Document Graph for Neural Machine Translation

Mingzhou Xu,[1] **Liangyou Li,** [2] **Derek. F. Wai,** [1] **Qun Liu,** [2] **Lidia S. Chao** [1]
[1] NLP2CT Lab, Department of Computer and Information Science, University of Macau, China
[2] Huawei Noah's Ark Lab, Hong Kong, China
[1]nlp2ct.mzxu@gmail.com, [2]{liliangyou,qun.liu}@huawei.com, [1]{derekfw,lidiasc}@um.edu.mo

Abstract

Previous works have shown that contextual information can improve the performance of neural machine translation (NMT). However, most existing document-level NMT methods failed to leverage contexts beyond a few set of previous sentences. How to make use of the whole document as global contexts is still a challenge. To address this issue, we hypothesize that a document can be represented as a graph that connects relevant contexts regardless of their distances. We employ several types of relations, including adjacency, syntactic dependency, lexical consistency, and coreference, to construct the document graph. Then, we incorporate both source and target graphs into the conventional Transformer architecture with graph convolutional networks. Experiments on various NMT benchmarks, including IWSLT English–French, Chinese-English, WMT English–German and Opensubtitle English–Russian, demonstrate that using document graphs can significantly improve the translation quality.

1 Introduction

Although neural machine translation (NMT) has achieved great success on sentence-level translation tasks, many studies pointed out that translation mistakes become more noticeable at the document-level (Wang et al. 2017; Tiedemann and Scherrer 2017; Zhang et al. 2018; Miculicich et al. 2018; Kuang et al. 2018; Voita et al. 2018; Läubli, Sennrich, and Volk 2018; Tu et al. 2018; Voita, Sennrich, and Titov 2019b; Kim, Tran, and Ney 2019; Yang et al. 2019). They proved that these mistakes can be alleviated by feeding the inter-sentential contexts into context-agnostic NMT models.

Previous works have explored various methods to integrate context information into NMT models. They usually take a limited number of previous sentences as contexts and learn context-aware representations using hierarchical networks (Miculicich et al. 2018; Wang et al. 2017; Tan et al. 2019) or extra context encoders (Jean et al. 2017; Zhang et al. 2018; Yang et al. 2019). Different from representation-based approaches, Tu et al. (2018) and Kuang et al. (2018) propose using a cache to memorize context information, which can be either history hidden states or lexicons. To keep tracking of most recent contexts, the cache is usually updated when new translations are generated. Therefore, long-distance contexts would likely to be erased.

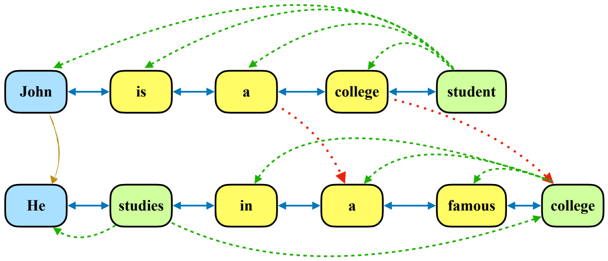


Figure 1: The diagram Illustrates the architecture of the proposed model. Solid lines in blue depict adjacency relations. Dash lines in green denote dependency relations. Lexical consistency is represented as dash-dotted lines in red. The brown line means a coreference relation. [1]

How to use long-distance contexts is drawing attention in recent years. Approaches, like treating the whole document as a long sentence (Junczys-Dowmunt 2019) and using memory and hierarchical structures (Maruf and Haffari 2018; Maruf, Martins, and Haffari 2019; Tan et al. 2019), are proposed to take global contexts into consideration. However, Kim, Tran, and Ney (2019) point out that not all the words in a document are beneficial to context integration, suggesting that it is essential for each word to focus on its own relevant context.

To address this problem, we suppose to build a document graph for a document, where each word is connected to those words which have a direct influence on its translation. Figure 1 shows an example of a document graph. Explicitly, a document graph is defined as a directed graph where: (1) each node represents a word in the document; (2) each edge represents one of the following relations between words: (a) adjacency; (b) syntactic dependency; (c) lexical consistency; or (d) coreference.

We apply a Graph Convolutional Network (GCN) on the document graph to obtain a document-level contextual representation for each word, fed to the conventional TRANSFORMER model (Vaswani et al. 2017) by additional atten-

[1] Dependency and coreference relations are from Stanford CoreNLP (https://corenlp.run/).

tion and gating mechanisms. We evaluate our model on four translation benchmarks, IWSLT English–French (En–Fr) and Chinese–English (Zh–En), Opensubtitle English–Russian (En–Ru), and WMT English–German (En–De). Experimental results demonstrate that our approach is consistently superior to previous works (Miculicich et al. 2018; Tu et al. 2018; Zhang et al. 2018; Junczys-Dowmunt 2019; Tan et al. 2019; Maruf, Martins, and Haffari 2019) on all the language pairs.

The contributions of this work are summarized as:

- We represent a document as a graph that connects relevant contexts regardless of their distances. To the best of our knowledge, this is the first work to introduce such graphs into document-level neural machine translation.
- We investigate several relations between words to construct document graphs and verify their effectiveness in experiments.
- We propose a graph encoder to learning graph representations based on GCN layers with an attention mechanism to combine representations of different sources.
- We propose a model architecture that integrates context representations into the conventional TRANSFORMER model via attention and gating mechanisms.

2 Approach

In this section, we introduce the proposed document graph and model for leveraging contextual information from documents. Firstly, we present a definition of the problem. Then, we describe the model architecture we use to integrate document graphs. Finally, the construction and representation learning of document graphs are explained in Section 2.3 and Section 2.4, respectively.

2.1 Problem Definition

Document-level NMT learns to translate from a document in a source language to a document in a target language. Formally, a source document is a set of M sentences $\mathbf{X} = [X^1, ..., X^m, ..., X^M]$, where $X^m = [x_1^m, ..., x_i^m, ..., x_{I_m}^m]$ indicates the mth sentence of the document. The corresponding target document is $\mathbf{Y} = [Y^1, ..., Y^m, ..., Y^M]$, where $Y^m = [y_1^m, ..., y_j^m, ..., y_{J_m}^m]$ is a translation of the source sentence X^m.

Given the source document to translate, we assume is a pair of source and target hidden graphs $G_{\mathbf{X},\hat{\mathbf{Y}}} = \langle G_{\mathbf{X}}, G_{\hat{\mathbf{Y}}} \rangle$ (called document graphs and defined in Section 2.3) to help generate the target document. Therefore, the translation probability from $\mathbf{X}$ to $\mathbf{Y}$ can be represented as:

$$\begin{aligned} &P(\mathbf{Y}|\mathbf{X}) \\ &= \sum_{G_{\mathbf{X},\hat{\mathbf{Y}}}} P(\mathbf{Y}|\mathbf{X}, G_{\mathbf{X},\hat{\mathbf{Y}}}) P(G_{\mathbf{X},\hat{\mathbf{Y}}}|\mathbf{X}) \qquad (1) \\ &\approx P(\mathbf{Y}|\mathbf{X}, G_{\mathbf{X},\hat{\mathbf{Y}}}) \qquad (2) \end{aligned}$$

Equation (1) is computationally intractable. Therefore, instead of considering all possible graph pairs, we only sample one pair of graphs according to the source document resulting in a simplified Equation (2). In this paper, we construct the source graph $G_{\mathbf{X}}$ directly from the source document (Section 2.3). In order to obtain an informative target graph (for both training and inference steps), we first translate the source document using a context-agnostic NMT system and then construct the target graph $G_{\hat{\mathbf{Y}}}$ from these translations(refer to Supplementary for details).

The translation of a document is further decomposed into translations of each sentence with document graphs as context:

$$P(\mathbf{Y}|\mathbf{X}) \approx \prod_{m=1}^{M} P(Y^m|X^m, G_{\mathbf{X}}, G_{\hat{\mathbf{Y}}}) \qquad (3)$$

2.2 Model Architecture

We augment the sentence-level TRANSFORMER model (Vaswani et al. 2017) with external encoders to learn the graph representations (described in Section 2.4). Such kind of architecture has proven beneficial for exploiting contextual knowledge (Zhang et al. 2018). Figure 2 illustrates the overall architecture of the proposed model.

Encoder Assume $H_{G_x} \in \mathbb{R}^{L \times d}$ is the representation of the source document graph after the graph encoder, where L denotes the number of nodes, and d is the dimension size. The encoder repeatedly aggregates input representations with a few stacked layers to generate the final representations of the current source sentence.

Taken the last encoder layer as an example, it first learns representations with a self-attention sublayer on an input H:

$$H_a = \mathrm{SelfAtt}(H) \in \mathbb{R}^{I \times d} \qquad (4)$$

where I is the length of the input sequence, the $\mathrm{SelfAtt}$ is a multi-head attention function (Vaswani et al. 2017), which maps three inputs Q, K and V to an output. When $Q = K = V$, we use $\mathrm{SelfAtt}$ in this paper. When $Q \neq K = V$, we call cross-attention denoted by $\mathrm{CrossAtt}$. Note that for simplicity, we ignore descriptions on residual connections (He et al. 2016), and layer normalization (Ba, Kiros, and Hinton 2016), which are standard components of the TRANSFORMER model and in all sublayers.

After the self-attention sublayer, the encoder integrates the graph representations H_{G_x} via severing it as the K and V of a stacked cross-attention sublayer, namely:

$$H_c = \mathrm{CrossAtt}(H_a, H_{G_x}) \in \mathbb{R}^{I \times d} \qquad (5)$$

Instead of using the standard residual connection, in this sublayer, we adopt a gated mechanism following Zhang, Titov, and Sennrich (2019) to dynamically control the influence of context information:

$$\mathrm{Gate}(H_a, H_c) = \lambda H_a + (1 - \lambda) H_c \qquad (6)$$

$$\lambda = \sigma(W_a H_a + W_c H_c) \qquad (7)$$

where λ are gating weights, and $\sigma(\cdot)$ denotes the sigmoid function. W_a and W_c are the trainable parameters.

Finally, we get the output of the encoder layer with a position–wise feed–forward network (Vaswani et al. 2017):

$$H_x = \mathrm{FNN}(\mathrm{Gate}(H_a, H_c)) \qquad (8)$$

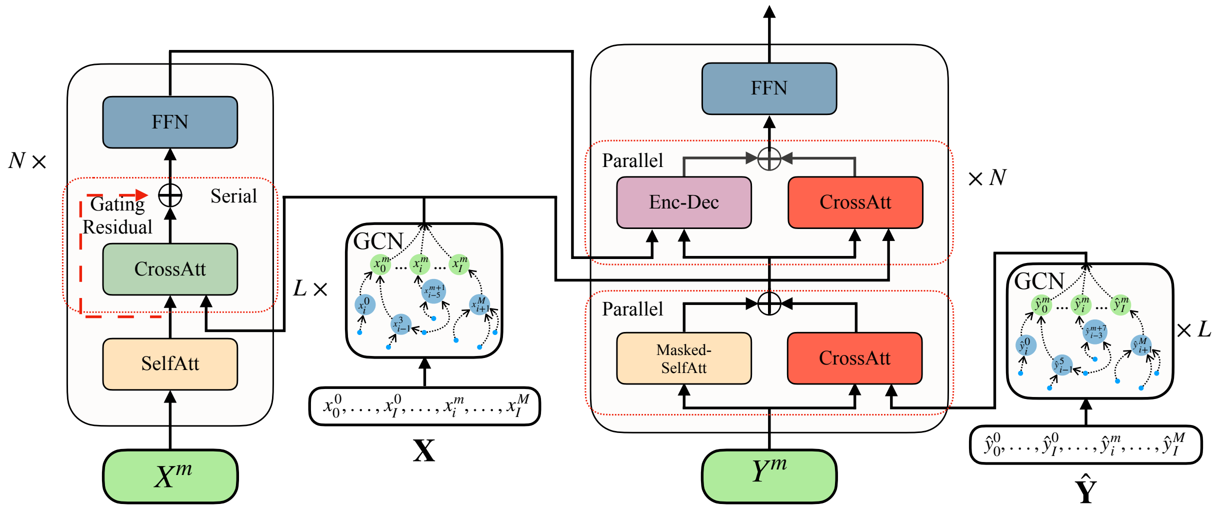


Figure 2: Illustration of the proposed model architecture. The dash blocks in red are used for context integration on both encoder and decoder. $\mathbf{X}$ is the document of the source language. The target document $\hat{\mathbf{Y}}$ consists of translations of $\mathbf{X}$ by a context-agnostic NMT. The parameters of CrossAtt in red are shared.

Decoder Different from the serial structure in the encoder, we consider both source and target document graphs in the decoder in a parallel structure (denoted by a function Parallel). This prevents the decoder from being deeper, and poor convergence (Zhang, Titov, and Sennrich 2019).

Assume $Y^m_{<t}$ to be the input representations of the target sequence with length $t-1$ and $H_{G_{\hat{y}}}$ the representations of target document graph. The first sublayer in a decoder layer aggregates representations from the current target sequence and the target graph:

$$\begin{aligned} H_a &= \mathrm{Parallel}(Y^m_{<t}, H_{G_{\hat{y}}}) \\ &= \mathrm{Gate}(\mathrm{SelfAtt}(Y^m_{<t}), \mathrm{CrossAtt}(Y^m_{<t}, H_{G_{\hat{y}}})) \end{aligned} \tag{9}$$

The second sublayer aggregates representations from the source sentence and the source graph as following:

$$\begin{aligned} H_c &= \mathrm{Parallel}(H_a, H_{G_x}, H_x) \\ &= \mathrm{Gate}(\mathrm{CrossAtt}(H_a, H_x), \\ &\qquad \mathrm{CrossAtt}(H_a, H_{G_x})) \end{aligned} \tag{10}$$

Similar to the encoder, we obtain the output of the decoder layer after a position–wise feed–forward network.

2.3 Graph Construction

Graphs used in this paper are directed, which can be represented as $G = (V, E)$, where V is a set of nodes and E is a set of edges where an edge $e = (u, v)$ with $u, v \in V$ denotes an arrow connection from the node u to the node v.

Given a document $\mathbf{X} = [\cdots; x_1^m, \cdots, x_{I_m}^m; \cdots]$ where x_i^m is the ith $(1 \le i \le I_m)$ word in the mth $(1 \le m \le M)$ sentence, we construct a document graph by treating words in the document as graph nodes and relations between words as graph edges. We consider both intra-sentential and inter-sentential relations. Figure 1 shows an example document graph. Note that not all edges are depicted for simplicity.

Intra-sentential Relations provide links between words in a sentence $X^m = x_1^m, \cdots, x_{I_m}^m$. These links are relatively local yet informative and help understand the structure and meaning of the sentence. In this paper, we consider two kinds of intra-sentential relations:

- **Adjacency** provides a local lexicalized context that can be obtained without resorting to external resources and has been proven beneficial to sentence modeling (Wu et al. 2018; Sperber et al. 2018). For each word x_i^m, we add two edges $(x_i^m, x_{i+1}^m\}$ and $(x_i^m, x_{i-1}^m\}$. This means we add links from the current word to its adjacent words.
- **Dependency** directly models syntactic and semantic relations between two words in a sentence. Dependency relations not only provide linguistic meanings but also allow connections between words with a longer distance. Previous practices have shown that dependency relations enhance representation learning of words (Marcheggiani and Titov 2017; Strubell et al. 2018; Lin, Yang, and Lai 2019). Given a dependency tree of the sentence and a word x_i^m, we add a graph edge (x_i^m, x_j^m) if x_i^m is a headword of x_j^m.

Inter-sentential Relations allow links from one sentence $X^m = x_1^m, \cdots, x_{I_m}^m$ to another following sentence $X^n = x_1^n, \cdots, x_{I_n}^n$. These relations provide discourse information, which is important for capturing document phenomena in document-level NMT (Tiedemann and Scherrer 2017; Voita et al. 2018). Accordingly, we consider two kinds of relations in our document graph:

- **Lexical consistency** considers repeated and similar words across sentences in the document, which reflects the cohesion of lexical choices. In this paper, we add edges $\{(x_i^m, x_j^n)\}$ if $x_i^m = x_j^n$ or $\mathrm{Lemma}(x_i^m) = \mathrm{Lemma}(x_j^n)$. Namely, the exact same words and words with the same lemma in the two sentences are connected in the graph.

- **Coreference** is a common phenomenon in documents and exists when referring back to someone or something previously mentioned. It helps understand the logic and structure of the document and resolve the ambiguities. In this paper we add a graph edge (x_i^m, x_j^n) if x_i^m is a referent of x_j^n given by coreference resolution.

Inter-sentential relations also exist between words in the same sentence, where $m = n$.

Using document graphs makes it easy to infuse relevant context information into the current word representation. Since not all words in the document are useful contexts (Kim, Tran, and Ney 2019), we exclude nodes with a distance greater than 2 to the current words in a graph. This lets our model focus more on relevant context and meanwhile reduces computation costs during learning graph representations.

2.4 Document Graph Encoder

As the document is projected into a document graph, a flexible graph encoder is required to encode the complex structure. Marcheggiani and Titov (2017) and Bastings et al. (2017) verified that GCNs can be applied to encode linguistic structures such as dependency trees. In this paper, we follow previous practices to use stacked GCN layers as the encoder of document graph.

GCNs are neural networks operating on graphs and aggregating information from immediate neighbors of nodes. Information of longer-distance nodes is covered by stacking GCN layers. Formally, given a graph $G(V, E)$, the GCN network first projects the nodes V into representations $H^0 \in \mathbb{R}^{I \times d}$, where d stands for hidden size and $I = |V|$. Node representations H^l of the lth layer can be updated as follows:

$$H^{l+1} = \sigma(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} (W^{l+1} H^l + B^{l+1})) \qquad (11)$$

where σ is the sigmoid function and $W^{l+1} \in \mathbf{R}^{\mathbf{d}\times\mathbf{d}}$, $B^{l+1} \in \mathbf{R}^{\times\mathbf{d}}$ are learnable parameters, $A \in \mathbf{R}^{\mathbf{I}\times\mathbf{I}}$ is an adjacency matrix that stores edge information:

$$A(i, j) = \begin{cases} 1, & \exists (v_i, u_j) \in E, \\ 0, & \text{otherwise.} \end{cases} \qquad (12)$$

The degree matrix $D \in \mathbf{R}^{\mathbf{I}\times\mathbf{I}}$ is assigned to weight the expected importance of a current node based on the number of input nodes, which can be calculated with the adjacency matrix:

$$D(i, j) = \begin{cases} \sum_{j'=1}^{I} A(j', i), & i = j, \\ 0, & \text{otherwise.} \end{cases} \qquad (13)$$

Equation (12) only considers input features. To fully use direction information in the graph, we apply GCN on different types of edges:

$$\hat{H}_t^{l+1} = \sigma(\hat{D}_t^{-\frac{1}{2}} \hat{A}_t \hat{D}_t^{-\frac{1}{2}} (\hat{W}_t^{l+1} H^l + B_t^{l+1})) \qquad (14)$$

where $t \in \{\text{in}, \text{out}, \text{self}\}$ represents one of the edge types, i.e., input edges, output edges, or a specific type of self-loop edges. We assume the contributions of the representations learned from a different kind of edges should be different. We then apply a type-attention mechanism, which works better than a linear combination in our experiments,[2] to combine these representations of different edge types:

$$H^{l+1} = \sum_t \alpha_t \hat{H}_t^{l+1} \qquad (15)$$

$$\alpha_t = \text{Softmax}(\frac{H^l \hat{H}_t^{l+1}}{\sqrt{d}}) \qquad (16)$$

where the α_t are attention weights given by a dot-product attention algorithm (Vaswani et al. 2017).

3 Experiments

Data We evaluate our approach on translation benchmarks with different corpus size: (1) IWSLT En–Fr and Zh–En translation tasks (Cettolo, Girardi, and Federico 2012) with around 200K sentence pairs for training. Following convention (Wang et al. 2017; Miculicich et al. 2018; Zhang et al. 2018), both language pairs take dev2010 as the development set. tst2010 is used for testing on En–Fr and tst2010∼tst2013 on Zh–En. (2) Opensubtitle2018 En–Ru translation corpus released by Voita et al. (2018), which contains 1.5M sentence pairs for training. (3) WMT19 En–De document-level translation task which consists of Europarl, Rapid and News-Commentary with a total of 3.7M sentence pairs. We use newsdev2019 as the development set and newstest2019 as the test set.[3]

All data are tokenized and segmented into subword units using the byte-pair encoding (Sennrich, Haddow, and Birch 2016). We apply 32k merge steps for each language on En-Fr, En-Ru, En-De tasks, and 30k for Zh-En task. As a node in a document graph represents a word rather than its subwords, we average embeddings of the subwords as the embedding of the node. The 4-gram BLEU (Papineni et al. 2002) is used as the evaluation metric.

Models and Baselines The proposed model is trained in two stages (Jean et al. 2015): conventional sentence-level TRANSFORMER models (denoted as BASE) are first trained with configurations following previous works (Zhang et al. 2018; Miculicich et al. 2018; Voita, Sennrich, and Titov 2019b; Vaswani et al. 2017); then, we fix sentence-level model parameters and only train document-level model parameters introduced by our methods. We set the layers of the document graph encoder to 2.[4]

To evaluate the performance of our model, we re-implement several document-level baselines on the TRANSFORMER architecture:

- CTX (Zhang et al. 2018) employs an additional encoder to learn context representations, which are then integrated by cross-attention mechanisms.
- HAN (Miculicich et al. 2018) uses a hierarchical attention mechanism with two levels (word and sentence) of abstraction to incorporate context information from both source and target documents.

[2] We report our experiments in Section 2 of Supplementary.

[3] Please refer to Table 1 in Supplementary for more details.

[4] More details are provided in Section 2 in Supplementary.

Model	**En-Fr**		**Zh-En**		**En-DE**		**En-Ru**		**Para.** △	**Speed**
	Dev	Test	Dev	Test	Dev	Test	Dev	Test		
BASE	29.56	35.77	10.92	16.7	35.91	34.89	28.44	29.05	-	24.9k
Constrained Context										
HAN	29.93	36.15	11.37	17.65	36.01	35.00	28.92	29.38	7.36 M	14.4k
CTX	30.23	36.67	11.37	17.57	36.05	35.23	28.79	29.31	22.06M	16.3k
CACHE	30.17	36.27	11.36	17.39	35.93	35.12	29.31	29.53	1.84 M	18.6k
Global Context										
MS	29.04	34.93	10.59	16.12	35.87	34.59	27.89	28.70	0.00 M	16.1k
HM-GDC	30.36	36.38	11.54	17.52	35.97	35.01	28.96	29.12	7.30 M	19.9k
SELECTIVE	30.53	36.87	11.57	17.86	36.14	35.47	29.67	29.64	8.39 M	7.7k
Our										
SRC–GRAPH	$30.84^{\Uparrow}$	$37.11^{\Uparrow}$	$11.75^{\uparrow}$	$18.31^{\Uparrow}_{\ddagger}$	36.21	$35.68^{\uparrow}$	$29.78^{\uparrow}$	29.72	22.59M	17.2k
+TGT	$31.62^{\Uparrow}_{\ddagger}$	$37.71^{\Uparrow}_{\ddagger}$	$12.01^{\Uparrow}_{\dagger}$	$18.53^{\Uparrow}_{\ddagger}$	$36.34^{\uparrow}$	$35.94^{\Uparrow}_{\dagger}$	$30.14^{\Uparrow}\ddagger$	$30.10^{\Uparrow}_{\dagger}$	22.59M	15.9k

Table 1: Main results (BLEU) on IWSLT Zh–En and EN–FR, WMT19 En–De, and Opensubtitle2018 En–Ru translation tasks. "↑ / ⇑" denotes significant improvement (Koehn 2004) over the best baseline model with constrained context on each task at $p < 0.05/0.01$, respectively. The significant improvement with respect to the SELECTIVE model is represented as "†/‡". "Para." and "Speed" indicate the model size (M = million) and training speed (tokens/second), respectively.

- CACHE (Tu et al. 2018) introduces a cache to memorize previous hidden states as dynamically updated context during decoding.
- MS (Junczys-Dowmunt 2019) treats a document as a long sentence and directly translates it with a sentence-level NMT model.
- HM-GDC (Tan et al. 2019) learns representations with a global context using a hierarchical attention mechanism.
- SELECTIVE (Maruf, Martins, and Haffari 2019) consider both source and target documents by selecting relevant sentences as contexts from a document.

3.1 Overall Results

Table 1 shows the overall results on four translation tasks. Our system achieves the best performance among all context-aware systems on all language pairs with comparable training speed. This verifies our hypothesis that document graphs are beneficial for modeling and leveraging the context. Compared with the CTX, our model has a comparable number of parameters indicating that the improvements of our method are not because of parameter increments. Although MS (Junczys-Dowmunt 2019) considers the whole document-level context as well, Table 1 shows that it does not achieve better performance than the BASE model.[5] By contrast, our method outperforms both the BASE model and strong document-level approaches, which only consider the limited context for better performance, suggesting the effectiveness of graph-based context learning.

3.2 Ablation Study

This section presents more details on the proposed model, including the influence of graph construction and variants of context integration.

Ablation	**Model**	**Dev**	**Test**
	BASE	29.56	35.77
Relations	+ADJACENCY	30.47	36.69
	+DEPENDENCY	30.31	36.75
	+LEXICAL	30.33	36.64
	+COREFERENCE	30.16	36.34
	+ALL	**30.84**	37.11

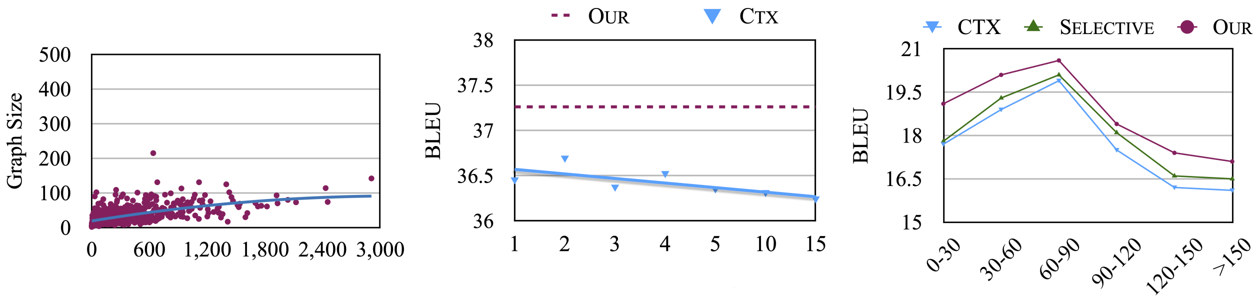
Table 2: Ablation study of source graph variants on the IWSLT En-Fr benchmark,where LEXICAL represents "Lexical consistency".

Graph Construction We first inspect each kind of edge relations individually by constructing graphs using only one of them. Table 2 shows that each kind of relations themselves improve the translation quality over the BASE model, which demonstrates the effectiveness of selected intra-sentential and inter-sentential relations. When we use all kinds of relations to construct graphs, translation quality is further improved, which indicates that the selected relations in this paper are complementary to some extent.

Integration Architectures As mentioned in Section 2.2, we use a SERIAL structure to integrate source graph into the encoder and a PARALLEL structure to integrate source and target graphs into the decoder. We call this kind of mixed structures as HYBRID. In this section, we present experimental results during evolving the context integration architecture, as shown in Table 3.

We first conduct experiments on architectures with only source graphs, which are integrated into both encoder and decoder following Zhang et al. (2018). We found that using

[5] To avoid the influence of incorrectly generated sentence boundaries (Junczys-Dowmunt 2019), we further calculate BLEU scores by treating a document as a sentence. We found that MS still underperforms the BASE model. Details in Table 5 of Supplementary.



(a) Text-Distance (b) Context-Length (c) Document-Size

Figure 3: (a) Illustration of context distance on the En–Fr task. The graph size means the number of nodes in a graph. The text distance denotes the number of words between two words in the document text. (b) Visualization of the effectiveness based on the number of sentences considered as contexts. The straights are the trend-line of the tested models. (c) Visualization of the effectiveness based on the number of sentences on a document, examined on testing set of Zh-En which contains 56 documents.

Architecture	Graph	Dev	Test
Serial	Src	30.55	36.65
Parallel	Src	30.53	36.70
Hybrid	Src	**30.84**	**37.11**
+Series	+Tgt	30.64	36.97
+Parallel	+Tgt	**31.62**	**37.71**
+Parallel	+Tgt-prev.	31.24	37.48

Table 3: BLEU scores of architecture variants for integrating graph representations on the IWSLT En–Fr task. Serial refers to the serial structure, while Parallel denotes the parallel structure. Hybrid represents using Serial in encoder and Parallel in the decoder. Tgt-prev. builds the graph on the previous sentence at the inference step. Note that source graphs are used in both encoder and decoder.

Hybrid structure, namely serial in the encoder and parallel in the decoder, outperforms using only one kind of structure. This suggests different information flow in the encoder and decoder is beneficial. Then, we integrate target graph representations into the decoder. Results show that using a parallel structure to incorporate target graphs in the decoder achieves the best performance. Finally, we test our best model with the graph, which is constructed with only previous sentences (i.e., Tgt-prev.) at the inference step. We found that the Tgt-prev. is only slightly worse than Tgt, indicating that the contributions of the target graph are more from the context rather than the translation itself of the current sentence.

3.3 Analysis

In this section, we analyze the proposed method to reveal its strengths and weaknesses in terms of (1) context distance and its influence; and (2) changes in document phenomena of translations.

Context Distance Because a graph allows to directly connect a word with its contexts regardless of their distances in the document (text distance), it can represent longer-distance contexts with a much smaller graph size, i.e., the number of nodes in a graph. Figure 3a shows statistics of graph size and text distance in the En–Fr development set. We found that the increase in graph size is much slower than text distance. This suggests that our model can encode relevant long-distance context without increasing much computational cost.

Model	Deixis	Lex.C	Ell.inf	Ell.VP
Base	50.0	45.8	67.6	36.0
HAN	56.3	52.3	72.7	58.7
Our	60.5	54.5	75.6	59.1

Table 4: Accuracy on Contrastive test sets. **Deixis** is the deictic words or phrases whose denotation depends on the context. **Lex.C** focuses on the reiteration of named entities. **Ell.inf** aims at the morphological form depend on the context. **Ell.VP** is a test for the ellipsis verb phrase.

Figure 3b shows the influence of text distance on translation quality. We found that Ctx performs worse when increasing the number of context sentences. One possible reason is that sequential structures introduce not only long-distance context but also more irrelevant information. By contrast, our model considers the whole document and is consistently better than Ctx. This suggests that graphs help the model focus on relevant contexts regardless of their distance.

Figure 3c shows evaluation results on different document lengths, i.e., the number of sentences in the document. We found that models considering global context (Selective and Our) achieve better results than Ctx. Our is consistently better than Selective as well, especially on shorter and longer documents. These results suggest that a global context is beneficial to document-level NMT and appropriate consideration of global context is essential.

Consistency Following Voita, Sennrich, and Titov (2019a), we evaluate our model on the consistency test sets.

It contains four tasks on En–Ru: 1) **Deixis** aims to detect the deictic words or phrases whose denotation depends on the context. 2) **Lex.C** is a lexical cohesion task, which focuses on the reiteration of named entities. 3) **Ell.inf** tests the model

Model	Position	Sentence
SRC	0	tongshi ye chuangjiu le yige guanyu ≪ moshoushijie ≫ de ru shishi ban de juda de zhishi ziyuan
	47	zheshi moshoushijie xilie de zhanlue youxi er zhe jiushi 16 nianqian de shiqing
	48	guren wan touzi youxi changda 18 nian women ze wan moshou 16 nian
REF	0	they are building an epic knowledge resource about the world of warcraft.
	47	that was the first real-time strategy game from the world of warcraft series. that was 16 years ago.
	48	they played dice games for 18 years, we’ve been playing warcraft for 16 years.
BASE	0	it also creates a tremendous resource of the world of ”world of warcraft.”
	47	this is the first game in the world of warcraft, which is years ago .
	48	we’ve been playing with a dice game for years.
HAN	0	and it was also a great science of ’ world of warcraft . ’
	47	this is the first real @-@ time game in world of warcraft , 16 years ago .
	48	they played dice for 18 years , and we played warcraft.
OUR	0	it also creates a knowledge of the epic knowledge of the world of warcraft .
	47	this is the first real strategic game in the world of warcraft, and this is what happened 16 years ago .
	48	we’ve been playing dice for 18 years, and we’ve been playing world of warcraft for 16 years .

Table 5: An example of Zh–En task. Compared with BASE and HAN, OUR system consistently generates “world of warcraft”.

on words whose morphological form depends on the context. 4) **Ell.VP** is to test whether the model can correctly predict the ellipsis verb phrase in Russian. As shown in Table 4, both the HAN and OUR models comprehensively improve the consistency over the context-agnostic BASE. On the **Deixis** and **Lex.C** tasks, our model outperforms the HAN over two points. We attribute this to the fact that our document graph contains intra-sentential relations, i.e., lexical consistency and coreference, directly linking relevant contexts for repeated and deictic words. While on the ellipsis tasks where graph edges are usually missing for elided verb phrases, our approach still obtains comparable performance as HAN. For example, given the following source sentence and its context (Voita, Sennrich, and Titov 2019b), the verbs “do” and “saw” are not connected in our graph. We want to cover these phenomena in future work.

Context	. . . you **saw** what happened.
Source	We all **did**.

To verify long-distance consistency, we perform case studies on the Zh–En task. Table 5 shows an example where a source phrase “moshoushijie” (world of warcraft) repeatedly appears in different positions in the document. We first found that both document-level NMT systems, i.e., HAN and OUR, generate more consistent translations of the source phrase than the context-agnostic BASE model. Compared with HAN model, OUR system surprisingly translates “moshou” (warcraft) into its full name, i.e., “world of warcraft” consistent with previous translations, suggesting a more effective capability of handling consistency.

4 Related work

In recent years, a variety of studies work on improving document-level machine translation with contextual information. Most of them focus on using a limited number of previous sentences. One typical approach is to equip conventional sentence-level NMT with an additional encoder to learn context representations, which are then integrated into encoder and/or decoder (Jean et al. 2017; Zhang et al. 2018; Voita et al. 2018). Wang et al. (2017) and Miculicich et al. (2018) adopted hierarchical mechanisms to integrate contexts into NMT models. Tu et al. (2018) and Kuang et al. (2018) used cache-base methods to memorize historical translations which are then used in following decoding steps.

Recently, several studies have endeavoured to consider the full document context. Macé and Servan (2019) averaged the word embeddings of a document to serve as the global context directly. Maruf and Haffari (2018) applied a memory network to remember hidden states of the document, which are then attended by a decoder. Maruf, Martins, and Haffari (2019) first selected relevant sentences as contexts and then attended to words in these sentences. Tan et al. (2019) learned global context-aware representations by firstly using a sentence encoder followed by a document encoder. Junczys-Dowmunt (2019) considered the global context by merely concatenating all the sentences in a document.

Unlike previous approaches, we represent document-level global context in source and target graphs encoded by graph encoders and integrated into conventional NMT via attention and gating mechanisms.

5 Conclusion

In this paper, we propose a graph-based approach for document-level translation, which leverages both source and target contexts. Graphs are constructed according to inter-sentential and intra-sentential relations. We employ a GCN-based graph encoder to learn the graph representations, which are then fed into the NMT model via attention and gating mechanisms. Experiments on four translation tasks show the proposed approach consistently improves translation quality across different language pairs. Further analyses demonstrate the effectiveness of graphs and the capability of leveraging long-distance context. In the future, we would like to enrich the types of relations to cover more document phenomena.

References

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. In *arXiv preprint arXiv:1607.06450*.

Bastings, J.; Titov, I.; Aziz, W.; Marcheggiani, D.; and Sima'an, K. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *ACL*.

Cettolo, M.; Girardi, C.; and Federico, M. 2012. Wit3: Web inventory of transcribed and translated talks. In *EAMT*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Jean, S.; Cho, K.; Memisevic, R.; and Bengio, Y. 2015. On Using Very Large Target Vocabulary for Neural Machine Translation. In *ACL*.

Jean, S.; Lauly, S.; Firat, O.; and Cho, K. 2017. Does neural machine translation benefit from larger context? In *arXiv preprint arXiv:1704.05135*.

Junczys-Dowmunt, M. 2019. Microsoft translator at wmt 2019: Towards large-scale document-level neural machine translation. In *WMT*.

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. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP*.

Kuang, S.; Xiong, D.; Luo, W.; and Zhou, G. 2018. Modeling Coherence for Neural Machine Translation with Dynamic and Topic Caches. In *Coling*.

Läubli, S.; Sennrich, R.; and Volk, M. 2018. Has Machine Translation Achieved Human Parity? A Case for Document-level Evaluation. In *EMNLP*.

Lin, P.; Yang, M.; and Lai, J. 2019. Deep mask memory network with semantic dependency and context moment for aspect level sentiment classification. In *IJCAI*. AAAI Press.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*.

Macé, V.; and Servan, C. 2019. Using whole document context in neural machine translation. In *IWSLT*.

Marcheggiani, D.; and Titov, I. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *EMNLP*.

Maruf, S.; and Haffari, G. 2018. Document Context Neural Machine Translation with Memory Networks. In *ACL*.

Maruf, S.; Martins, A. F.; and Haffari, G. 2019. Selective Attention for Context-aware Neural Machine Translation. In *NAACL*.

Miculicich, L.; Ram, D.; Pappas, N.; and Henderson, J. 2018. Document-Level Neural Machine Translation with Hierarchical Attention Networks. In *EMNLP*.

Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; and Auli, M. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *NAACL-HLT*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *ACL*.

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

Shaw, P.; Uszkoreit, J.; and Vaswani, A. 2018. Self-Attention with Relative Position Representations. In *NAACL*.

Sperber, M.; Niehues, J.; Neubig, G.; Stüker, S.; and Waibel, A. 2018. Self-Attentional Acoustic Models. In *Interspeech*.

Strubell, E.; Verga, P.; Andor, D.; Weiss, D.; and McCallum, A. 2018. Linguistically-informed self-attention for semantic role labeling. In *EMNLP*.

Tan, X.; Zhang, L.; Xiong, D.; and Zhou, G. 2019. Hierarchical Modeling of Global Context for Document-Level Neural Machine Translation. In *EMNLP-IJCNLP*.

Tiedemann, J.; and Scherrer, Y. 2017. Neural Machine Translation with Extended Context. In *DiscoMT*.

Tu, Z.; Liu, Y.; Shi, S.; and Zhang, T. 2018. Learning to remember translation history with a continuous cache. In *TACL*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention Is All You Need. In *NISP*.

Voita, E.; Sennrich, R.; and Titov, I. 2019a. Context-Aware Monolingual Repair for Neural Machine Translation. In *EMNLP*.

Voita, E.; Sennrich, R.; and Titov, I. 2019b. When a Good Translation is Wrong in Context: Context-Aware Machine Translation Improves on Deixis, Ellipsis, and Lexical Cohesion. In *ACL*.

Voita, E.; Serdyukov, P.; Sennrich, R.; and Titov, I. 2018. Context-Aware Neural Machine Translation Learns Anaphora Resolution. In *ACL*.

Wang, L.; Tu, Z.; Way, A.; and Liu, Q. 2017. Exploiting Cross-Sentence Context for Neural Machine Translation. In *EMNLP*.

Wu, W.; Wang, H.; Liu, T.; and Ma, S. 2018. Phrase-level Self-Attention Networks for Universal Sentence Encoding. In *EMNLP*.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* .

Yang, B.; Tu, Z.; Wong, D. F.; Meng, F.; Chao, L. S.; and Zhang, T. 2018. Modeling Localness for Self-Attention Networks. In *EMNLP*.

Yang, Z.; Zhang, J.; Meng, F.; Gu, S.; Feng, Y.; and Zhou, J. 2019. Enhancing Context Modeling with a Query-Guided Capsule Network for Document-level Translation. In *EMNLP-IJCNLP*.

Zhang, B.; Titov, I.; and Sennrich, R. 2019. Improving Deep Transformer with Depth-Scaled Initialization and Merged Attention. In *EMNLP-IJCNLP*.

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 *EMNLP*.

A Experiments

Data The statistics of the datasets are reported in Table 6. As seen, we evaluate our approach on four translation benchmarks:

- In the En–Fr task on IWSLT, the training set contains 1,823 documents with 220K sentence pairs. For the development and testing, we follow Zhang et al. (2018) to use dev2010 and tst2010, which contains 8 documents with 887 sentence pairs and 11 documents with 1,664 sentence pairs.
- For the Zh-En benchmark on IWSLT, the training set is consisted of 1,718 documents with 208K sentence pairs. We take dev2010 for developing and tst2010∼tst2013 for testing as the setting in previous works (Wang et al. 2017; Miculicich et al. 2018), which contains 8 documents with 887 sentence pairs and 56 documents with 5,473 sentence pairs, respectively.
- For the En–Ru translation tasks, we carry out the experiment on the Opensubtitle2018 released by Voita et al. (2018), which contains 1.5M sentences for training and 10k for developing and testing. In this version, the corpus did not provide the document boundary but a current sequence with 3 previous sentences. Therefore, we treat them as a document for our method.
- WMT19 En–De document-level translation task which consists of Europarl, Rapid and News-Commentary with a total of 62,592 documents and 3.7M sentence pairs. The newsdev2019 and newstest2019 are used as the development set and testing set, which contains 122 documents with 2,998 sentence pairs and 123 documents with 1,997 sentence pairs, respectively. Note that, for our method aims at the document-level translation, we filter the document which contains sentence less than 10.

All data are lower-cased, tokenized and segmented into subword units using the byte-pair encoding (BPE) (Sennrich, Haddow, and Birch 2016). For the Chinese language, we segment the data set with the jieba toolkit but the Moses tokenizer.pl for the other languages. Following Zhang et al. (2018) and Voita et al. (2018), we apply BPE for each language on En-Fr,En-Ru and En-De with 32k merge steps. For the Zh-En task, we use 30k merge step for each language.

BPE for Graph Encoding All our data set are segmented by BPE algorithm but the graph constructions are not. To overcome this inconsistency, we initial the representation of each node which should be segmented to sub-words (conveniently, we call this node the root node.) via a simple average method. For each root node, we first get all the word embeddings of its sub-words. Then, we average these word embeddings and serve it as the representation of the root node.

Settings We incorporate the proposed approach into the widely used context-agnostic framework TRANSFORMER (Vaswani et al. 2017) on FAIRSEQ toolkit (Ott et al. 2019). The proposed approach is trained on a two-stage training method following the previous works in document-level translation (Zhang et al. 2018; Miculicich et al. 2018; Voita, Sennrich, and Titov 2019b; Vaswani et al. 2017). In the first stages, we train the conventional context-agnostic TRANSFORMER models with BASE settings, which sets the number of layers to 6 and hidden size to 512. For the IWSLT and Opensubtitle benchmarks, we training the context-agnostic model with 0.2 dropout. The learning rate is set to 0.0007 with 4k warm-up steps. In the second stage, the document-level models are first initialized by the parameters of the models pre-trained in the first stage. Then, we only train the document-level model parameters with one-tenth learning rate by fixing the learned parameters in the first stage.

In the training step, we set the batch size in the first stage referring to the previous works and a half in second stage. For En-Fr and Zh-En , each mini-batch contains approximately 24K words. For En-Ru and En-De, each mini-batch contains approximately 16k and 32k tokens, respectively. We set the dropout of the document graph encoder of source and target side to 0.2 and 0.2, respectively. For the layers of the document graph encoder, we set it to 2 which is based on the experiment results in Table 8. For the decoder, we share the parameters of two CrossAtt mechanism. In decoding, the beam size is set to 4. Following the setting of previous work (Zhang et al. 2018; Miculicich et al. 2018; Voita, Sennrich, and Titov 2019b), we set the hyper-parameter α of length penalty (Wu et al. 2016) to 0.6 for En–Fr, En–De, 0.5 for En-Ru and 1 for Zh–En.

Construction of Target Graph For the target graph, we use the pseudo data to construct the document graph. As our models are trained on two-stage, we use the context-agnostic model at the first stage to generate the pseudo target data from the source data. And then we serve these translations as the target document to construct our target document graph. For consistency, we generate the target graph for the training step and inference step in the same way.

B Ablation Study

Graph Encoder We extend the GCN-based graph encoder with an attention mechanism to combine different representations, which is different from the gate-based method in previous work (Bastings et al. 2017). Table 7 shows that the attention-based aggregation works better in our model. We presume this is because the attention mechanism balances the contributions of different representations.

Table 8 shows the influence of the graph encoder with various number of layers. We found that stacking two graph encoder layers obtains the best performance. Further increasing the number of layers does not lead improvement. This

Benchmark	Language	Training		Development		testing	
		Doc.	Sent.	Doc.	Sent.	Doc.	Sent.
IWSLT [6]	En–Fr	1,823	220K	8	887	11	1,664
	Zh–En	1,718	199K	8	887	56	5,473
Opensubtitle[7]	En–Ru	1.5M	1.5M	10K	10K	10K	10K
WMT[8]	En–De	62,592	3.7M	122	2,998	123	1,997

Table 6: Statistics of the Dataset, where “Doc.” is the count of documents and “Sent.” denotes the number of sentence pairs.

Aggregation	BLEU
GATING UNITS	30.71
ATTENTION	**30.84**

Table 7: Results of aggregation methods in the graph encoder for combining representations learned from different edge directions. GATING UNITS denotes the weights of summation are calculated by a gating mechanism (Bastings et al. 2017). ATTENTION generates weights with an attention mechanism.

#Layers	BLEU
1	30.77
2	**30.84**
3	30.82

Table 8: Influence of the number of Graph encoder layers used in the graph encoder on IWSLT En–Fr task.

Ablation	Model	BLEU
Context Scope	ALL	30.22
	RELATED	30.62
	CURRENT	30.84

Table 9: Ablation study of context scope on IWSLT En-Fr benchmark.

Model	En–Fr	Zh–En	En–DE	En–Ru
BASE	38.47	20.35	38.19	32.20
MS	37.60	18.66	37.95	30.28
OUR	40.12	21.80	38.53	34.12

Table 10: Evaluation results on IWSLT Zh–En and EN–FR, WMT19 En–De and Opensubtitle18 En–Ru translation tasks by treating each document as a sentence during BLEU calculation.

finding is consistent with existing works as well (Marcheggiani and Titov 2017; Bastings et al. 2017).

Context Scope Although document graphs include various relevant contexts together with a current sentence, it does not mean using all of them is the best practice when integrating to NMT. Therefore, we conduct experiments to investigate influence of context scope, which defines a set of nodes directly attended by the encoder and decoder. We define three context scopes: (a)ALL refers to attending to all nodes of input graphs; (b) RELATED means each source word learn contextual knowledge from its immediate neighbours in the document graph; (c) CURRENT denotes that the source sentence only attends to their corresponding nodes in the document graph.

As shown in Table 9, different context scopes consistently improve the model performance over the BASE model. However, the best performance is achieved when the CURRENT scope is adopted. This demonstrates that restricting the attention scope to a local area is beneficial to leverage the context. This finding is consistent with previous practices (Luong, Pham, and Manning 2015; Yang et al. 2018; Shaw, Uszkoreit, and Vaswani 2018). Accordingly, we only use the CURRENT method to leverage the context knowledge in this paper.

Document BLEU To avoid the influence of incorrectly generated sentence boundaries (Junczys-Dowmunt 2019), we further calculate BLEU scores by treating a document as a sentence with results shown in Table 10. We found that MS still under-performs the BASE model. By contrast, our method outperforms both the BASE model and strong document-level approaches which only consider limited context for better performance, suggesting the effectiveness of graph-based context learning.

C Overall Results