Document-level Neural Machine Translation with Document Embeddings

Shu Jiang[1,2,3], **Hai Zhao**[1,2,3] *, **Zuchao Li**[1,2,3], **Bao-Liang Lu**[1,2,3]

[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] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai, China
jshmjs45@gmail.com, zhaohai@cs.sjtu.edu.cn, charlee@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 are satisfied with a smattering sense of brief document-level information, while this work focuses on exploiting detailed document-level context in terms of multiple forms of document embeddings, which is capable of sufficiently modeling deeper and richer document-level context. The proposed document-aware NMT is implemented to enhance the Transformer baseline by introducing both global and local document-level clues on the source end. Experiments show that the proposed method significantly improves the translation performance over strong baselines and other related studies.

Introduction

Neural Machine Translation (NMT) 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 (Kalchbrenner and Blunsom 2013; Sutskever, Vinyals, and Le 2014; Cho et al. 2014; Bahdanau, Cho, and Bengio 2015; Vaswani et al. 2017). So far, despite the big advance in model architecture, most models keep taking a standard assumption to translate every sentence independently, ignoring the implicit or explicit sentence correlation from document-level contextual clues during translation.

However, document-level information has shown helpful in improving the translation performance from multiple as-

pects: consistency, disambiguation, and coherence (Kuang et al. 2018). If translating every sentence is completely independent of document-level context, it will be difficult to keep every sentence translations across the entire document consistent with each other. Moreover, even sentence independent translation may still benefit from document-level clues through effectively disambiguating words by referring to multiple sentence contexts. At last, document-level clues as a kind of global information across the entire text may effectively help generate more coherent translation results compared to the way only adopting local information inside a sentence alone.

There have been few recent attempts to introduce the document-level information into the existing standard NMT models. Various existing methods (Jean et al. 2017; Tiedemann and Scherrer 2017; Wang et al. 2017; Voita et al. 2018; Kuang and Xiong 2018; Maruf and Haffari 2018; Jiang et al. 2019; Li, Jiang, and Liu 2019; Kim, Tran, and Ney 2019; Rysov et al. 2019) focus on modeling the context from the surrounding text in addition to the source sentence.

For the more high-level context, 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 inter-sentence connection. Tan et al. (2019) integrate their proposed Hierarchical Modeling of Global Document Context model (HM-GDC) into the original Transformer model to improve the document-level translation.

However, most of the existing document-level NMT methods focus on introducing the information of disambiguating global document or the surrounding sentences but fail to comprehend the relationship among the current sentence, the global document information, and the local document information, let alone the refined global document-level clues.

In this way, our proposed model can focus on the most relevant part of the concerned translation from which exactly encodes the related 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 tasks.

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.

Mining Document-level Information

Tiedemann and Scherrer (2017) propose to simply extend the context during the NMT model training by *concatenation* method.

Wang et al. (2017) obtain all sentence-level representations after processing each sentence by *Document RNN*. The last hidden state represents the summary of the whole sentence, as well as the summary of the global context is represented by the last hidden state over the sequence of the above sentence-level representations.

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

Macé and Servan (2019) present a *Word Embedding Average* method to add source context that capture the whole document with accurate boundaries, taking every word into account by an averaging method.

Integrating Document-level Information

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.

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.

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.

Tu et al. (2018) propose to augment NMT models by *cache-based neural model* 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.

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. This *context-aware Transformer 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.

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.

Tan et al. (2019) propose a global document context model to improve the document-level translation, which is hierarchically extracted from the entire global text with a sentence encoder to model intra-sentence information and a document encoder to model document-level inter-sentence context representation.

Ma, Zhang, and Zhou (2020) propose a *Flat-Transformer model* with a simple and effective unified encoder that model the bi-directional relationship between the contexts and the source sentences.

Chen et al. (2020) propose to improve document-level NMT by the means of discourse structure information, and the encoder is based on a HAN Miculicich et al. (2018). They parse the document to obtain its discourse structure, then introduce a Transformer-based path encoder to embed the discourse structure information of each word and combine the discourse structure information with the word embedding.

Most of the previous works only focus on the context embedding or considering the global text, but our work is able to mine the relationship among input sentences, the whole document, and context sentences like the previous sentences.

Background

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:

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

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

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

where $g(\cdot)$ is a nonlinear function that outputs the probability of previously generated word y_i, and $\mathbf{c}_i$ is the i-th source representation. Then i-th decoding hidden state $\mathbf{s}_i$ is computed as

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

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.
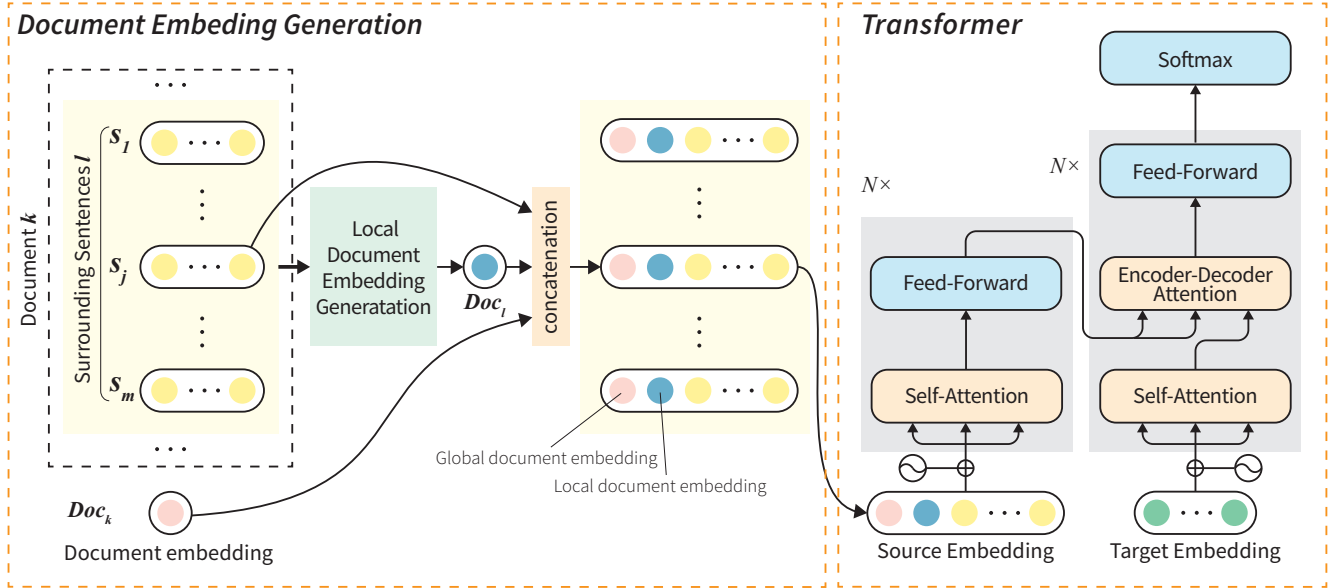
Figure 1: The framework of our model.

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.

The encoder is composed of a stack of N (usually equal 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 in the Transformer allows the model to jointly process information from different representation spaces at distinct positions. It linearly projects the queries Q, keys K, and values V h times 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:

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

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.

Model

As shown in Figure 1, we introduce document-level clues into the NMT training through an embedding method, two types of document embeddings (namely, *global* and *local*) are directly concatenated into the source embedding during training.

Document Embedding Generation

For a document $k = \{\mathbf{x}_1, ..., \mathbf{x}_j, ..., \mathbf{x}_{m_k}\}$, the sentence in this document is represented $\mathbf{x}_j = \{x_1, ..., x_i, ..., x_{n_j}\}$, the word vector in the sentence is denoted as x_i.

The *global* document embedding is obviously generated from the whole document k in the corpus with fine-defined document boundaries. While generating the *local* document embedding, we consider the surrounding sentences $\{\mathbf{x}_j\}_{j=1}^m$ of the current sentence $\mathbf{x}_j$ as the document l.

In this paper, we consider the following methods to obtain the document embedding.

Word Embedding Average   According to Macé and Servan (2019), we consider the *document embedding* of a document k by averaging all N word vectors x in this document and therefore has the same dimension.

$$\text{Doc}_k = \frac{1}{N} \sum_{i=1}^{N} x_{i,k} \qquad (5)$$

Document RNN   Inspired by Wang et al. (2017) which proposes a cross-sentence context-aware RNN approach to produce the context representation. Firstly, the sentence RNN reads the corresponding words $x_{i,j}$ in the sentence $\mathbf{x}_j$ sequentially and updates its hidden state by

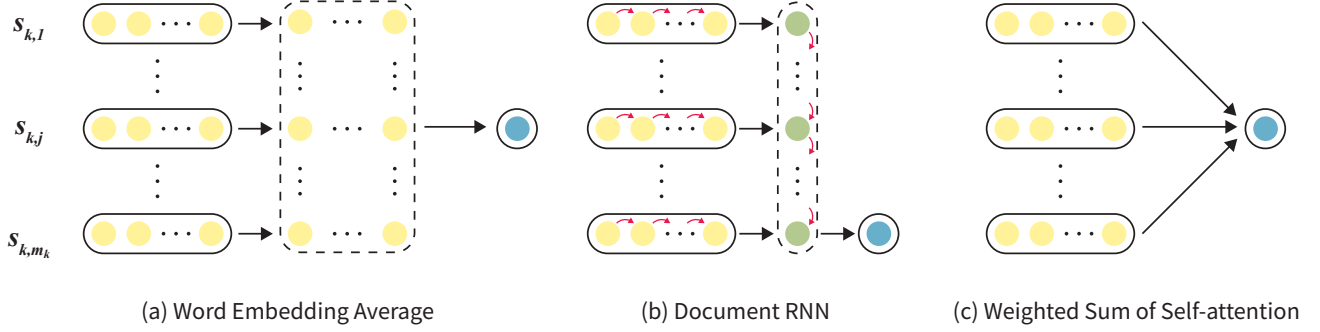$$h_{i,j} = f(h_{i-1,j}, x_{i,j}) \qquad (6)$$

| (a) Word Embedding Average | (b) Document RNN | (c) Weighted Sum of Self-attention |

Figure 2: Three methods for calculating the *document embedding*.

Data	Zh-En				En-Fr				En-De			
	#Doc	#Sent	#Src	#Tgt	#Doc	#Sent	#Src	#Tgt	#Doc	#Sent	#Src	#Tgt
train	1976	0.2M	4.8M	5.4M	1914	0.2M	5.5M	5.9M	1705	0.2M	4.9M	5.0M
dev	8	0.9K	23.6K	23.4K	8	0.9K	23.6K	24.3K	8	0.9K	23.7K	24.6K
tst10	11	1.6K	35.3K	36.0K	11	1.6K	36.3K	36.3K	11	1.6K	36.5K	37.7K
tst11	16	1.4K	27.0K	30.9K	16	1.4K	31.0K	33.5K	16	1.4K	30.9K	32.7K
tst12	15	1.7K	30.4K	34.9K	15	1.7K	35.2K	38.7K	15	1.7K	35.1K	36.8K
tst13	20	1.4K	31.5K	34.1K	20	1.4K	34.3K	37.8K	16	1.0K	24.3K	24.7K
tst14	15	1.3K	26.5K	29.5K	15	1.3K	29.7K	33.0K	15	1.3K	29.8K	31.0K
tst15	12	1.2K	25.2K	27.5K	12	1.2K	27.7K	29.6K	12	1.1K	24.1K	24.8K
tst-all	62	5.6K	0.2M	0.2M	89	8.6K	0.2M	0.2M	85	8.1K	0.2M	0.2M

Table 1: Detail of training, development and test sets.

where $f(\cdot)$ is an activation function, and $h_{i,j}$ is the hidden state at time i. The last state $h_{n_j,j}$ represents the summary of the whole sentence $\mathbf{x}_j$ and denoted as Sent_j, i.e. $h_{n_j,j} \equiv \mathrm{Sent}_j$.

Then, all sentence-level representations are fed into document RNN as follows:

$$H_{j,k} = f(H_{j-1,k}, \mathrm{Sent}_{j,k}) \tag{7}$$

where $h_{j,k}$ is the recurrent state at time j in the document k. Similarly, we use the last hidden state to indicate the summary of the global document, i.e. $H_{j,k} \equiv \mathrm{Doc}_k$.

Weighted Sum of Self-attention The input sentence $\mathbf{x}_j$ first goes through a multi-head attention layer to encode the contextualized information to each word representation:

$$\mathbf{X}_j^{(n)} = \mathrm{MultiHead}(\mathbf{X}_j^{(n-1)}, \mathbf{X}_j^{(n-1)}, \mathbf{X}_j^{(n-1)}), \tag{8}$$

where $\mathbf{X}_j^{(0)} = \mathbf{x}_j$. Each word representation is as a vector $s \in \mathbb{R}^d$, where d is the size of hidden state in MultiHead function.

Then we compute the compatibility function using a feed-forward network FFN with a single hidden layer:

$$\mathbf{X}_j^{(n)} = [\mathrm{FNN}(\mathbf{X}_{1,j}^{(n)}); ...; \mathrm{FNN}(\mathbf{X}_{n,j}^{(n)})] \tag{9}$$

where $\mathbf{X}_j^{(n)} \in \mathbb{R}^{i \times d}$ is the representation of the source sentence $\mathbf{x}_j$ at the n-th layer ($n = 1, .., N$). We treat the output

at the last layer as the representation of the input sentence $\mathbf{x}_j$ i.e. $\mathbf{X}_j^{(N)} \equiv \mathrm{Sent}_j$.

Finally, we calculated the *document embedding* by acquiring the weighted sum of all sentence embeddings in the document k:

$$\mathrm{Doc}_k = \sum_{j=1}^{m_k} \alpha_j \mathrm{Sent}_{j,k}. \tag{10}$$

where α_j is the weight of each sentence embedding $\mathrm{Sent}_{j,k}$ trained by the model.

Document Embedding Integration

At first, we train a baseline Transformer model (noted *Baseline* model) on a standard corpus that does not contain any document-level information and extract word embeddings from it. Then, we train an enhanced model (noted *Enhanced* model) benefiting from these extracted word embeddings. This process can be treated as *pre-training* of word embeddings. The *Enhanced* model directly adopts the pre-trained embeddings from the *Baseline* model, namely, word embeddings will be fixed during its model training.

In our model, we calculated the *global* document embedding Doc_k for each document k using *Word Embedding Average* method. It should be noted that the enhanced model does not fine-tune its embeddings to preserve the relationship between words and document vectors during training.

Language	Model	tst-all	tst10	tst11	tst12	tst13	tst14	tst15
Zh-En	RNN Search*	16.80	12.96	17.40	15.39	15.89	13.15	15.85
	Baseline	19.50	16.44	21.24	18.93	20.05	17.99	21.04
	Enhanced	19.98	16.80	21.56	19.31	20.67	**18.41**	21.75
	Enhanced+global	20.13	16.63	21.55	19.16	20.88	18.02	21.79
	Enhanced+global+local	**20.18**	**16.84**	**21.68**	**19.36**	**20.90**	18.38	**21.89**
En-Fr	RNN Search*	37.13	31.76	38.18	36.91	35.59	33.28	32.38
	Baseline	39.92	36.03	43.07	42.83	40.34	38.10	38.70
	Enhanced	39.98	36.47	43.51	43.07	41.23	38.35	38.85
	Enhanced+global	41.25	**37.57**	43.97	44.49	**42.34**	39.28	39.17
	Enhanced+global+local	**41.46**	37.30	**44.48**	**44.71**	42.25	**39.44**	**39.35**
En-De	RNN Search*	25.28	23.33	25.46	22.18	24.68	20.56	22.59
	Baseline	27.94	28.12	30.41	26.67	29.11	25.28	27.24
	Enhanced	28.46	28.39	30.05	28.11	30.33	26.39	28.59
	Enhanced+global	29.03	**29.56**	**31.21**	27.93	30.72	26.21	28.33
	Enhanced+global+local	**29.21**	29.44	31.09	**28.48**	**30.99**	**26.76**	**28.92**

Table 2: Performance (BLEU scores) comparison on the different datasets. "Enhanced" indicates the *Enhanced* model defined in Section benefiting from the word embeddings extracted from the *Baseline* Model. The *global* and *local* embeddings are generated by "Word Embedding Average" method and "Document RNN" method respectively. The proposed methods were significantly better than the baseline Transformer at significance level p-value<0.05 . The scores in bold indicate the best ones on the same dataset.

Corpus	Zh-En IWSLT2015	Zh-En IWSLT2017	En-De IWSLT2017
HAN (Miculicich et al. 2018)	17.79	-	-
HM-GDC (Tan et al. 2019)	-	17.63	-
Flat-Transformer (Ma, Zhang, and Zhou 2020)	-	-	26.61
Transformer+HAN+DS (Chen et al. 2020)	-	-	24.84
Our model(*Enhanced*+global+local (avg+rnn))	**19.40**	**20.61**	**29.21**

Table 3: Comparison with the related works.

Because of the limitation of GPU memory, it is impractical to feed all the word vectors in the document k into the Model and calculate the *global* document embedding using *Document RNN* and *Weighted Sum of Self-attention* method, which needs train the hidden variables to generate the document embedding. On the basis of this restriction, we generate the *local* document embedding from the *surrounding sentences* $l = \{\mathbf{x}_j\}_{j=1}^{m}$ during training and denote it as Doc_l. In practice, we using the input sentences in a mini-batch as the *surrounding sentences*. Thus,the source embedding $\mathbf{s}$ for input sentence $\mathbf{x}$ can be represented as follows:

$$\mathbf{s} = \{\text{Doc}_k, \text{Doc}_l, x_1, x_i, ..., x_n\} \quad (11)$$

In the case that it needs to ensemble N document embeddings, for example to ensemble "Document RNN" and "Weighted Sum of Self-attention", we calculate the weighed sum of them:

$$\text{Doc} = \sum_{i=1}^{N} \beta_i \text{Doc}_i \quad (12)$$

where β_i is the weight of each document embedding Doc_i trained by the model.

Experiments

Setup

Data   As we focus on document-level NMT, it poses a document-annotation requirement on the evaluation dataset that needs for well defined document boundaries marking each sentence with its global document tag. Thus we train and evaluate our model on the corpus from the TED Talks on three language pairs, i.e., Chinese-to-English (Zh-En), English-to-French(En-Fr), and English-to-German(En-De). The TED talk documents are the parts of the IWSLT2017 Evaluation Campaign Machine Translation task[1]. We use *dev2010* as the development set and combine the *tst2010-2015* as the test set. The statistics of the corpora are listed in Table 1.

Data preprocessing   The English and Spanish datasets are tokenized by *tokenizer.perl* and truecased by *truecase.perl* provided by MOSES[2], a statistical machine translation system proposed by Koehn et al. (2007). The Chinese corpus

---

[1]https://wit3.fbk.eu/mt.php?release=2017-01-trnted
[2]https://github.com/moses-smt/mosesdecoder

Enhanced+	Global	Local	tst-all	tst10	tst11	tst12	tst13	tst14	tst15
	/	/	19.98	16.80	21.56	19.31	20.67	**18.41**	21.75
	avg	/	20.13	16.63	21.55	19.16	20.88	18.02	21.79
	/	avg	19.96	16.91	21.50	19.08	20.89	18.12	21.75
	/	rnn	19.87	16.69	21.65	19.36	20.68	18.23	21.62
Zh-En	/	attn	20.09	16.70	21.54	19.22	20.76	18.20	21.76
	avg	avg	20.12	16.73	21.57	19.25	20.79	18.23	21.78
	avg	rnn	20.18	16.84	21.68	19.36	**20.90**	18.34	**21.89**
	avg	attn	20.07	16.68	21.52	19.20	20.74	18.18	21.74
	avg	rnn+attn	**20.23**	**16.93**	**21.70**	**19.42**	20.86	18.04	21.76
	/	/	39.98	36.47	43.51	43.07	41.23	38.35	38.85
	avg	/	41.25	**37.57**	43.97	44.49	42.34	39.28	39.17
	/	avg	41.32	37.18	44.13	44.63	**42.39**	39.19	39.22
	/	rnn	40.68	37.09	43.81	43.85	41.85	38.88	39.08
En-Fr	/	attn	40.58	36.98	43.70	43.74	41.75	38.78	38.97
	avg	avg	41.01	36.85	44.13	44.42	41.83	38.59	38.74
	avg	rnn	**41.46**	37.30	**44.48**	**44.71**	42.25	**39.44**	**39.35**
	avg	attn	40.89	37.36	43.84	43.77	41.31	39.00	39.08
	avg	rnn+attn	40.95	37.35	44.07	44.11	42.12	39.15	39.34
	/	/	28.46	28.39	30.05	28.11	30.33	26.39	28.59
	avg	/	29.03	29.56	31.21	27.93	30.72	26.21	28.33
	/	avg	28.92	28.88	30.46	27.47	**31.75**	26.48	28.69
	/	rnn	29.18	**29.77**	**31.42**	28.11	30.66	26.20	27.86
En-De	/	attn	28.46	28.69	30.35	27.74	30.24	26.02	28.18
	avg	avg	28.91	29.14	30.79	28.18	30.69	26.46	28.62
	avg	rnn	**29.21**	29.44	31.09	**28.48**	30.99	**26.76**	**28.92**
	avg	attn	28.47	28.70	30.36	27.75	30.25	26.03	28.19
	avg	rnn+attn	28.73	28.96	30.61	28.00	30.51	26.28	28.44

Table 4: Ablation study on the *Enhanced* model with different document embeddings. The tag *avg*, *rnn* and *attn* mean that the document embedding is generated by "Word Embedding Average" method, "Document RNN" and "Weighted Sum of Self-attention" method.

is tokenized by *Jieba* Chinese text segmentation[3]. Words in sentences are segmented into subwords by Byte-Pair Encoding (BPE) Sennrich, Haddow, and Birch (2016) with 32k BPE operations.

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)[4]. 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. All hidden states have 512 dimensions, 8 heads for multi-head attention, and the training batch contains about 6,520 source tokens. We use the original regularization and optimizer in Transformer (Vaswani et al. 2017). Finally, we evaluate the performance of the model by BLEU score (Papineni et al. 2002) using *multi-bleu.perl* on the *tokenized* text.

[3]https://github.com/fxsjy/jieba
[4]https://github.com/thumt/THUMT

Translation Performance

Table 2 demonstrates the BLEU scores for different models on multiple corpora. The *RNN* model is a re-implemented attention-based NMT system RNNSearch* (Hinton et al. 2012) and Transformer (Vaswani et al. 2017) using THUMT kit. The *Baseline* Model is also the *pre-trained* model mentioned in the Section .

The results in Table 2 demonstrate that our model is significantly better than the baseline Transformer at significance level p-value<0.05. The *global* embedding (generated by the "Word Embedding Average" method) and the *local* embedding (generated by "Document RNN" method) in our model can effectively exploit the document-level information from the global text and the surrounding sentences and improve the performance of the *Enhanced* model.

The *Enhanced* model trained with the embedding form the *Baseline* model outperforms the *Baseline* model by 0.48 BLEU point on the Zh-En dataset, 0.06 BLEU point on the En-Fr dataset, and 0.52 BLEU point on the En-De dataset. When we add the *global* document embedding to the *Enhanced* model, the *Enhanced+global* model, surpassed the *Baseline* model by 0.73 BLEU points and Zh-En, 1.40 BLEU points on En-Fr and 1.24 BLEU points on En-

Context sentences	*Zhǐdǎo shnggg shj 80 nindi, zhge ychng sh gntǐng rn gunxi de.* *(until the 1980s, the farm was in the hands of the Argentinians.)* *Tmen zi zhǐ yng ni ng shǐthu zhl jbn shng sh shd.* *(they raised beef cattle on what was essentially wetlands.)*
Source sentence	*Dngshǐ tmen b shu chu zu.*
Reference sentence	*they did it by draining the land.*
Transformer model	*and then they take the water off.*
Our model	*and then they pulled the water out.*

Table 5: The first example of the translation result. The context sentences are the previous 2 sentences before the source sentence and words in red from context indicate the heuristic clues for better translation. The Chinese sentences are converted to Pinyin version and the English translation have been provided.

Context sentences	*Du wzhng fngf zhng de mi yzhng, wmen du xyo zhsho 100 rn de tundu.* *(in each of these five paths, we need at least a hundred people.)* *Lmin de hndu rn, n hu jud tmen hn fngkung, zh ji dule.* *(and a lot of them, you'll look at and say, "They're crazy ." that's good.)*
Source sentence	*W rnwi, zi TED tundu l yjng yu hndu rn kish zhl y c.*
Reference sentence	*and, I think, here in the TED group, we have many people who are already pursuing this.*
Transformer model	*I think there are so many people in the TED community that are working on this.*
Our model	*and I think theres a lot of people in the TED community who have been working on this.*

Table 6: The second example of the translation result. The context sentences are the previous 2 sentences before the source sentence. The Chinese sentences are converted to Pinyin version and the English translation have been provided.

De. Moreover, after taking the *local* document embedding into account, the *Enhanced+global+local* model achieves the gains of 1.11 BLEU point, 1.54 BLEU point, and 1.27 BLEU point on these three datasets individually over the *Baseline* model.

Comparison with the related work

We also compared our proposed method on the corpus mentioned in the Hierarchical Attention Networks (HAN) NMT (Miculicich et al. 2018) model, Hierarchical Modeling of Global Document Context methods (HM-GDC) (Tan et al. 2019), Flat-Transformer (Ma, Zhang, and Zhou 2020), and document-level NMT based on a HAN with discourse structure information (Transformer+HAN+DS) modelChen et al. (2020) and the results in Table 3 show that our model significantly outperforms the related work.

Ablation Study

We investigate the impact of different document embedding methods by removing one or more of them. As shown in Table 4, all of the components greatly contribute to the performance of our proposed model. If we remove any document embedding in the *Enhanced* model, the performance drops dramatically. Such results indicate that both the *global* document embedding and the *local* document embedding play an important and complementary role in our model. For the *local* document embedding, we compare different document embedding generation methods and find out that the "Document RNN" method has great effects on *Enhanced* model.

Translation Quality

We also provide examples to illustrate what do these document embeddings capture. Table 5 shows the first example, which is extracted from line 111 of TED Talks (Zh-En). The source sentence does not involve any information to indicate the time status, but the context sentences mention the time information "*shngg shj 80 nindi*" (which means "the 1980s" in English). Thus our model can recognize the past tense of the source sentence exactly. Table 6 demonstrates the translation example form line 549 of TED Talks (Zh-En) as the second example. Although the source sentence does not contain any word to represent the discourse relationship with previous contexts, our model is able to infer discourse relationship and add the connection word to make the translation more fluent.

Conclusion

In this paper, we explore more comprehensive document-level neural machine translation. Assuming that document-level clues are indeed helpful for better translation, it is kept an open problem for finding a good way to effectively introduce such helpful clues into sentence-independent NMT. Taking document embedding as our default representation for document-level clues, we distinguish two types of document embeddings, the global and the local, which targetedly capture both the general information in the whole document scope and the specific detailed information in the surrounding text. For the concerned document-level NMT, we for the first time survey multiple ways for generating document embeddings and conduct extensive experiments. Taking a

strong Transformer baseline, our experimental results show that our global and local document embeddings may effectively enhance the baseline systems, showing that more sufficient and richer document clues indeed greatly help standard sentence-independent NMT.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, 1–15. San Diego, USA. URL https://arxiv.org/pdf/1409.0473v7.pdf.

Chen, J.; Li, X.; Zhang, J.; Zhou, C.; Cui, J.; Wang, B.; and Su, J. 2020. Modeling Discourse Structure for Document-level Neural Machine Translation. In *Proceedings of the First Workshop on Automatic Simultaneous Translation*, 30–36. Seattle, Washington: Association for Computational Linguistics. doi:10.18653/v1/2020.autosimtrans-1.5. URL https://www.aclweb.org/anthology/2020.autosimtrans-1.5.

Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 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)*, 1724–1734. Doha, Qatar: Association for Computational Linguistics. doi:10.3115/v1/D14-1179. URL http://aclweb.org/anthology/D14-1179.

Hinton, G. E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. R. 2012. Improving neural networks by preventing co-adaptation of feature detectors.

Jean, S.; Lauly, S.; Firat, O.; and Cho, K. 2017. Does Neural Machine Translation Benefit from Larger Context? *arXiv e-prints* arXiv:1704.05135.

Jiang, S.; Wang, R.; Li, Z.; Utiyama, M.; Chen, K.; Sumita, E.; Zhao, H.; and liang Lu, B. 2019. Document-level Neural Machine Translation with Inter-Sentence Attention. *arXiv: 1910.14528* .

Kalchbrenner, N.; and Blunsom, P. 2013. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1700–1709. Seattle, Washington, USA: Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D13-1176.

Kim, Y.; Tran, D. T.; and Ney, H. 2019. When and Why is Document-level Context Useful in Neural Machine Translation? In *DiscoMT*.

Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; Dyer, C.; Bojar, O.; Constantin, A.; and Herbst, E. 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*, 177–180. Prague, Czech Republic: Association for Computational Linguistics. URL http://aclweb.org/anthology/P07-2045.

Kuang, S.; and Xiong, D. 2018. Fusing Recency into Neural Machine Translation with an Inter-Sentence Gate Model. In *Proceedings of the 27th International Conference on Computational Linguistics*, 607–617. Santa Fe, New Mexico, USA: Association for Computational Linguistics. URL http://aclweb.org/anthology/C18-1051.

Kuang, S.; Xiong, D.; Luo, W.; and Zhou, G. 2018. Modeling Coherence for Neural Machine Translation with Dynamic and Topic Caches. In *Proceedings of the 27th International Conference on Computational Linguistics*, 596–606. Santa Fe, New Mexico, USA: Association for Computational Linguistics. URL http://aclweb.org/anthology/C18-1050.

Li, L.; Jiang, X.; and Liu, Q. 2019. Pretrained Language Models for Document-Level Neural Machine Translation. *arXiv: 1911.03110* .

Ma, S.; Zhang, D.; and Zhou, M. 2020. A Simple and Effective Unified Encoder for Document-Level Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 3505–3511. Online: Association for Computational Linguistics. doi:10.18653/v1/2020.acl-main.321. URL https://www.aclweb.org/anthology/2020.acl-main.321.

Macé, V.; and Servan, C. 2019. Using Whole Document Context in Neural Machine Translation. *ArXiv* abs/1910.07481.

Maruf, S.; and Haffari, G. 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)*, 1275–1284. Melbourne, Australia: Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P18-1118.

Michel, P.; and Neubig, G. 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)*, 312–318. Melbourne, Australia: Association for Computational Linguistics. URL http://aclweb.org/anthology/P18-2050.

Miculicich, L.; Ram, D.; Pappas, N.; and Henderson, J. 2018. Document-Level Neural Machine Translation with Hierarchical Attention Networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2947–2954. Brussels, Belgium: Association for Computational Linguistics. URL http://aclweb.org/anthology/D18-1325.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. URL http://aclweb.org/anthology/P02-1040.

Rysov, K.; Rysov, M.; Musil, T.; Polkov, L.; and Bojar, O. 2019. A Test Suite and Manual Evaluation of Document-Level NMT at WMT19. In *WMT*.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural Machine Translation of Rare Words with Subword Units.

In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1715–1725. Berlin, Germany: Association for Computational Linguistics. doi:10.18653/v1/P16-1162. URL http://aclweb.org/anthology/P16-1162.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N. D.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 27*, 3104–3112. Curran Associates, Inc. URL http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

Tan, X.; Zhang, L.; Xiong, D.; and Zhou, G. 2019. Hierarchical Modeling of Global Context for Document-Level Neural Machine Translation. In *EMNLP-IJCNLP*. doi:10.18653/v1/D19-1168. URL https://www.aclweb.org/anthology/D19-1168.

Tiedemann, J.; and Scherrer, Y. 2017. Neural Machine Translation with Extended Context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, 82–92. Copenhagen, Denmark: Association for Computational Linguistics. URL http://aclweb.org/anthology/W17-4811.

Tu, Z.; Liu, Y.; Lu, Z.; Liu, X.; and Li, H. 2017. Context Gates for Neural Machine Translation. *Transactions of the Association for Computational Linguistics* 5: 87–99. URL http://aclweb.org/anthology/Q17-1007.

Tu, Z.; Liu, Y.; Shi, S.; and Zhang, T. 2018. Learning to Remember Translation History with a Continuous Cache. *Transactions of the Association for Computational Linguistics* 6: 407–420. doi:10.1162/tacl_a_00029. URL https://www.aclweb.org/anthology/Q18-1029.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*, 5998–6008. Curran Associates, Inc. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Voita, E.; Serdyukov, P.; Sennrich, R.; and Titov, I. 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)*, 1264–1274. Melbourne, Australia: Association for Computational Linguistics. URL http://aclweb.org/anthology/P18-1117.

Wang, L.; Tu, Z.; Way, A.; and Liu, Q. 2017. Exploiting Cross-Sentence Context for Neural Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2826–2831. Copenhagen, Denmark: Association for Computational Linguistics. URL http://aclweb.org/anthology/D17-1301.

Zhang, J.; Ding, Y.; Shen, S.; Cheng, Y.; Sun, M.; Luan, H.; and Liu, Y. 2017. THUMT: An Open Source Toolkit for Neural Machine Translation. *CoRR* abs/1706.06415. URL http://arxiv.org/abs/1706.06415.

Zhang, J.; Luan, H.; Sun, M.; Zhai, F.; Xu, J.; Zhang, M.; and Liu, Y. 2018. Improving the Transformer Translation Model with Document-Level Context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 533–542. Brussels, Belgium: Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1049.