Document-level Neural Machine Translation with Inter-Sentence Attention

Shu Jiang[1,2], Rui Wang[3], Zuchao Li[1,2], Masao Utiyama[3], Kehai Chen[3], Eiichiro Sumita[3], Hai Zhao[1,2*], Bao-liang Lu[1,2]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
[3] National Institute of Information and Communications Technology (NICT)
jshmjs45@gmail.com,wangrui,@nict.go.jp,charlee@sjtu.edu.cn,
{eiichiro.sumita,khchen,mutiyama}@nict.go.jp,
zhaohai@cs.sjtu.edu.cn,bllu@sjtu.edu.cn

Abstract

Standard neural machine translation (NMT) is on the assumption of document-level context independent. Most existing document-level NMT methods only focus on briefly introducing document-level information but fail to concern about selecting the most related part inside document context. The capacity of memory network for detecting the most relevant part of the current sentence from the memory provides a natural solution for the requirement of modeling document-level context by document-level NMT. In this work, we propose a Transformer NMT system with associated memory network (AMN) to both capture the document-level context and select the most salient part related to the concerned translation from the memory. Experiments on several tasks show that the proposed method significantly improves the NMT performance over strong Transformer baselines and other related studies.

1 Introduction

Neural Machine Translation (NMT) (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017) established on the encoder-decoder framework, where the encoder takes a source sentence as input and encodes it into a fixed-length embedding vector and the decoder generates the translation sentence according to the encoder embedding, has achieved advanced translation performance in recent years. So far, most models take a standard assumption to translate every sentence independently, ignoring the document-level contextual clues during translation. However, document-level information can improve the translation performance from multiple aspects: consistency, disambiguation, and coherence (Kuang et al., 2018). If translating every sentence is independent of document-level context, it will be difficult to keep every sentence translations across the entire text consistent with each other. Moreover, the document-level context can also assist the model to disambiguate words with multiple senses. At last, the global context helps translate in a coherent way.

There have been few recent attempts to introduce the document-level information into the existing standard NMT models. Jean et al. (2017) model the context from the surrounding text in addition to the source sentence, and Tiedemann and Scherrer (2017) extend the source sentence and translation units with the contextual segments to improve the translation. Wang et al. (2017) use a hierarchical Recurrent Neural Network (RNN) to import the information of previous sentences. Miculicich et al. (2018) propose a multi-head hierarchical attention machine translation model to capture the word-level and sentence-level information. The cache-based model raised by Kuang et al. (2018) uses the dynamic cache and topic cache to capture the connection from neighboring sentences. In addition, Wang et al. (2017), Kuang and Xiong (2018) and Voita et al. (2018) all add the contextual information to the NMT model by applying the gating mechanism proposed by Tu et al. (2017) to dynamically control the auxiliary global context information at each decoding step. However, most of the existing document-level NMt methods have to inconveniently prepare

the contextual input or model the global context in advance.

Inspired by the observation that human and document-level machine translation model always refer to the context of the source sentence during the translation, like query in their memory, we propose to utilize the document-level sentences associated with the source sentences to help predict the target sentence. To reach such a goal, we adopt a Memory Network component(Weston et al., 2015; Sukhbaatar et al., 2015; Guan et al., 2019) which provides a natural solution for the requirement of modeling document-level context in document-level NMT. In fact, Maruf and Haffari (2017) have already presented a document-level NMT model which projects the document contexts into the tiny dense hidden state space for RNN model using memory networks and updates word by word, and their model is effective in exploiting both source and target document context.

Differing from any previous work, this paper presents a Transformer NMT model with document-level Memory Network enhancement (Weston et al., 2015; Sukhbaatar et al., 2015) which concludes contextual clues into the encoder of the source sentence by the Memory Network. Not like the work of Maruf and Haffari (2017)'s which memorizes the whole document information into a tiny dense hidden state, the memory in our work calculates the associated document-level contextualized information in the memory with the current source sentence using attention mechanism. In this way, our proposed model is able to focus on the most relevant part of the concerned translation from the memory which exactly encodes the concerned document-level context.

The empirical results indicate that our proposed method significantly improves the BLEU score compared with a strong Transformer baseline and performs better than other related models for document-level machine translation on multiple language pairs with multiple domains.

2 Related Work

The existing work about NMT on document-level can be divided into two parts: one is how to obtain the document-level information in NMT, and the other is how to integrate the document-level information.

2.1 Mining Document-level Information

Concatenation Tiedemann and Scherrer (2017) propose to simply extend the context during the NMT model training in different ways: (1) extending the source sentence which includes the context from the previous sentences in the source language, and (2) extending translation units which increase the segments to be translated.

Document RNN Wang et al. (2017) propose a cross-sentence context-aware RNN approach to produce a global context representation called Document RNN. Given a source sentence in the document to be translated and its N previous sentences, they can obtain all sentence-level representations after processing each sentence. The last hidden state represents the summary of the whole sentence as it stores order-sensitive information. Then the summary of the global context is represented by the last hidden state over the sequence of the above sentence-level representations.

Specific Vocabulary Bias Michel and Neubig (2018) propose a simple yet parameter-efficient adaption method that only requires adapting the bias of output softmax to each particular use of the NMT system and allows the model to better reflect personal linguistic variations through translation.

2.2 Integrating Document-level Information

Adding Auxiliary Context Wang et al. (2017) add the representation of cross-sentence context into the equation of the probability of the next word directly and jointly update the decoding state by the previous predicted word and the source-side context vector.

Gating Auxiliary Context Tu et al. (2017) introduce a context gate to automatically control the ratios of source and context representations contributions to the generation of target words. Wang et al. (2017) also introduce this mechanism in their work to dynamically control the information flowing from the global text at each decoding step.

Inter-sentence Gate Model Kuang and Xiong (2018) propose an inter-sentence gate model, which is based on the attention-based NMT and uses the same encoder to encode two adjacent sentences and controls the amount of information flowing from the preceding sentence to the translation of the current sentence with an inter-sentence gate. This gate framework assigns element-wise

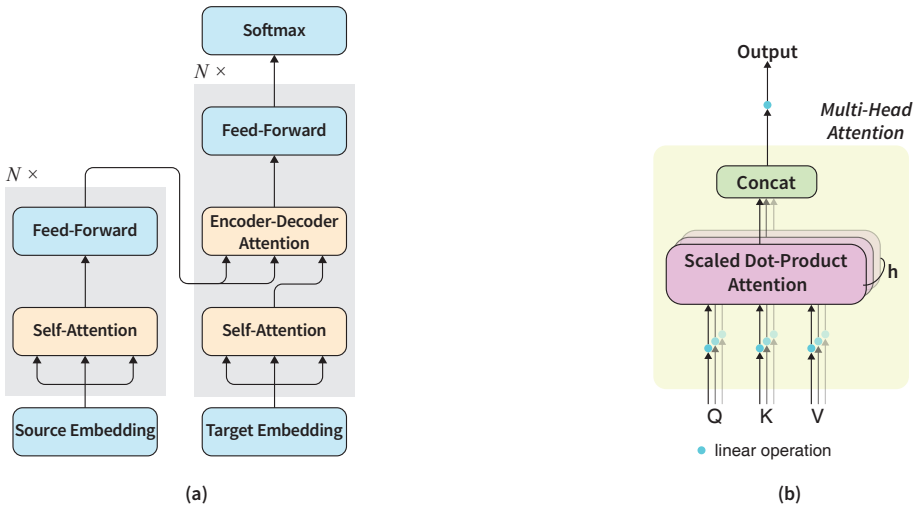

Figure 1: (a) Transformer architecture. (b) Multi-Head attention.

weights to the input signals which are calculated by the context vectors of two adjacent sentences, target word representation and the decoder hidden state.

Cache-based Neural Model Tu et al. (2018) propose to augment NMT models with an external cache to exploit translation history. At each decoding step, the probability distribution over generated words is updated online depending on the translation history retrieved from the cache with a query of the current attention vector, which assists NMT models to dynamically adapt over time. The cache-based neural model proposed by Kuang et al. (2018) consists of two components: topic cache and dynamic cache. When the decoder shifts to a new test document, the topic cache is emptied and filled with target topical words for the new test document. The dynamic cache is continuously expanded with newly generated target words from the best translation hypothesis of previous sentences. The final word prediction probability for the target word is calculated by a gate mechanism which combines the prediction probability from the cache-based neural model and the original NMT decoder.

Hierarchical Attention Networks Miculicich et al. (2018) propose a Hierarchical Attention Networks (HAN) NMT model to capture the context in a structured and dynamic pattern. For each predicted word, it uses word-level and sentence-level abstractions and selectively focuses on different words and sentences.

Context-Aware Transformer Voita et al. (2018) introduce the context information into the Transformer (Vaswani et al., 2017) and leave the Transformer's decoder intact while processing the context information on the encoder side. The model calculates the gate from the source sentence attention and the context sentence attention, then exploits their gated sum as the encoder output. Zhang et al. (2018) also extend the Transformer with a new context encoder to represent document-level context while incorporating it into both the original encoder and decoder by multi-head attention.

3 Background

3.1 Neural Machine Translation

Given a source sentence $\mathbf{x} = \{x_1, ..., x_i, ..., x_S\}$ in the document to be translated and a target sentence $\mathbf{y} = \{y_1, ..., y_i, ..., y_T\}$, NMT model computes the probability of translation from the source sentence to the target sentence word by word:

$$P(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{T} P(y_i|y_{1:i-1}, \mathbf{x}), \quad (1)$$

where $y_{1:i-1}$ is a substring containing words $y_1, ..., y_{i-1}$. Generally, with an RNN, the probability of generating the i-th word y_i is modeled as:

$$P(y_i|y_{1:i-1}, \mathbf{x}) = \text{softmax}(g(y_{i-1}, \mathbf{s}_{i-1}, \mathbf{c}_i)), \quad (2)$$

where $g(\cdot)$ is a nonlinear function that outputs the probability of previously generated word y_i, and

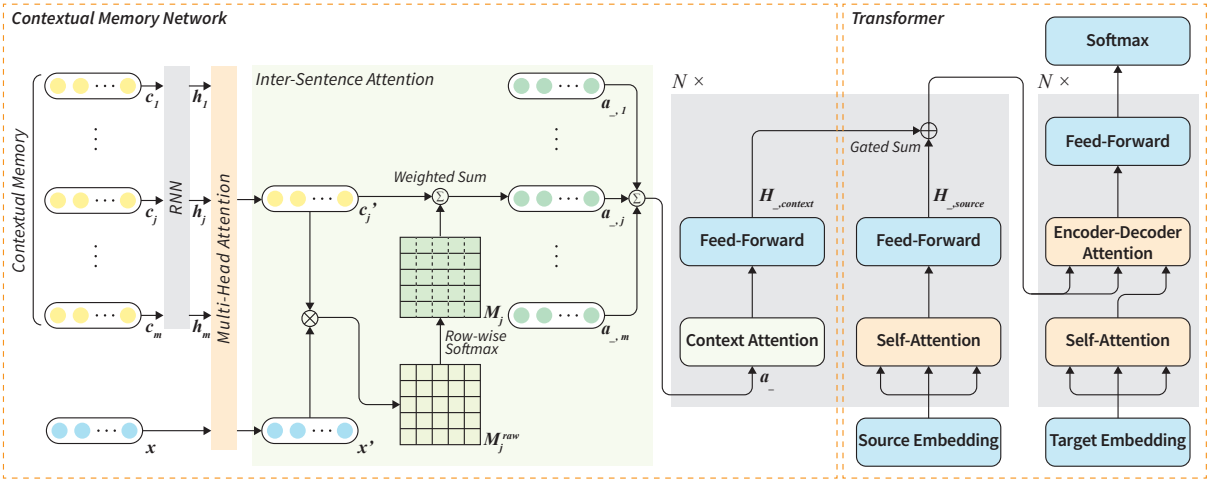

Figure 2: The framework of our model.

$\mathbf{c}_i$ is the i-th source representation. Then i-th decoding hidden state $\mathbf{s}_i$ is computed as

$$\mathbf{s}_i = f(\mathbf{s}_{i-1}, y_{i-1}, \mathbf{c}_i). \quad (3)$$

For NMT models with an encoder-decoder framework, the encoder maps an input sequence of symbol representations $\mathbf{x}$ to a sequence of continuous representations $\mathbf{z} = \{z_1, ..., z_i, ..., z_S\}$. Then, the decoder generates the corresponding target sequence of symbols $\mathbf{y}$ one element at a time.

3.2 Transformer Architecture

Only based on the attention mechanism, Vaswani et al. (2017) propose a network architecture called Transformer for NMT, which uses stacked self-attention and point-wise, fully connected layers for both encoder and decoder.

As illustrated in Figure 1 (a), the The encoder is composed of a stack of N (usually equals to 6 identical layers and each layer has two sub-layers: (1) multi-head self-attention mechanism, and (2) a simple, position-wise fully connected feed-forward network.

Multi-head attention demonstrated in the Figure 1 (b) in the Transformer allows the model to jointly process information from different representation spaces at different positions. It linearly projects the queries Q, keys K, and values V h times with different, learned linear projections to d_k, d_k, and d_v dimensions respectively, then the attention function is performed in parallel, generating d_v-dimensional output values, and yielding the final results by concatenating and once again projecting them. The core of multi-head attention is Scaled Dot-Product Attention and calculated as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V. \quad (4)$$

The second sub-layer is a feed-forward network, which contains two linear transformations with a ReLU activation in between.

Similar to the encoder, the decoder is also composed of a stack of N identical layers but it inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. The Transformer also employs residual connections around each of the sub-layers, followed by layer normalization. Thus, the Transformer is more parallelizable and faster for translating than earlier RNN methods.

3.3 Memory Network

Memory networks (Weston et al., 2015) utilize the external memories as inference components based on long-range dependencies, which can be categorized into a sort of lazy machine learning (Aha, 2013). Using the similar memorizing mechanism, memory-based learning methods have been also applied in multiple traditional models (Daelemans, 1999; Fix and Hodges Jr, 1951; Skousen, 1989, 2013; Lebowitz, 1983; Nivre et al., 2004). A memory network introduced by Weston et al. (2015) is a set of vectors $M = \{\mathbf{m}_1, ..., \mathbf{m}_K\}$ and the memory cell $\mathbf{m}_k$ is potentially relevant to a discrete object (for example, a word) x_k. The memory is equipped with a *read* and optionally a *write* operation. Given a query vector $\mathbf{q}$, the output vector produced by reading from the memory

is $\sum_{i=1}^{K} p_i \mathbf{m}_i$, where $p_i = \text{softmax}(\mathbf{q}^T \cdot M)$ scores the match between the query vector $\mathbf{q}$ and the i-th memory cell $\mathbf{m}_i$.

4 Model

4.1 Framework

Our NMT model consists of two components: Contextual Associated Memory Network (CAMN) and a Transformer model. For the CAMN component, the core part is a neural controller, which acts as a "processor" to read memory from the contextual storage "RAM" according to the input before sending this memory to other components. The controller calculates the correlation between the input and memory data i.e. "memory addressing".

4.2 Encoders

Our model requires two encoders, the context encoder for CAMN and the source encoder for translation from input sentence representation.

Inspired by Wang et al. (2017) which introduces the Document RNN to summarize the cross-sentence context information, we use an RNN on the context sentence to generate the context representation, and the hidden state at each time step can represent the relation from the first word to the current word. The source encoder is composed of a stack of N layers, as the same as the source encoder in the original Transformer (Vaswani et al., 2017).

4.3 Contextual Associated Memory Network

The proposed contextual associated memory network consists of three parts, context selection, inter-sentence attention, and context gating.

Context Selection

We aim to utilize the context sentences and their representations to assist our model to predict the target sentences. For the sake of fairness, we can treat all sentences in the document as our source. However, it is impossible to attend all the sentences in training dataset because of the extremely high computing and memorizing cost. According to Voita et al. (2018), whose model gets the best performance when using a context encoder for the previous sentence, we use the previous sentence of the source sentence $\mathbf{x}$ as the context sentence $\mathbf{c}$. Then, at each training step, we compose all the context sentences of the source sentences in the batch with size m as the context sentences $\{\mathbf{c}_j\}_{j=1}^{m}$.

Inter-Sentence Attention

This part aims to attain the inter-sentence attention matrix, which can be also regarded as the core memory part of the CAMN. The input sentence x and the context sentences in the memory $\{\mathbf{c}_j\}_{j=1}^{m}$ first go through a multi-head attention layer to encode the contextualized information to each word representation:

$$\mathbf{x}' = \text{MultiHead}(\mathbf{x}, \mathbf{x}, \mathbf{x}), \tag{5}$$

$$\mathbf{c}'_j = \text{MultiHead}(\mathbf{h}_j, \mathbf{h}_j, \mathbf{h}_j) \;\; j \in \{1, 2, ..., m\}, \tag{6}$$

where $\mathbf{h}_j$ is the RNN output of the context sentence $\mathbf{c}_j$ and the hidden state $h_{k,j}$ at time k is

$$h_{k,j} = f(h_{k-1,j}, c_{k,j}), \tag{7}$$

where $f(\cdot)$ is an activation function, and $c_{k,j}$ is the k-th word in the context sentence $\mathbf{c}_j$.

The lists of new word representations are denoted as follows:

$$\mathbf{x}' = \{x'_1, ..., x'_i, ..., x'_S\} \tag{8}$$

and

$$\mathbf{c}'_j = \{c'_{1,j}, ..., c'_{k,j}, ..., c'_{K_j,j}\} \tag{9}$$

Each word representation is as a vector $x \in \mathbb{R}^d$, where d is the size of hidden state in MultiHead function.

Then, for each context sentence representation $\mathbf{c}'_\mathbf{j}$, we apply the multi-head attention by treating the input sentence representation $\mathbf{x}'$ as the query sequence, on them and get the attention matrix M_j^{raw}:

$$M_j^{raw} = \mathbf{x}' \otimes \mathbf{c}_j'^T. \tag{10}$$

Every element $M_{raw}(i,k) = x'_i \cdot c'^T_{k,j}$ can be regarded as an indicator of similarity between the i-th word in input sentence representation $\mathbf{x}'$ and the k-th word in memory sentence representation $\mathbf{c}'_j$.

Finally, we perform a softmax operation on every column in M_j^{raw} to normalize the value so that it can be considered as the probability from input sentence representation $\mathbf{x}'$ to memory sentence representation $\mathbf{c}'_j$:

$$\alpha_{i,j} = \text{softmax}([M_j^{raw}(i,1), ..., [M_j^{raw}(i,K_j)]), \tag{11}$$

$$M_j = [\alpha_{1,j}, ..., \alpha_{i,j}, \alpha_{S,j}]. \quad (12)$$

We treat the probability vector $\alpha_{i,j}$ as a set of weights to sum all the representations in $\mathbf{c}'_j$ and get the memory-sentence-specified argument embedding $a_{i,j}$:

$$a_{i,j} = \alpha_{i,j} \cdot \mathbf{c}'^T_j = \sum_{k=1}^{K_j} \alpha_{i,j} c'_{k,j}. \quad (13)$$

Because the context sentences are different, the overall contributions of these word representations should be different as well. We let the model itself learn how to make use of these contextual word representations.

Following the attention combination mechanism (Libovický and Helcl, 2017), we use a weighted average strategy to combine these attention representations from different memory sources. We calculate the mean value of every raw similarity matrix M^{raw}_j to indicate the similarity between input sentence $\mathbf{x}$ and context sentence $\mathbf{c}_j$, and we use the softmax function to normalize them to get a probability vector β indicating the similarity of input sentence $\mathbf{x}$ towards all the associated sentences $\{\mathbf{c}_j\}^m_{j=1}$:

$$\begin{aligned}\beta &= \text{softmax}([g(M^{raw}_1), ..., g(M^{raw}_m)]) \\ &= [\beta_1, ..., \beta_j, ...\beta_m], \end{aligned} \quad (14)$$

where $g(\cdot)$ represents the mean function. Then, we use the probability vector β as weight to sum all the contextual attention embedding $a_{i,j}$ for the final contextual attention embedding a_i of the i-th word x_i in input sentence $\mathbf{x}$:

$$a_i = \sum_{j=1}^{m} \beta_j a_{i,j}. \quad (15)$$

Context Gating

We annotate the i-th source attention embedding x'_i and i-th contextual attention embedding a_i after the feed-forward operation as $H_{i,source}$ and $H_{i,context}$. Then we use a context gate (Tu et al., 2017) to integrate the source and context attentions and control the flow from the source side and the context side. The gate g_i is calculated by

$$g_i = \sigma(W_g[H_{i,source}, H_{i,context}] + b_g). \quad (16)$$

Their gated sum H_i is

$$H_i = g_i \otimes H_{i,source} + (1 - g_i) \otimes H_{i,context}, \quad (17)$$

where σ is the logistic sigmoid function, $\otimes$ is the point-wise multiplication and W_g is trained by the model. As illustrated in Figure 2, the output of the gate H_i is integrated into the encoder-decoder attention part at decoding step.

Multiple Context Attention

For multiple context sentences in the memory (m¿1) , we have two ways to integrate the memory information. One way is *concatenate multiple context attention*which concatenate the multiple context sentences into one context sequence with the break symbol '####' (Wang et al. 2017) to identify the sentence boundary.

$$c = \{c_1, \text{'####'}, c_2, \text{'####'}, ..., \text{'####'}, c_m\}. \quad (18)$$

The other way is *parallel multiple context attention* which calculates the weighted sum among the attentions between the each context sentence and the current sentence by softmax function as shown in Eq. (13) [1].

5 Experimental Setup

Table 1: Data statistics of sentences.

	TED Talks		Subtitles	News
	Zh-En	Es-En	Es-En	Es-En
Training	209,941	180,853	48,301,352	238,872
Tuning	887	887	1,000	2,000
Test	5,473	4,706	1,000	14,522

5.1 Data

The proposed document-level NMT model will be evaluated on two language pairs, i.e., Chinese-to-English (Zh-En) and Spanish-to-English (Es-En) on three domains: talks, subtitles, and news.

TED Talks Zh-En TED talk documents are the parts of the IWSLT2015 Evaluation Campaign Machine Translation task[2]. We use *dev2010* as the development set and combine the *tst2010-2013* as

[1] In this paper, due to the lack of the computational resource, we experiment only with the concatenate multiple context attention until now, for the experiment on parallel multiple context attention, we leave for the next version.

[2] https://wit3.fbk.eu/mt.php?release=2015-01

Table 2: BLEU scores on the different datasets. The marks "†" after scores indicate that the proposed methods were significantly better than the baseline Transformer at significance level p-value<0.05 (Collins et al., 2005). The scores in bold indicate the best ones on the same dataset.

	TED Talks		Subtitles	News
Models	Zh-En	Es-En	Es-En	Es-En
RNNSearch*	16.09	36.55	39.90	22.94
Transformer	17.76	39.03	39.96	23.71
Context-aware Transformer (Voita et al., 2018)	18.24	38.74	40.19	23.76
Transformer with HAN (Miculicich et al., 2018)	17.79	37.24	36.23	22.76
Our model	**18.65**†	**39.19**†	**40.70**†	**24.38**†

Table 3: Ablation study on these datasets.

	TED Talks		Subtitles	News
Models	Zh-En	Es-En	Es-En	Es-En
Contextual Associated Memory Network	**18.65**	**39.19**	**40.70**	**24.38**
- w/o RNN Context	18.44	38.46	40.10	22.87
- w/o Inter-sentence Attention	18.36	38.74	40.38	23.96
- w/o RNN Context & Inter-sentence Attention	17.92	38.46	39.96	22.09

the test set. The Es-En corpus is also a subset of the IWSLT2014. We use the *dev2010* for development set and *test2010-2012* as the test set.

Subtitles The Es-En corpus is a subset of OpenSubtitles2018[3] (Lison and Tiedemann, 2016)[4]. We randomly select 1,000 continuous sentences for each development set and test set.

News The Es-En News-Commentaries11 corpus[5] has document-level delimitation. We evaluate on the WMT sets (Bojar et al., 2013): *newstest2008* for development, and *newstest2009-2013* for testing.

Table 1 lists the statistics of all the concerned datasets.

5.2 Data preprocessing

The English and Spanish datasets are tokenized by *tokenizer.perl* and truecased by *truecase.perl* provided by MOSES[6], a statistical machine translation system proposed by Koehn et al. (2007). The Chinese corpus is tokenized by *Jieba* Chinese text segmentation[7]. Words in sentences are segmented into subwords by Byte-Pair Encoding (BPE) (Sennrich et al., 2016) with 32k BPE operations.

5.3 Model Configuration

We use the Transformer proposed by Vaswani et al. (2017) as our baseline and implement our work using the THUMT, an open-source toolkit for NMT developed by the Natural Language Processing Group at Tsinghua University (Zhang et al., 2017)[8]. We follow the configuration of the Transformer "base model" described in the original paper (Vaswani et al., 2017). Both encoder and decoder consist of 6 hidden layers each, and we choose the previous sentence as the context sentence in the memory network. All hidden states have 512 dimensions, 8 heads for multi-head attention and the training batch contains about 6,520 source tokens. Finally, we evaluate the performance of the model by BLEU score (Papineni et al., 2002) using *multi-bleu.perl* on the *tokenized* text.

6 Results

6.1 Translation Performance

Table 2 demonstrates the BLEU scores for different models on multiple corpora. The baseline is a re-implemented attention-based NMT system RNNSearch* (Hinton et al., 2012) and Transformer (Vaswani et al., 2017) using THUMT kit.

[3] http://www.opensubtitles.org/

[4] http://opus.nlpl.eu/OpenSubtitles2018.php

[5] http://opus.nlpl.eu/News-Commentary11.php

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

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

[8] https://github.com/thumt/THUMT

Table 4: Results on TED Talks (Zh-En) dataset with different context sentence size N and context selection

Context selection	N=1	N=2	N=3	N=4	N=5
Previous N sentence(s)	**18.65**	18.46	18.14	18.03	15.53
Next N sentence(s)	18.57	18.45	17.69	17.43	15.14
Random N context sentence(s)	18.38	18.37	18.11	17.42	15.88

Table 5: Example of the translation result. The context sentences are three previous sentences before the source sentence and words in deeper blue from context indicate more heuristic clues for better translation. Salient contextual words have been provided with English translation.

Context sentences	它 去年(last fall) 秋天 遭遇 破产 因为 他们 遭到 入侵 。 有人 闯进去 彻底 毁 了 它 。 我 上周(last week) 在 与 荷兰政府 代表 开会 时 问过(asked) , 我 问(asked) 一位 领导 是否 他(he) 发现 有 可能 有人 会 因为(because of) Diginotar 攻击 而 死亡 。
Source sentence	他的回答(answer)是肯定的。
Reference sentence	**and** his answer **was** yes .
Transformer model	his answer is yes .
HAN model	his answer **was** yes .
Our model	**and** his answer **was** yes .

We also employ the Context-aware model proposed by Voita et al. (2018) on these datasets.

The results in Table 2 demonstrate that our proposed model significantly outperforms all the comparing models, especially, our model is significantly better than the baseline Transformer at significance level p-value$<$0.05. Our proposed model outperforms the RNNSearch* baseline by 2.56 BLEU point on the TED Talks (Zh-En) dataset, 2.64 BLEU point on the TED Talks (Es-En) dataset, 0.80 BLEU point on the OpenSubtitles (Es-En) dataset and 1.44 BLEU point on the WMT dataset (Es-En).

Furthermore, our proposed model achieves the gains of 0.89 BLEU point, 0.16 BLEU point, 0.74 BLEU point, and 0.44 BLEU point on these four datasets individually over the Transformer baseline. Compared with the Context-aware Transformer proposed by Voita et al. (2018), our proposed approach also raises the average 0.49 BLEU score on these different datasets. Moreover, the average increase of BLEU score over the Transformer with HAN proposed by Miculicich et al. (2018)[9] is 2.23 point.

[9]The results of HAN are reported by its authors.

6.2 Ablation Experiments

We investigate the impact of different components of our model by removing one or more of them.

- If we do not employ the RNN operation on the context encoder, the multi-head attention works directly on the context embedding.

- If the model is trained without the inter-sentence attention module of the CAMN, we select the context sentence randomly from the training set, and the context attention is generated by the hidden states of the context embedding after RNN.

- If we remove the RNN operation and the inter-sentence attention, the context attention is produced by the word embedding of randomly selected context sentence and the context encoder with a stack of N multi-head attention and feed-forward layers is as the same as the source encoder.

As shown in Table 3, all of the components greatly contribute to the performance of our proposed model. If we remove any step in Context Encoder, the performance drops dramatically. Such results indicate that all features introduced by our CAMN enhanced model play an important and complementary role in our model.

6.3 Effect of Contextual Information

• **Different definition of context sentence** The context sentence in our work is the previous sentence of the current sentence. We investigate the effect of the different context sentence definition on the TED Talks (Zh-En) dataset. Like the work of (Voita et al., 2018), we use the context encoder for the previous sentence, next sentence and the random selected context sentence form the document.

• **Different context size** We also compare the effect with the different context size N on the TED Talks (Zh-En) dataset.

As shown in the Table 4, the model use the previous sentence as the context encoder could get the best performance on the TED Talks (Zh-En) dataset. Moreover, more contextual information by *concatenate multiple context attention* does not appear beneficial and the BLEU score does not get better with longer context sentence.

6.4 Translation Quality

Table 5 shows an example from the TED Talks (Zh-En)[10], on which the translation our model is compared to those of other methods. The translation of HAN model is downloaded from Miculicich et al. (2018)'s GitHub[11]. This example shows that our proposed model is capable of recognizing the tense and even discourse relation from document-level context.

7 Conclusion and Future Work

We propose a memory network enhancement over Transformer based NMT which provides a natural solution for the requirement of modeling document-level context. Experiments show that our model performs better on the datasets of multiple domains and language pairs and has the ability to capture salient document-level contextual clues and select the most relevant part related to the input sequence from the memory.

In our future work, we consider introducing the discourse information to enhance our model. But it will bring a lot of noise, and the internal structure may be particularly complex. Therefore, it is necessary to effectively abstract its key feature information. The discourse information will provide the heuristic features that will improve the performance during the training and decoding.

[10]This example is extracted from line 4,123 of TED Talks (Zh-En).

[11]https://github.com/idiap/HAN_NMT/tree/master/test_out

References

David W Aha. 2013. *Lazy learning*. Springer Science & Business Media.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, pages 1–15, San Diego, USA.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 531–540, Ann Arbor, Michigan. Association for Computational Linguistics.

Walter Daelemans. 1999. Introduction to the special issue on memory-based language processing. *Journal of Experimental & Theoretical Artificial Intelligence*, 11(3):287–296.

Evelyn Fix and Joseph L Hodges Jr. 1951. Discriminatory analysis-nonparametric discrimination: consistency properties. Technical report, California Univ Berkeley.

Chaoyu Guan, Yuhao Cheng, and Hai Zhao. 2019. Semantic role labeling with associated memory network. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3361–3371, Minneapolis, Minnesota. Association for Computational Linguistics.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does Neural Machine Translation Benefit from Larger Context? *arXiv e-prints*, page arXiv:1704.05135.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

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

Shaohui Kuang and Deyi Xiong. 2018. Fusing recency into neural machine translation with an inter-sentence gate model. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 607–617, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Shaohui Kuang, Deyi Xiong, Weihua Luo, and Guodong Zhou. 2018. Modeling coherence for neural machine translation with dynamic and topic caches. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 596–606, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Michael Lebowitz. 1983. Memory-based parsing. *Artificial Intelligence*, 21(4):363–404.

Jindřich Libovický and Jindřich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 196–202, Vancouver, Canada. Association for Computational Linguistics.

Pierre Lison and Jrg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portoro, Slovenia. European Language Resources Association (ELRA).

Sameen Maruf and Gholamreza Haffari. 2018. Document context neural machine translation with memory networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1275–1284, Melbourne, Australia. Association for Computational Linguistics.

Paul Michel and Graham Neubig. 2018. Extreme adaptation for personalized neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 312–318, Melbourne, Australia. Association for Computational Linguistics.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. 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, Brussels, Belgium. Association for Computational Linguistics.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*.

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

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

Royal Skousen. 1989. *Analogical modeling of language*. Springer Science & Business Media.

Royal Skousen. 2013. *Analogy and structure*. Springer Science & Business Media.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

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

Zhaopeng Tu, Yang Liu, Zhengdong Lu, Xiaohua Liu, and Hang Li. 2017. Context gates for neural machine translation. *Transactions of the Association for Computational Linguistics*, 5:87–99.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. 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, Melbourne, Australia. Association for Computational Linguistics.

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

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. *arXiv preprint arXiv:1410.3916*.

Jiacheng Zhang, Yanzhuo Ding, Shiqi Shen, Yong Cheng, Maosong Sun, Huan-Bo Luan, and Yang Liu. 2017. THUMT: an open source toolkit for neural machine translation. *CoRR*, abs/1706.06415.

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