multi-head attention mechanism. Then the attention output is integrated into the current sentence encoder output with a gate. The output from the final document-level encoder layer is then integrated into the decoder. Similar to Voita et al. (2018), Zhang et al. (2018) also leverage a context-aware encoder for the Transformer model. Different from Voita et al. (2018) training the context-aware model from scratch, they take the pre-trained sentence-level Transformer to initialize the parameter of the context-aware model. In the second training stage, the sentence-level NMT module parameters are fixed, they just learn the document-level parameters.

Werlen et al. (2018) proposed the hierarchical attention network (HAN) based architecture to achieve the contextual representation in a structured manner make use of both the sentence-level and the word-level abstractions. Maruf et al. (2019b) propose a novel method based on sparse attention to hierarchical attention for document-level NMT. A Query-guided Capsule Network (QCN) (Yang et al., 2019a) uses an improved dynamic routing algorithm for improving context modeling for the document-level NMT model.

Most of the multi-encoder document-level NMT architecture (Zhang et al., 2018; Voita et al., 2018), can be classified as two categories as below.



(b) Inside

(a) Outside

Figure 2.6: The overview of two widely used multi-encoder system architectures. In the inside architecture, H_s and H_c is the key and value, and target is the query. In the outside architecture, H_s is the query, and H_c is the key and value, and target is the query. (Li et al., 2020)

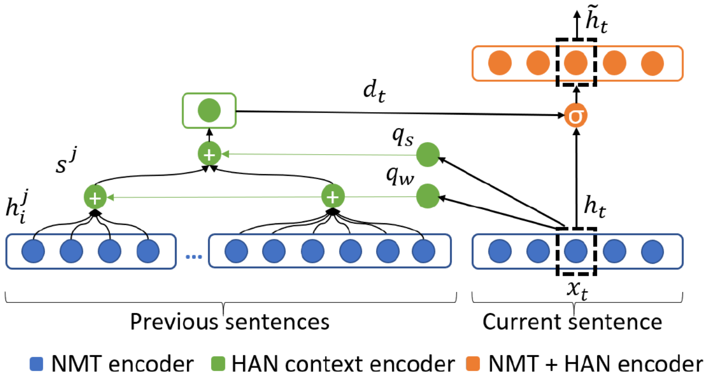


Figure 2.7: The HAN based document-level NMT model architecture (Werlen et al., 2018)

- **Inside integration** As shown in Figure 2.6(a), firstly, the decoder attend to the current sentence encoder and context encoder output using two attention network respectively. After that, the fusion vector can be obtained using the gating mechanism in the decoder.
- **Outside integration** As shown in Figure 2.6(b), a new representation can be achieved by transforming the context representation and current sentence representation using an attention network. After that, the source sentence representation and the new representation are fused using a gate mechanism. Finally, the fused representation is integrated into the decoder.

More recently, Li et al. (2020) investigated how much the multi-encoder document-level NMT model such as Voita et al. (2018); Zhang et al. (2018) can benefit from leveraging the context. After conducting experiments on the small-scale dataset, they found that the additional context encoder performs as a noise generator that can provide richer training signals to the NMT model. Comparable improvements were achieved when the model was trained with the wrong context or if Gaussian noise was added to the encoder output. Therefore, they argued for showing the multi-encoder document-level NMT system can really capture context information to improve translation performance, especially when taking experiment on small scale datasets.

2.6 Flat-Transformer

Most of the existing document-level NMT systems (Zhang et al., 2018; Werlen et al., 2018; Maruf et al., 2019b) are multi-encoder based systems. Different from

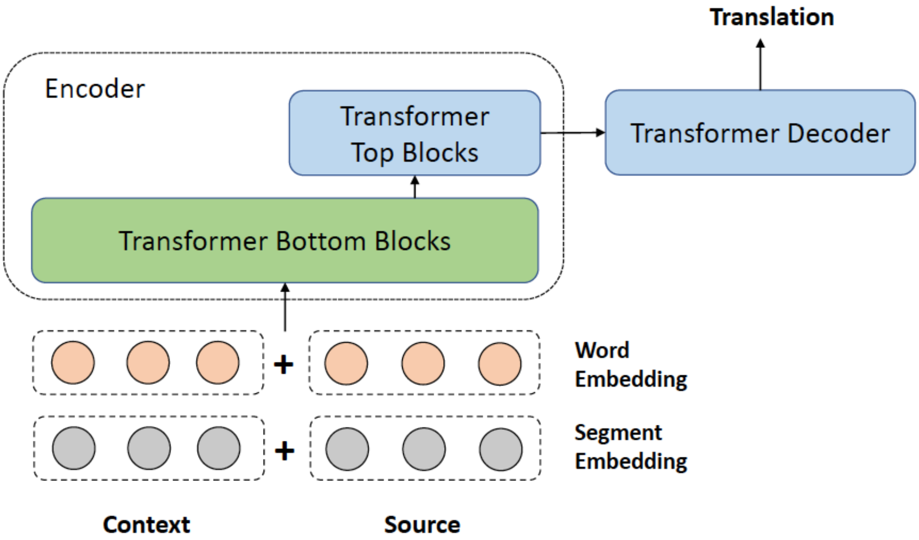


Figure 2.8: The architecture of Flat-Transformer model. (Ma et al., 2020)

the multi-encoder systems, the input of the single-encoder systems is the concatenation of contexts and current source sentences. Therefore, when extracting the context features, it can take advantages of the interaction between the contexts and the source sentences, while the multi-encoder systems fails to exploit this information. In addition, the architecture of the single-encoder system is identical to the recent pre-training models such as BERT. However, the previous single-encoder document-level NMT systems suffers from two problems. Firstly, the contexts and the source sentences are modeled equally, which is contrary to the fact that the current source sentences play more important roles in translation. Secondly, if the input sequences are very long, the attention is distracted.

2.6.1 Segment Embedding

The flat structure of the single-encoder NMT systems can not distinguish the source sentences and the context sentences. To solve this problem, they present the segment embedding to distinguish these two types of inputs. Formally, given the source input of the current sentence $\mathbf{x}$ and the surrounding context $\mathbf{c}$, we transform them into the word embedding and the segment embedding. After that, we concatenate them into a single input:

$$\mathbf{e} = [E(\mathbf{c}) : E(\mathbf{x})] \tag{2.7}$$

$$\mathbf{s} = [S(\mathbf{c}) : S(\mathbf{x})] \tag{2.8}$$

where $E(.)$ denote the word embedding matrix, and $S(.)$ denote the segment embedding matrix, [:] denotes the concatenation operation. Finally, we sum e and s as the input of the proposed encoder.

2.6.2 Unified Flat Encoder

The input sequences of Flat-Transformer are much longer than the sentence-level Transformer due to the document-level context, which makes it more challenging. Firstly, as the memory consumption and the computation cost increase significantly, then it become difficult to enlarge the model size, which make it hard to apply the big pre-training model. Secondly, the attention is distracted, and its weights drop significantly after the normalization function.

To solve those problems, they propose a unified flat encoder. As shown in Figure 2.8, at the bottom of the Transformer encoder blocks, the concatenated sequence of the context sentences and the current sentence is feed into the self-attention and the feed forward layer:

$$h_1 = \text{ Transformer } (\mathbf{e} + \mathbf{s}; \theta) \tag{2.9}$$

where θ denote the parameter in the Transformer blocks. In the top of the encoder , every self-attention and feed forward layer only focus on the current sentences:

$$h_2 = \mathrm{Transformer}\left(h_1[a : b]; \theta\right) \tag{2.10}$$

where a and b are the start and end positions of the source sentences in the input sequence. In this way, the attention can be more focused more on the current sentence, and the contexts is used as the supplementary semantics of the current sentence. The total number of bottom and top blocks is equal to the number of standard Transformer encoder, so the parameters do not exceed the number of standard Transformer.

2.7 BERT: Bidirectional Encoder Representations from Transformers

The BERT model's architecture is a multi-layer Transformer encoder. BERT aims to pre-train deep bidirectional representations in the unlabeled text by jointly conditioning on both left and right context in all layers. Therefore, it is only necessary to add an output layer to fine-tune the pre-trained BERT to create the state-of-the-art model suitable for many natural language understanding tasks, such as machine reading comprehension and text classification, without significant task-specific architecture modifications.

2.7.1 Input/Output Representations

BERT use WordPiece embedding (Wu et al., 2016) with the vocabulary size of 30000. The special classification token ”[CLS]” is the first token of each input sequence. In the text classification task, the last layer’s hidden state which corresponds to the ”[CLS]” token is taken as the whole sequence representation. In the situation where the input is a sentence pair, we combine them together into a single sequence. In order to distinguish sentences, we use special token ”[SEP]” to separate them, also, to indicate whether one token belongs to sentence A or sentence B,we add a learned embedding to every token. For a given token, after adding the corresponding token, segment, and position embeddings, the input representation for BERT is constructed. Figure 2.9 is the illustration of BERT input composition.

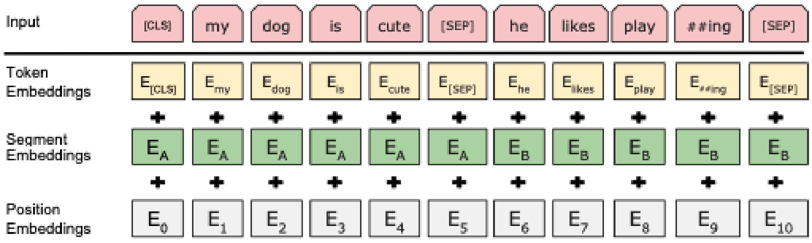


Figure 2.9: The input of BERT (Devlin et al., 2019)

2.7.2 Pre-training BERT

The pre-training process of BERT is consisting of two tasks: the Masked Language Model (MLM) task and Next Sentence Prediction (NSP) task.

Task 1: Masked LM In the MLM task, we randomly maske some percentage of the input tokens, then the BERT needs to predict those masked parts. The last layer’s hidden states which corresponds to the masked parts are fed into an output softmax over the vocabulary. We randomly mask 15% of all WordPiece tokens in each sequence. In this way, we can achieve a bidirectional pre-trained model, however, there is no ”[MASK]” token in the fine-tuning process, this lead to a gap between pre-training and fine-tuning. To solve this problem, we do not always use the ”[MASK]” token to replace the masked words. The pre-training data generator randomly selects 15% of the token positions for predicting. When the i-th token is chosen, the i-th token is replaced with (1) the ”[MASK]” token with the probability of 80% (2)the unchanged i-th token with the probability of 10%. (3) a random token with the probability of 10%.

Task 2: Next Sentence Prediction (NSP) There are a lot of significant downstream tasks such as Question Answering (QA) and Natural Language Inference (NLI) need to understand the relationship information between two sentences, this information can not be directly captured by language modeling. For the purpose of training a model that can understand the relationship between two sentences, they designed a binary predictive pre-training named the next sentence prediction, which can be easily generated from any monolingual corpus. For each pre-training example, when selecting the sentences pair A and B, B is the actual next sentence that follows A with the probability of 50%, B is a randomly selected sentence from the corpus with the probability of 50%.

Pre-training Data They use the English Wikipedia (2,500M words) and the BooksCorpus (800M words) (Zhu et al., 2015) for the pre-training corpus. For Wikipedia, they ignore lists, tables, and headers, and just extract the text passages. Rather than a shuffled sentence-level corpus, it is important to use a document-level corpus to extract long contiguous sequences.

2.8 Application of BERT in NMT

Given the huge success of BERT in many NLU tasks, it is natural to investigate the ways of incorporating BERT into NMT. BERT has only a Transformer encoder, it is initially designed for natural language understanding tasks. For NLU tasks, such as neural machine translation, the model needs to use encoder-decoder based architecture, how to apply BERT into these tasks is a challenging problem. There are some recent works on applying BERT to NMT.

Lample and Conneau (2019) use a multilingual pre-trained BERT model to initialize the entire encoder and decoder and achieved huge improvements on supervised MT and unsupervised MT tasks. MASS (Masked Sequence-to-Sequence Pre-Training) (Song et al., 2019) leverages Sequence-to-Sequence MLM to jointly pre-train both the encoder and decoder. Their method can outperform the BERT-like pre-training Lample and Conneau (2019) both on supervised MT and unsupervised MT. Yang et al. (2019b); Weng et al. (2019); Chen et al. (2020) leverage knowledge distillation to acquire knowledge from BERT to NMT. Li et al. (2019); Ma et al. (2020) use BERT to initialize parameters of document-level NMT model encoder. BERT-fused model (Zhu et al., 2020) exploits the representation from BERT by integrating it into all layers of the Transformer model.

2.9 BERT-fused model

Zhu et al. (2020) propose the BERT-fused model. At first, they tried two strategies that leverage BERT for NMT:

- The downstream models are initialized using the pre-trained BERT, then fine-tune the models.
- The downstream models use BERT as context-aware embeddings.

In the first strategy, they use a pre-trained BERT model to initialize the encoder of an NMT model, then finetune the NMT model using the machine translation datasets. They could not find huge improvement. In the second strategy, they find that this strategy can outperform the first one. This motivates them to propose the BERT-fused model.

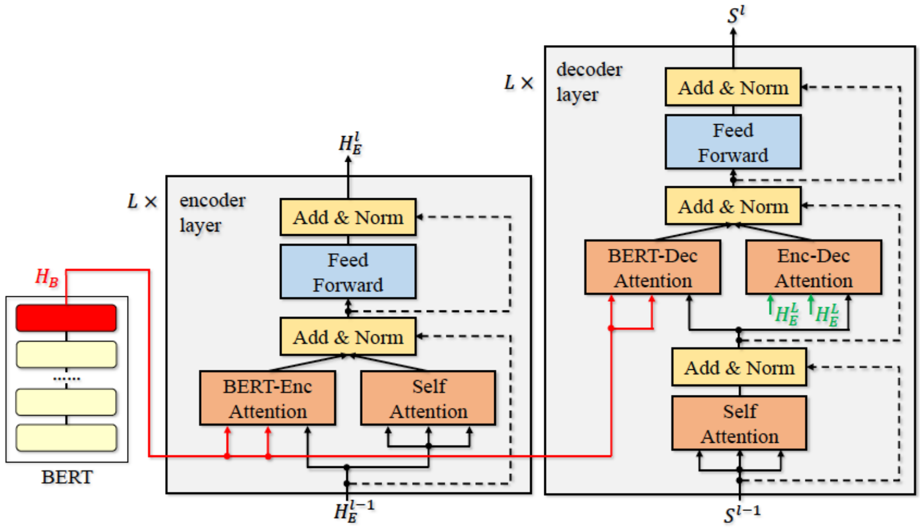


Figure 2.10: The overview of BERT-fused model. (Zhu et al., 2020)

2.9.1 Architecture

An illustration of BERT-fused model is shown in Figure 2.10. H_B denote the output from the last layer of BERT model. H_E^L denote the output of the last layer from BERT and encoder.

Step-1: Firstly, for any input sequence x, it is encoded by BERT into the representation $H_B = BERT(x)$.

Step-2: In the $l - th$ layer, we have

$$\mathbf{A}^l = \frac{1}{2}(\text{MultiHead}\left(H_E^{l-1}, H_E^{l-1}, H_E^{l-1}\right) + \text{MultiHead}\left(H_E^{l-1}, H_B, H_B\right)) \quad (2.11)$$

Where $MultiHead$ denote the multi-head attention mechanism in Vaswani et al. (2017).The output of the l-th layer is:

$$H_E^l = FFN(\mathbf{A}^l) \quad (2.12)$$

Where $FFN(.)$ denote a position-wise fully connected feed-forward neural network.

Step-3: For the decoder of the Transformer NMT model, at the l-th layer, we have

$$\mathbf{B}^l = \text{MultiHead}\left(S^{l-1}, S^{l-1}, S^{l-1}\right) \quad (2.13)$$

$$\mathbf{C}^l = \frac{1}{2}(\text{MultiHead}\left(B^l, H_B, H_B\right) + \text{MultiHead}\left(B^l, H_E^L, H_E^L\right)) \quad (2.14)$$

$$S^l = FFN(\mathbf{C}^l) \quad (2.15)$$

In their NMT system, the output of BERT is used as an extra sequence representation, and they utilize the multi-head attention model to merge it into the Transformer NMT model. This is a general method that utilizes the pre-training model even though the NMT model and the BERT are using different tokenization method.

2.9.2 Drop-net Trick

They propose a drop-net trick to make full use of the representations output by the Transformer NMT encoder and BERT encoder. The drop-net will effect Eqn.(2.7) and Eqn.(2.10). The drop-net rate is denoted as $p_{\text{net}} \in [0, 1]$. For any layer l, at each training iteration, a random variable U^l is uniformly sampled from $[0, 1]$, then all the $\mathbf{A}^l$ in Eqn.(2.7) are computed as follow:

$$\mathbf{A}^l = \mathbb{I}\left(U^l < \frac{p_{\text{net}}}{2}\right) \cdot \text{MultiHead}\left(H_E^{l-1}, H_E^{l-1}, H_E^{l-1}\right)$$

$$+\text{I}\left(U^l > 1 - \tfrac{p_{\text{net}}}{2}\right) \cdot \text{MultiHead}\left(H_E^{l-1}, H_B, H_B\right))$$

$$+\text{I}\left(\tfrac{p_{\text{net}}}{2} \leq U^l \leq 1 - \tfrac{p_{\text{net}}}{2}\right) \cdot \tfrac{1}{2}\left(\text{MultiHead}\left(H_E^{l-1}, H_E^{l-1}, H_E^{l-1}\right) + \text{MultiHead}\left(H_E^{l-1}, H_B, H_B\right)\right)$$

(2.16)

where $\mathbb{I}(.)$ is the indicator function. For any layer, in $p_{\text{net}}/2$ of the time, either the self-attention or the BERT-encoder attention is used only; in $1 - p_{\text{net}}$ of the time, both the two attention outputs are used.

Similarly, with the drop-net trick, when training of the decoder, we have

$$\mathbf{C}^l = \mathbb{I}\left(U^l < \frac{p_{\text{net}}}{2}\right) \cdot \text{MultiHead}\left(\boldsymbol{B}^l, \boldsymbol{H}_B, \boldsymbol{H}_B\right)$$

$$+\mathrm{I}\left(U^l > 1 - \tfrac{p_{\text{net}}}{2}\right) \cdot \text{MultiHead}\left(\boldsymbol{B}^l, \boldsymbol{H}_E^L, \boldsymbol{H}_E^L\right)$$

$$+\,\mathrm{I}\left(\tfrac{p_{\text{net}}}{2} \le U^l \le 1 - \tfrac{p_{\text{net}}}{2}\right) \cdot \tfrac{1}{2}\left(\text{MultiHead}\left(\boldsymbol{B}^l, \boldsymbol{H}_E^L, \boldsymbol{H}_E^L\right) + \text{MultiHead}\left(\boldsymbol{B}^l, \boldsymbol{H}_B, \boldsymbol{H}_B\right)\right)$$

(2.17)

Using drop-net to train BERT-fused model can prevent the model from overfitting. In the inference process, it is calculated as Eqn.(2.7) and Eqn.(2.10).

2.9.3 Differences with our work

BERT-fused model can also be extended to document-level NMT, but our work is different in the modeling and experimental part. While Zhu et al. (2020) are mainly focusing on improving sentence-level machine translation performance, they proposed a drop-net trick to combine the output of BERT encoder and the standard Transformer encoder, our proposed context gate combination can better leverage document-level context information since it is more correspond to the fact that different source sentences require a different amount of context information for translation. Also, the training process of BERT-fused model is different with our work. While Zhu et al. (2020) train the document-level NMT model from scratch, our work propose a two-step training process that can reduce the training time significantly.

2.10 Document-level NMT using large context and BERT

Since the performance of document-level NMT model degrades on larger contexts, most of the previous work in document-level NMT only take advantages of limited context. Li et al. (2019) try make use of large contexts. Their model is based on the Transformer model. They propose approaches to narrow the performance gap between models using contexts of different lengths. Their contributions are as follow:

- They utilize the pre-trained language models to initialize the parameters of the NMT model encoder. Unlike the previous pre-training techniques using large-scale sentence-level parallel corpora, the pre-trained language models are trained on monoligual documents.
- They introduce approached of manipulating context representation integration to control the impact of large contexts.
- They proposed multitasking training and added additional tasks to encoders to regularize their model and work with deeper encoders to further improve their system.

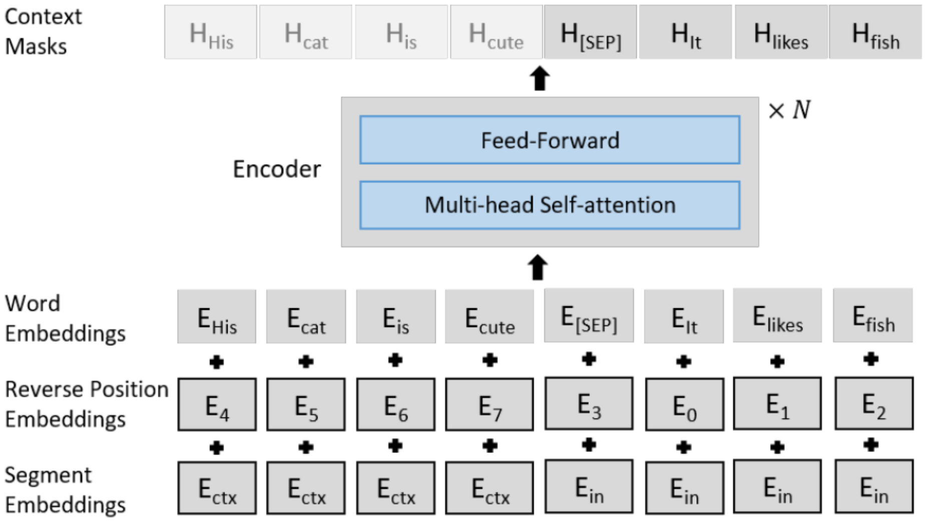


Figure 2.11: The architecture of document-level NMT using BERT to initialize the encoder. (Li et al., 2019)

The architecture of their proposed model is in Figure 2.11. Compared with the origi- nal Transformer NMT model, there are the following differences:

- They use segment embeddings to distinguish the contexts and the current sentence in the input.
- They propose to use reverse position embedding as an alternative to the original sequential position embedding.

- In the decoding process, they use context masks to prevent attention weights on the contexts.

Since they concatenate the contexts and the current source sentence together as the input of their NMT model input, there is only one encoder in their model, their model is a single-encoder document-level NMT model.

Chapter 3

Methodology

In this chapter, we introduce our proposed model that incorporates pre-trained BERT into document-level NMT. A detailed description of each part of the model will be given in the sub-sections.

3.1 Problem Statement

Formally, we denote $\mathbf{X} = x_1, x_2, \ldots, x_N$ as a source-language document with N source sentences. The corresponding target-language document is denoted by $\mathbf{Y} = y_1, y_2, \ldots, y_M$. Since the sentence mismatches can be fixed by merging sentences with sentence alignment algorithms (Sennrich and Volk, 2011), we assume that $N = M$. Therefore, we can assume that (x_i, y_i) is a parallel sentence pair.

If we use the target-side document context, there will be the translation error propagation problem (Wang et al., 2017): the errors made when translating a sentence will be propagated to the translation process of subsequent sentences. Also, leveraging source-side document-level context x_{-i}, which conveys the same information with y_{-i}, can better compute representations on the target side (Zhang et al., 2018). Therefore, we omit the target-side document-level context y_{-i}.

As a result, the document-level machine translation probability can be approximated as

$$P(Y \mid X; \boldsymbol{\theta}) \approx \prod_{i=1}^{N} P\left(y_i \mid x_i; x_{<i}; x_{>i}; \boldsymbol{\theta}\right) \tag{3.1}$$

where x_i is the source sentence aligned to y_i, $\{x_{<i}, x_{>i}\}$ are the document-level context sentences used to translate y_i.

3.2 Word Embedding

Our proposed model is mainly comprised of two modules: a pre-trained BERT module and a Transformer NMT module (Vaswani et al., 2017). For the input of the NMT module, we just regard the sentence as a tokens sequence. After preprocessing the data, we apply byte pair encoding (Sennrich et al., 2016) to divide words in all sentences into subword units and build a vocabulary upon these units. After that, we need to feed the sentence into the model. In order to enable the model to deal with those inputs, we leverage the word embedding layer to convert the preprocessed input sentences into vectors. Word embedding has been the basis for most of the neural network models, it has been widely used to solve various NLP tasks.

In our work, we train the NMT model's embedding by ourselves. About the embedding of the BERT module, we use the pre-trained BERT embedding.

In order to map the segmented words into the vector, firstly, we build a dictionary to map a word to an index. After that, we can achieve word vector from embedding table by the corresponding index. The procedure is like this:

$$\begin{aligned} i_x &= \operatorname{map}\left(w_x\right) \\ x &= \operatorname{select}\left(E, i_x\right) \end{aligned} \tag{3.2}$$

We denote a segmented word from the source sentence as w_x, i_x is its corresponding index. $E \in \mathbb{R}^{V \times d}$ is the embedding table, each row in the embedding table corresponds to a word vector which is d dimension. After achieving the index i_x, we select the corresponding row, and then we achieve the word vector x. Finally, we feed this vector into the encoder or decoder of the Transformer NMT model, it depends on whether it is the source language or target language. The process described in Figure 3.1

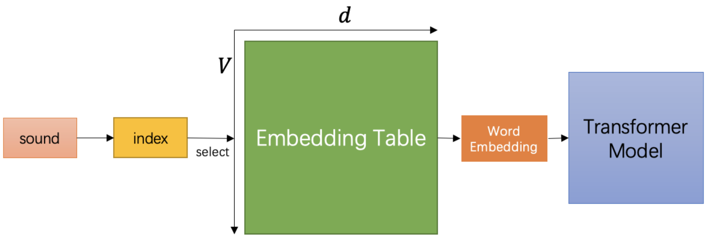


Figure 3.1: The process that how to feed a word into the Transformer model.

3.3 BERT Context Encoder

A pre-training task of BERT is the ”next sentence prediction” task, enable BERT to capture the relationship information between two sentences, this relationship information will be beneficial for document-level NMT. Similar with Zhang et al. (2018); Voita et al. (2018), we use an additional context encoder to model the document-level contextual information. To take advantage of the sentence relationship information captured by BERT, we leverage BERT as the context encoder.

Inspired by the input for NSP pre-training task, the input x_{ctx} for BERT context encoder is the concatenation of the document-level context sentences $(x_{<i}, x_{>i})$ and the current sentence x_i showing in Figure 3.2:

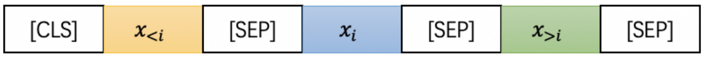


Figure 3.2: The input for BERT context encoder

Where ”$[SEP]$” and ”$[CLS]$” are the special tokens in BERT input. The context input x_{ctx} and the current sentences are encoded by BERT into document-level contextual representation $\mathbf{C}_B = BERT(x)$. Where $BERT(x)$ is the last layer's hidden state of the BERT context encoder.

3.4 BERT Context Representation Integration

Inspired by Zhang et al. (2018) using multi-head attention to integrate context representation, and Zhu et al. (2020) using the multi-head attention to integrate the current sentence representation encoded by BERT, we use the multi-head attention mechanism to incorporate BERT context representation $\mathbf{C}_B$ into both the encoder and the decoder of Transformer NMT module.

3.4.1 Integration into the Encoder

As shown in Figure 3.6, following Vaswani et al. (2017), we use a stack of L identical layers to encode the current sentence x_i. Every layer consists of two attention modules with different parameters. The first attention module is a multi-head self-attention which is the same with standard Transformer NMT model:

$$\mathbf{B}^{(l)} = \mathrm{MultiHead}\left(\mathbf{S}^{(l-1)}, \mathbf{S}^{(l-1)}, \mathbf{S}^{(l-1)}\right) \tag{3.3}$$

where $\mathbf{S}^{(0)}$ denotes the input word embedding of sentence x_i.

The second attention modules is a context attention that integrate BERT document-level context representation into the encoder as shown in the Figure 3.3:

$$\mathbf{D}^{(l)} = \mathrm{MultiHead}\left(\mathbf{S}^{(l-1)}, \mathbf{C}_B, \mathbf{C}_B\right) \tag{3.4}$$

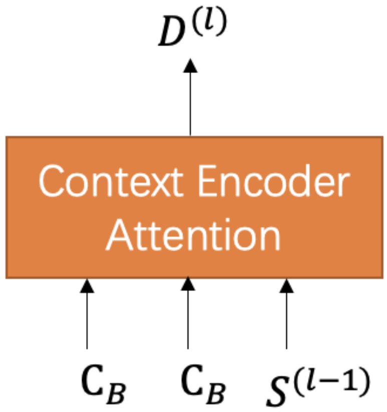


Figure 3.3: Context Attention

This context attention can also solve the problem that the Transformer NMT module and the BERT module may use distinct word segmentation rules.

3.4.2 Context Gate Integration

After achieving the outputs of the two attention modules, the most intuitive way to combine them is by adding the outputs directly. However, if we directly add the outputs of the two attention modules, the influence of document-level context will be enhanced in an uncontrolled way as the context information will be added to every layer. Also, when translating different source sentences, different amount of context information are required. Inspired by context gate in Werlen et al. (2018); Zhang et al. (2018), we propose to leverage context gate to combine the output of the two attention modules.

$$\begin{aligned} g^l &= \sigma\left(W_g^l\left[\mathbf{B}^{(l)}, \mathbf{D}^{(l)}\right] + b_g^l\right) \\ \mathbf{A}^{(l)} &= g^l \odot \mathbf{B}^{(l)} + \left(1 - g^l\right) \odot \mathbf{D}^{(l)} \end{aligned} \tag{3.5}$$

Where σ is a sigmoid function. Then the combination is further processed by a position-wise fully connected feed-forward neural network $FFN(.)$:

$$\mathbf{S}^{(l)} = FFN(\mathbf{A}^{(l)}) \tag{3.6}$$

$\mathbf{S}^{(l)}$ is the representation for the source sentence x_i and its context at the l-th layer.

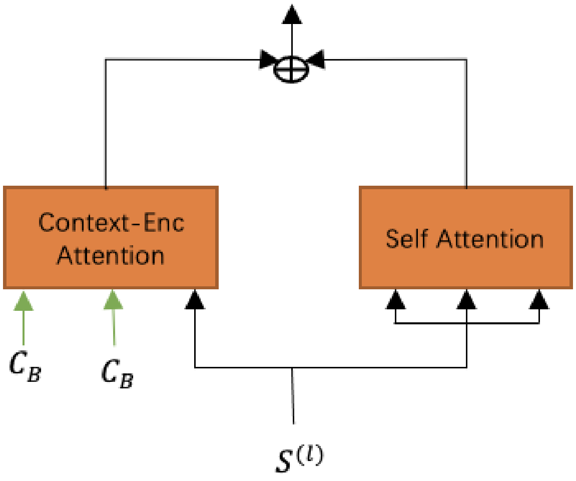


Figure 3.4: Illustration of Context Gate Integration

3.4.3 Integration into the Decoder

When generating the t-th target word $y_{i,t}$ in the i-th sentence, the partial translation is denoted by $\mathbf{y}_{i,<t} = y_{i,1}, ..., y_{i,t-1}$

Similar to the encoder layer, we use context gate and attention mechanism to integrate the BERT document-level context representation into standard Transformer decoder. Unlike the encoder, there are 3 attention modules in each layer of the decoder. In the l-th layer, the first attention module is a multi-head self-attention:

$$\mathbf{E}^{(l)} = \text{MultiHead}\left(\mathbf{T}^{(l-1)}, \mathbf{T}^{(l-1)}, \mathbf{T}^{(l-1)}\right) \tag{3.7}$$

where $\mathbf{T}^{(0)}$ is the word embedding of the partial translation $\mathbf{y}_{i,<t}$. The second attention module is a context attention that integrate document-level context representation into the decoder:

$$\mathbf{F}^{(l)} = \text{MultiHead}\left(\mathbf{E}^{(l)}, \mathbf{C}_B, \mathbf{C}_B\right) \tag{3.8}$$

The third attention module is an encoder-decoder attention that integrates the representation of the corresponding source sentence:

$$\mathbf{G}^{(l)} = \text{MultiHead}\left(\mathbf{E}^{(l)}, \mathbf{S}^{(L)}, \mathbf{S}^{(L)}\right) \tag{3.9}$$

Similar to the context gate integration in the encoder layer, we also use context gate to integrate document-level context representation into the decoder.

$$\begin{aligned} d^l &= \sigma\left(W_d^l\left[\mathbf{F}^{(l)}, \mathbf{G}^{(l)}\right] + b_d^l\right) \\ \mathbf{H}^{(l)} &= d^l \odot \mathbf{F}^{(l)} + \left(1 - d^l\right) \odot \mathbf{G}^{(l)} \end{aligned} \tag{3.10}$$

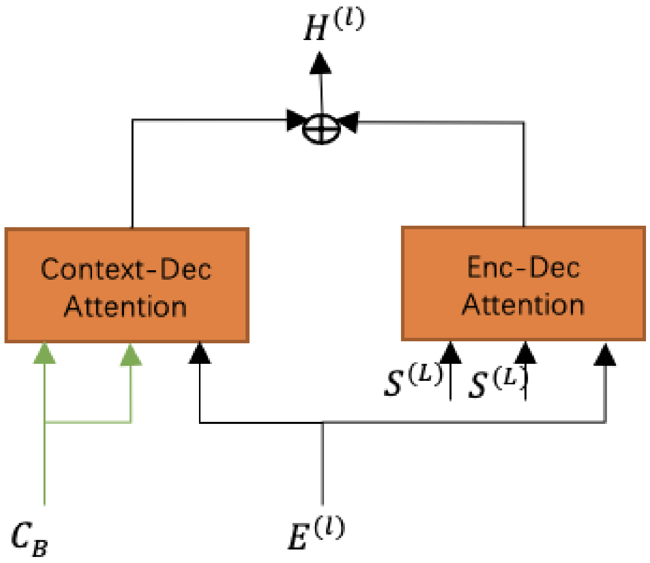


Figure 3.5: Integrating BERT context representation into decoder

The method of integrating BERT context representation into decoder is illustrate in the Figure 3.5.

Then we use a position-wise fully connected feed-forward neural network to achieve the final representation in the l-th layer.

$$\mathbf{T}^{(l)} = FFN(\mathbf{H}^{(l)}) \tag{3.11}$$

After achieving the final representations of the last decoder layer $\mathbf{T}^{(L)}$, the output probability of the current target sentence y_i are computed as:

$$\begin{aligned} & p\left(y_i \mid x_i, x_{<i}, x_{>i}\right) \\ & =\prod_t p\left(y_{i,t} \mid y_{i,\leq t}, x_i, x_{<i}, x_{>i}\right) \\ & =\prod_t \operatorname{softmax}\left(E\left[y_{i,t}\right]^{\top} \mathbf{T}_{i,t}^{L}\right) \end{aligned} \tag{3.12}$$

3.5 Overview of the Proposed Document-level NMT Model

We have shown the overview of our proposed model in Figure 3.6.

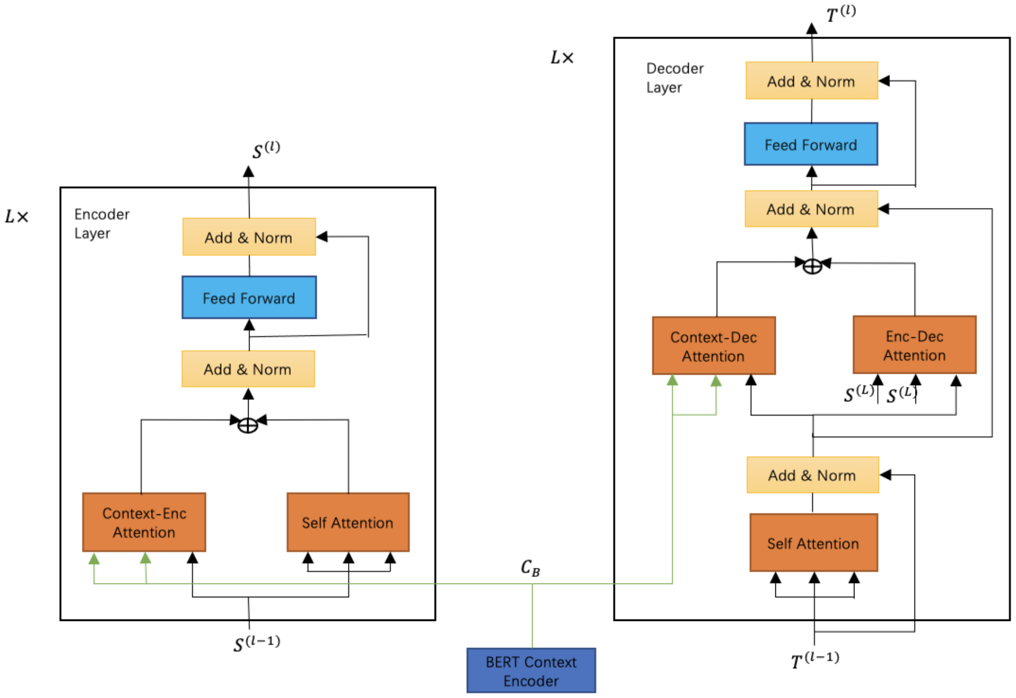


Figure 3.6: Illustration of using BERT as context encoder for document-level NMT model. C_B denote the output of BERT context encoder, $S^{(L)}$ denote the last layer output of Transformer encoder

3.6 Training

Given a document-level parallel corpus D_d, the training objective of document-level NMT model is maximizing the log-likelihood of the training data:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left\{ \sum_{\langle \mathbf{X}, \mathbf{Y} \rangle \in D_d} \log P(\mathbf{Y} \mid \mathbf{X}; \boldsymbol{\theta}) \right\} \tag{3.13}$$

The parameters in our model can be divided into two parts: the parameters of the pre-trained BERT module $\boldsymbol{\theta}_B$, the parameters of the Transformer NMT module $\boldsymbol{\theta}_N$. If the parameters of the pre-trained BERT module keep updating, the performance of the NMT system will not be improved (Zhu et al., 2020), which is different from the application of BERT in natural language understanding tasks, this is because applying BERT into NMT systems will suffer from catastrophic forgetting problem (Yang et al., 2019b). Therefore, in this work, we keep the parameters of BERT module $\boldsymbol{\theta}_B$ unchanged, we only update the parameters for the Transformer NMT module $\boldsymbol{\theta}_N$. This can also decrease the training time of our proposed document-level NMT model.

Since the parameter amount of the proposed model is very large, although the parameters of the BERT model have been fixed, if we train the document-level NMT model from scratch, the training process is still very time-consuming. Inspired by the two-step training process in Zhang et al. (2018), we also propose a two-step training process for our model.

At first, we divide the document-level parallel corpus D_d into sentence-level corpus D_s. Then we train a sentence-level Transformer NMT model using the sentence-level corpus D_s. The training objective of the sentence-level NMT is:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}_N}{\operatorname{argmax}} \left\{ \sum_{\langle \mathbf{X}, \mathbf{Y} \rangle \in D_s} \log P(\mathbf{Y} \mid \mathbf{X}; \boldsymbol{\theta}_N) \right\} \tag{3.14}$$

After achieving the sentence-level NMT model, the parameters of the Transformer NMT module $\boldsymbol{\theta}_N$ in our proposed model is initialized using the parameter of the sentence-level NMT model. Finally, we train our model using the document-level parallel corpus D_d:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}_N}{\operatorname{argmax}} \left\{ \sum_{\langle \mathbf{X}, \mathbf{Y} \rangle \in D_d} \log P(\mathbf{Y} \mid \mathbf{X}; \boldsymbol{\theta}_B; \boldsymbol{\theta}_N) \right\} \tag{3.15}$$

Our two-step training process is also similar to the training process of the document-level NMT model in Zhang et al. (2018). The main difference is that in the second step of their approach, they keep the parameter of the Transformer NMT model $\boldsymbol{\theta}_N$ fixed, they only update the parameter of the context encoder, while our approach keep the parameter of the context encoder fixed, we only update the parameter of the Transformer NMT model $\boldsymbol{\theta}_N$.

Chapter 4

Experimentation

In this chapter, we introduce the dataset we use in this work, the experiment settings and the baseline models we would like to compare with. Finally, we will explain the metric we use to evaluate the results.

4.1 Dataset

To prove the generalization ability of our proposed approach, we take experiments on two widely language pairs: English to German (En-De) and Chinese to English (Zh-En). For En-De language pair, we conduct experiments on two dataset: TED and New commentary. For Zh-En language pair, we take experiments on TED dataset. The details of the dataset are as follow:

- **TED (En-De)**: This corpus is from the IWSLT 2017 MT track (Cettolo et al., 2012), it has the TED talks' transcripts that are aligned at the sentence level. Every TED talk is regarded as a document. We use tst2016-2017 as our test set, the rest part are used as our validation set.

- **TED (Zh-En)**: This corpus is taken from the IWSLT 2015 evaluation campaigns (Cettolo et al., 2012). We take dev2010 as our validation set, and tst2010-2013 for our test set. There are 0.21 million sentence pairs in the training set, 887 sentence pairs in the validation set, 5.5 thousand sentence pairs in the test set.

- **News Commentary (En-De)**: We use the document-separated sentence-aligned News Commentary v11 [1] corpus as the training set. We take WMT'16 news-test2015 as our validation set, and news-test2016 for the test set.

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

Dataset	**Sent No.**	**Doc len avg**
TED	206126 / 8967 / 2271	121.4 / 96.4 / 98.7
News Commentary	236287 / 2169 / 2999	38.9 / 26.8 / 19.4

Table 4.1: Statistics of the train/valid/test corpora of En-De pair.

The corpora statistics of En-De language pair are shown in Table 4.1.

In order to compare the results of our approach to the previous state-of-the-art models conveniently, we obtain the preprocessed dataset from the previous work. For En-De language pair, we obtain the processed datasets from Maruf et al. (2019b)[2]. For Zh-En language pair, we obtain the processed datasets from Werlen et al. (2018)[3]. We apply the same train/valid/test datasets with the previous works.

4.2 Implementation Details

For English and German languages, we apply the scripts of Moses toolkit[4] to tokenize the sentences. For Chinese, we use the scripts of Jieba toolkit[5] to tokenize the sentences. We take byte pair encoding (Sennrich et al., 2016) to segment all sentences with 30K merge operations.

Firstly, we train a Transformer sentence-level NMT model until convergence, then use the obtained model to initialize the Transformer NMT module in our approach. The context encoder attention module and context decoder attention module are randomly initialized. For En-De language pair, the pre-trained BERT type is ”*bert_base_uncased*”. For Zh-En language pair, the pre-trained BERT type is ”*bert_base_chinese*”. Our BERT implementation is based on Huggingface[6] library. To balance the accuracy and the computation cost, we only use the concatenation of one previous sentence and the current sentence as the input for the BERT context encoder.

We use the same model configuration with the setting in the Maruf et al. (2019b). For the Transformer NMT model, the FFN layer dimension is 2048, and the hidden size is 512. There are 4 layers in the encoder and 4 layer in the decoder; the number of attention head is 8. The dropout (Srivastava et al., 2014) rate is 0.1 for sentence model and 0.2 for document-level model.

In the training process, we choose the Adam (Kingma and Ba, 2014) as the

[2] https://github.com/sameenmaruf/selective-attn

[3] https://github.com/idiap/HAN_NMT

[4] https://github.com/moses-smt/mosesdecoder

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

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

optimizer. About the hyper-parameters of Adam optimizer, the two momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.98$, $\epsilon = 1 \times 10^{-8}$. The learning rate linearly increases from 0 to 5×10^{-4} for the first 4000 warming-up steps and then decreases proportionally to the inverse square root of the update numbers. The batch size is limited to 4000 tokens. We also apply label smoothing to the cross-entropy loss, and the smoothing rate is 0.1. The deep learning library we use is Pytorch (Paszke et al., 2019). We use the open-source toolkit Fairseq (Ott et al., 2019)[7] as our Transformer NMT module and document-level NMT model training process implementation.

4.3 Baseline Models

We have compared the translation performance of 9 baseline models with our model in the experiment:

- **Tansformer**: Vaswani et al. (2017) proposed Transfomer model and achieved significant improvements over previous state-of-the-art RNN and CNN models, now it is being widely used as baseline model in machine translation research.
- **Hierarchical Attention**: Werlen et al. (2018) propose a hierarchical attention (HAN) document-level NMT model that can capture the connections between sentences in a dynamic and structured manner. HAN significantly improves the document translation performance over two strong NMT baselines.
- **Document-aware Transformer**: Zhang et al. (2018) use an additional context encoder to model the document-level contextual representation, the contextual representation is then integrated into the encoder and the decoder of original Transformer NMT model. Their approach improves over sentence-level Transformer significantly.
- **Selective Attention**: Maruf et al. (2019b) propose a novel method based on sparse attention to hierarchical attention for document-level NMT. Their approach is both efficient and scalable. Experiments on three En-De datasets showing their method outperform two recent document-level NMT baselines.
- **Transformer+Cache**: Tu et al. (2018) extend NMT models using a cache-like light-weight memory network that can store previous hidden represen-

[7] https://github.com/pytorch/fairseq

tations as the history of translation. In this way, the Transformer model can be applied to document-level MT tasks.

- **Query-Guided Capsule Network**: A Query-guided Capsule Network (QCN) Yang et al. (2019a) was proposed, it uses an improved dynamic routing algorithm for improving context modeling for the document-level NMT model. The Experiments on En-De in three domains showed their approach significantly outperformed sentence-level NMT model and achieved state-of-the-art results on two datasets.

- **Flat-Transformer**: Ma et al. (2020) propose a single-encoder document-level NMT model which can outperform the multi-encoder baseline models in terms of METEOR and BLEU scores. And the pre-trained model like BERT can further improve the translation quality of their proposed approach.

- **BERT-fused**: Zhu et al. (2020) proposed BERT-fused model, they first leverage BERT to achieve the representation of a source sentence, then the representation is integrated into every layer of the encoder and decoder of the Transformer NMT model. Their model can also be applied to document-level NMT tasks.

- **BERT-Doc**: Li et al. (2019) propose to use pre-trained language models like BERT in document-level NMT, they also propose a selective method for controlling the influence of huge contexts. Experiments on IWSLT datasets showed that our their systems achieved the state-of-the-art performance on Zh-En, Fr-En and Es-En language pairs.

4.4 Evaluation metrics

We have leveraged two metrics to evaluate the translation quality: BLEU score (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005).

4.4.1 BLEU score

BLEU (bilingual evaluation understudy) (Papineni et al., 2002) score is a widely used metrics to measure the quality of machine translation. The BLEU score is fast to calculate and does not rely on experience. In addition, it is language independent, which makes it a suitable tool for evaluating machine translation, especially when we are dealing with different language pairs.

4.4.2 METEOR score

Besides BLUE score, Maruf et al. (2019b); Yang et al. (2019a); Ma et al. (2020) have also apply another metric METEOR (Banerjee and Lavie, 2005) to evaluate the translation quality for document-level machine translation in En-De language pair. Meteor evaluates translations by calculating scores based on explicit word-to-word matches between the output translation and a given reference translation. For the En-De language pair, we have also applied this metric to compare with previous work.

Chapter 5

Evaluation

In this chapter, we show the experiments results of our proposed approach and compare the results to the baseline model. Also, We conduct ablation study to investigate the different integration ways. We also conducted ablation study. Firstly, we show that our approach can really capture contextual representation to improve the translation result. Finally, we analyze the results in detail to confirm the translation quality.

5.1 Translation performance

For En-De language pair, we take experiments on TED and News Commentary datasets, then we compare the results with seven previous works. The results are evaluated with BLEU and METEOR scores.

For Zh-De language pair, we take experiments on TED datasets, then we compare the results with three previous works. The results are evaluated with BLEU scores.

We list the results of our experiments in Table 5.1 for En-De pair, and Table 5.2 for Zh-En pair.

For En-De language pair, we compare with six document-level NMT baselines: Document-aware Transformer (Zhang et al., 2018), Hierarchical Attention NMT (Werlen et al., 2018), Selective Attention NMT (Maruf et al., 2019b) and Query-guided Capsule Network (Yang et al., 2019a), Flat-Transformer (Ma et al., 2020), using BERT for initializing the encoder of Flat-Transformer (+BERT). Most results of the previous work are taken from Ma et al. (2020), except for BERT-fused (Zhu et al., 2020). The result for BERT-fused (Zhu et al., 2020) is our re-implementation based on their code [1]. We the concatenation of the current

[1] https://github.com/bert-nmt/bert-nmt

Model	TED		News	
	BLEU	METEOR	BLEU	METEOR
HAN (Werlen et al., 2018)	24.58	45.48	25.03	44.02
SAN (Maruf et al., 2019b)	24.62	45.32	24.84	44.27
QCN (Yang et al., 2019a)	25.19	45.91	22.37	41.88
Doc-Transformer (Zhang et al., 2018)	24.01	45.30	22.42	42.30
Transformer (Vaswani et al., 2017)	23.28	44.17	22.78	42.19
Flat-Transformer (Ma et al., 2020)	24.87	47.05	23.55	43.97
+BERT	**26.61**	**48.53**	24.52	45.40
BERT-fused (Zhu et al., 2020)	25.59	47.71	25.05	45.51
Our Reproduced Transformer	23.99	45.57	22.50	42.80
Our Proposed Model	26.23	48.00	**26.55**	**47.25**

Table 5.1: Results on the two document-level machine translation benchmarks for En-De language pair

Model	**BLEU**
Transformer+Cache (Tu et al., 2018)	17.32
HAN (Werlen et al., 2018)	17.79
BERT-Doc (Li et al., 2019)	**20.72**
Our Reproduced Transformer	17.20
Our proposed model	19.01

Table 5.2: BLEU scores on TED dataset for Zh-En language pair

sentence and one previous sentence as the input for BERT module. The reproduced Transformer uses the 4-layers setting, which is the same as our proposed model. The drop-net rate is 1.0, which can achieve the best result according to their experiments.

For Zh-En language pair, we compare with three document-level NMT model: Transformer+Cache (Tu et al., 2018), HAN (Werlen et al., 2018), BERT-Doc (Li et al., 2019). The previous work's results are from Li et al. (2019).

For En-De language pair, as shown in Table 5.1, by leveraging document-level context representation given by BERT, our proposed model obtains 2.24/4.05 gains over our reproduced sentence-level Transformer baselines in terms of BLEU score, and 2.43/4.45 in terms of METEOR score. For Zh-En language pair, as shown in Table 5.2, our proposed approach obtains 1.81 gains over our reproduced sentence-level Transformer model in terms of BLEU score. Among them, our proposed approach achieves new state-of-the-art performance on the News dataset, showing the superiority of exploiting BERT document-level context representation.

Our proposed approach achieved huge improvements on the News dataset, but relatively smaller gains on the TED dataset and haven't achieved state-of-the-art performance. Since the BERT model is pre-trained using BooksCorpus and Wikipedia, and the documents in the News dataset is more similar to the pre-training corpus, BERT can better encode context information on the News dataset. Also, Li et al. (2019) and Ma et al. (2020) have used the pre-trained BERT model as the encoder of the Transformer NMT model, the amount of trainable parameters is larger than our proposed model.

5.2 Ablation study

5.2.1 Effect of Context Integration

In this part, we study the effectiveness of three BERT context representation integration method.

Table 5.3 shows the effect of integrating BERT context representation into only the encoder, only decoder, and both the encoder and the decoder of the Transformer model. As we can see, integrating BERT context representation into the encoder can achieve +3.15 BLEU score improvement, integrating BERT context representation into the decoder can achieve +3.05 BLEU score improvement, integrating BERT context representation into both the encoder and the decoder can achieve +4.05 BLEU score improvement.

We can find that integrating BERT context representation into the encoder brings more improvements, it is also beneficial to integrate representation into

Integration	BLEU
none	22.50
encoder	25.65
decoder	25.55
both	26.55

Table 5.3: Effectiveness of different BERT representation integration way. The ”none” denotes no BERT representation is integrated, the ”encoder” denote BERT representation is only integrated into the encoder, the ”decoder” denotes BERT representation is only integrated into the decoder, the ”both” denotes integrate BERT representation into both the encoder and the decoder.

the decoder. The results indicate that the BERT context representation should be integrated into both the encoder and decoder to achieve the best performance.

5.2.2 Does the BERT encoder really capture the contextual information?

Li et al. (2020) investigated whether context-encoder in multi-encoder document-level NMT model can capture contextual representation in the training process to improve translation quality, they provide three classes of input to the context encoder of multi-encoder document-level NMT model:

- *Context*: Concatenation of the previous source language sentence and the current source language sentence.
- *Fixed*: Concatenation of a fixed source language sentence and the current source language sentence.
- *Random*: Concatenation of a source language sentence composing of words randomly selected from the source language vocabulary and the current source language sentence.

The input of *Fixed* and *Random* are fake context input since they have not leveraged the real context of the current sentence. We infer that if the document-level NMT system relies on the contextual representation in the preceding sentences, the translation quality of *Fixed* and *Random* should significantly drop because of the incorrect context input. To our surprise, in the experiments of Li et al. (2020), in most cases, both *Fixed* and *Random* input can get comparable or even better translation quality than the right context input. In the rest part of Li et al. (2020), they give an explanation that the context encoder does not only capture context.

News	BLEU
Context	26.55
Fixed	26.14
Random	25.96

Table 5.4: BLEU scores using three context inputs

Instead, it is more similar to a noise generator, providing additional supervision signals for training the sentence-level NMT models.

To investigate whether the BERT context encoder has captured contextual representation to improve translation quality, we follow the experimental setting in Li et al. (2020) presenting three types of input for the BERT context encoder and make experiments using the News dataset.

As shown in Table 5.4, the performance of *Fixed* and *Random* decrease (-0.41 BLEU score for *Fixed* input, -0.59 BLEU score for *Random* input) because of the incorrect context, which is different from the result in Li et al. (2020). This indicates that the BERT context encoder in our proposed approach can really capture the contextual representation to improve translation performance.

Although the translation quality of *Fixed* and *Random* decreases, they can still outperform the standard Transformer model significantly (+3.64 BLEU score for *Fixed* input, +3.46 BLEU score for *Random* input) . This is because the current sentence usually plays a more important role in target sentence generation, although the wrong context input is given, our proposed model can still leverage the representation of the current sentence which is obtained by BERT, this representation can be used as the extra representation of current sentence to improve translation performance. This indicates the prospects of applying our proposed approach to sentence-level NMT model, we can use only the current sentence as the input for BERT encoder module, the sentence-level NMT model can also be improved significantly.

5.3 Analysis

We use three examples to illustrate how document-level context information helps translation Table 5.5, 5.6, 5.7.

In Table 5.5, ”shouguo jiaoyu” should be translated into ”educated”, but the Transformer model hasn’t translated it. Given the context ”qu shangxue”, which means go to school, our model translated it rightly. This example indicates that by integrating document-level context, our model can better understand word sense to generate translation.

Context	...wo de muqin, qu shangxue, bing yinci...
Source	danshi wo na shouguo jiaoyu de muqin chengwei le yiming jiaoshi
Reference	but my educated mother became a teacher.
Transformer	but my mother became a teacher.
Our model	but my mother, who was educated, became a teacher.

Table 5.5: An example of Chinese-English translation (1). ”shouguo jiaoyu de” should be translated into ”eduacated”. By taking advantage of ”qu shangxue” in the document-level context, our model translated this word correctly.

Context	er zhishi yanqi he yanjuan de yuanyin shi, zhongyiyuan duizhan canyiyuan
Source	zhongyiyuan buxiang rang huashengdun chenzui yu quanyi
Reference	The House of Representatives didn’t want Washington to get drunk on power.
Transformer	the House doesn’t want to let the House of Washington to be in power.
Our model	the House didn’t want Washington to spend time in power.

Table 5.6: An example of Chinese-English translation (2). This sentence should use past tense.

In Table 5.6, the translation should use past tense. The meaning of the context is ”And the reason for the delay and the boredom was that the House of Representatives were against the Senate.”, from the information given by the context, our model translate the tense rightly. This example indicates that by integrating document-level context, our model can better translate verb’s tense for English.

In Table 5.7, ”xiangmu” should be translated into ”project”, but the Transformer model hasn’t translated it. The context is ”suoyi –suoyi zhengfu deren shuo:” na jiezhe zuo.”, which means ”So — — So the government says, ’Do it again.’”, our model translated this word rightly. Although the context information is not helpful to translate this word, our model can still take advantage of the current sentence representation given by BERT encoder as the extra representation, the extra representation of current sentence given by BERT is also very helpful to improve translation quality, this is correspond to what we found in section 5.2.2.

Context	suoyi –suoyi zhengfu deren shuo:” na jiezhe zuo. ”
Source	women zai shijie shang 300 ge shequ kaizhan le zhege xiangmu.
Reference	we’ve done this project in 300 communities around the world.
Transformer	we have 300 communities in the world.
Our model	we started this project in 300 communities around the world.

Table 5.7: An example of Chinese-English translation (3). ”xiangmu” should be translated into ”project”.

Chapter 6

Conclusion

In this chapter, we give the conclusion of this research and several future research directions.

6.1 Conclusion

In this research, we propose a novel document-level NMT approach that uses the pre-trained BERT as a context encoder which can capture the document-level contextual information to improve translation performance. The document-level contextual information is integrated into the Transformer NMT model by using the multi-head attention mechanism and the context gate. To show the effectiveness of our approach, we took several experiments:

In the first part of our experiments, we trained our document-level NMT model using our proposed two-step training strategy, the model was trained using three datasets, two for English-to-German language pair, and one for Chinese-to-English language pair. Then we compare the results with previous state-of-the-art models.

In the second part, we tried to use three contextual representation integration ways. We tried to integrated BERT contextual representation into the encoder, the decoder, both the encoder and the decoder of the Transformer NMT model. Using this way, we can investigate the effectiveness of different integration way.

In the third part, we follow the experimental setting in Li et al. (2020) present three kinds of input for the BERT context encoder and compare the improvements from those three inputs. In this way, we can investigate whether the BERT model can really capture document-level contextual information to improve translation quality.

At last, we checked several translation examples to investigate where our document-level NMT approach can outperforme the sentence-level Transformer model.

The main conclusions are:

- Our proposed approach outperformed some strong document-level MT baseline models on English-to-German and Chinese-to-English language pair, achieving new state-of-the-art performance on the English-to-German News Commentary dataset, those shown the effectiveness and generalization ability of our approach.
- The results of different document-level contextual information integration way show that integrating contextual information into encoder can achieve more improvements than integrating into the decoder. Integrating document-level contextual information into both the encoder and the decoder, the best result can be archived.
- The results about presenting three kinds of input show that the BERT context encoder can really capture the document-level contextual information to improve translation performance.
- Even though given the wrong context input, the BERT encoder can still provide the extra representation of the current sentence to improve translation quality.

6.2 Limitation and future work

Our work has three main limitations:

- Although the BERT module in our NMT model is not trainable, the computation amount of our model is still very huge, the training and inference process is very time-consuming.
- We have only tried to use one previous sentence as document-level context, in practical, one sentence in a document is relate to more than one sentence in the document.
- BERT is majorly pre-trained on some large-resource languages such as English, Chinese, German, we have not extended our approach to low-resource languages.

Based on the limitations of this work, our future work is:

- Compress our model into a light version to reduce the parameter size, in this way, the training and inference time can also be reduced.

- Try to use more than one context sentences as BERT context encoder's input.
- Try to use XLM-R (Conneau et al., 2020), which is pre-trained using multilingual language, and test the performance on low-resource language to show the generization ability of our approach.

Bibliography

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, 2019.

Bei Li, Hui Liu, Ziyang Wang, Yufan Jiang, Tong Xiao, Jingbo Zhu, Tongran Liu, and changliang Li. Does multi-encoder help? a case study on context-aware neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3512–3518, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.322. URL `https://www.aclweb.org/anthology/2020.acl-main.322`.

Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. A survey on document-level machine translation: Methods and evaluation. *arXiv preprint arXiv:1912.08494*, 2019a.

Shuming Ma, Dongdong Zhang, and Ming Zhou. A simple and effective unified encoder for document-level machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3505–3511, 2020.

Lesly Miculicich Werlen, Dhananjay Ram, Nikolaos Pappas, and James Henderson. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954, 2018.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. Incorporating bert into neural machine translation. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=Hyl7ygStwB`.

Liangyou Li, Xin Jiang, and Qun Liu. Pretrained language models for document-level neural machine translation. *arXiv preprint arXiv:1911.03110*, 2019.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, and Robert L Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.

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

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*, 2018.

David Justice. " introduction to text linguistics", by robert-alain de beaugrande (book review). *Romance Philology*, 37(2):236–237, 1983.

Yves Scherrer, Jörg Tiedemann, and Sharid Loáiciga. Analysing concatenation approaches to document-level nmt in two different domains. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 51–61, 2019.

Deyi Xiong, Qun Liu, and Shouxun Lin. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 521–528, 2006.

Zhongjun He, Qun Liu, and Shouxun Lin. Improving statistical machine translation using lexicalized rule selection. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 321–328, 2008.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, 2018.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542, 2018.

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

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2019.

Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. Modeling coherence for discourse neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7338–7345, 2019.

Rongxiang Weng, Heng Yu, Shujian Huang, Shanbo Cheng, and Weihua Luo. Acquiring knowledge from pre-trained model to neural machine translation. *arXiv preprint arXiv:1912.01774*, 2019.

Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, and Jingjing Liu. Distilling knowledge learned in bert for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7893–7905, 2020.

Jörg Tiedemann and Yves Scherrer. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4811. URL `https://www.aclweb.org/anthology/W17-4811`.

Zhengxin Yang, Jinchao Zhang, Fandong Meng, Shuhao Gu, Yang Feng, and Jie Zhou. Enhancing context modeling with a query-guided capsule network for document-level translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1527–1537, 2019a.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936, 2019.

Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Yong Yu, Weinan Zhang, and Lei Li. Towards making the most of bert in neural machine translation. *arXiv preprint arXiv:1908.05672*, 2019b.

Rico Sennrich and Martin Volk. Iterative, mt-based sentence alignment of parallel texts. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*, pages 175–182, 2011.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. Exploiting cross-sentence context for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831, 2017.

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

Mauro Cettolo, Christian Girardi, and Marcello Federico. Wit3: Web inventory of transcribed and translated talks. In *Conference of european association for machine translation*, pages 261–268, 2012.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*, 2019.

Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. Learning to remember translation history with a continuous cache. *Transactions of the Association for Computational Linguistics*, 6:407–420, 2018.

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

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W05-0909`.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale, 2020.