Document-level Neural Machine Translation Using Dependency RST Structure

Anonymous ACL submission

Abstract

Document-level machine translation (MT) extends the translation unit from the sentence to the whole document. Intuitively, discourse structure can be useful for document-level MT for its helpfulness in long-range dependency modelling. However, few efforts have been paid on leveraging discourse information for document-level neural machine translation (NMT). In this paper, we propose a dependency Rhetorical Structure Theory (RST) tree enhanced NMT model, RST-Transformer. The model only needs to encode the dependency RST tree of the source document via the attention mask, and can enhance both the encoder and the decoder. Experiments on English-German datasets in both non-pretraining and pretraining settings show that our discourse information enhanced approach outperforms the current state-of-the-art document-level NMT model.

1 Introduction

As the neural machine translation (NMT) (Bahdanau et al., 2014; Vaswani et al., 2017) gets close to human performance on the sentence-level translation, the mistakes at the document-level become more obvious (Kim et al., 2019). Previous work (Voita et al., 2019; Ma et al., 2021) shows these mistakes could be reduced by introducing contexts into context-agnostic NMT models.

Previous methods that explored to integrate context information into NMT models can be broadly divided into two categories. The first category takes fixed scope sentences as context and modeling the context-aware representation by extra context encoder (Miculicich et al., 2018; Wang et al., 2017a) or unified encoder (Ma et al., 2020; Zheng et al., 2020). The second category takes global context into consideration by using translation memory (Kuang et al., 2018; Tu et al., 2018), hierarchical model (Tan et al., 2019; Maruf et al., 2019) or reinforcement learning (Kang et al., 2020). However, exploring structured information for document-level NMT has so far received relatively little attention (Xiaomian and Chengqing, 2020; Chen et al., 2020; Xu et al., 2020b). Structured information is proven be helpful for sentence-level MT in enforcing meaning preservation (Marcheggiani et al., 2018), handling data sparsity (Song et al., 2019) and modeling long-range dependencies (Zhang et al., 2019; Wu et al., 2017). In another aspect, the discourse structure helps many other document-level tasks like summarization (Xu et al., 2020a), sentiment analysis (Huber and Carenini, 2020), translation evaluation (Guzmán et al., 2014; Joty et al., 2017).

In this paper, we present RST-Transformer, a discourse-aware document-level NMT model build upon Transformer (Vaswani et al., 2017). To shorten long-range dependencies and mask unrelated sentences, we take dependency RST tree (Mann and Thompson, 1987), a structure describ-
ing how text spans in a document related to each other, as the indicator for guiding document-level MT. Figure 1 shows an example for cross-sentence dependencies, where the "It" in sentence 3 refers to "coat" in the sentence 1. In German, every noun has a gender (masculine, feminine, neuter) and the pronoun referring to it also use the same gender. For example, If the masculine noun "coat" is changed to a feminine noun, like "pants", the corresponding translation of "It" should be changed from "Er" to "Sie". Guided by the dependency RST tree, the RST-Transformer can discard redundant context and obtain additional capacity to include more long-range dependencies, producing fewer document-level mistakes.

More specifically, the RST-Transformer has two different attention modules: Sentence attention (SentAttn) and RST attention (RSTAttn), which have sentence attention mask and RST attention mask, respectively. The sentence attention mask is used to differentiate the current sentence and its context. It makes the SentAttn focus on local sentences. The mask in RSTAttn is converted by the dependency RST tree of the source document. It makes the RSTAttn focus on spans related to the current span in the dependency RST tree. To fusion the sentence-level information encoded in SentAttn and document-level information encoded in the RSTAttn, we investigate three fusion methods: Serial, Parallel and Mix.

We evaluate our model on three commonly used document MT benchmark datasets for English-German translation. The results show that the RST-Transformer outperforms Transformer and current state-of-the-art document-level NMT models both on non-pretraining and pretraining settings. Further, we demonstrate that the RST-Transformer can still surpass other models without introducing the dependency RST tree in the decoding process.

2 Problem Definition

Formally, we denote \( X = \{X_1, X_2, \ldots, X_N\} \) as the source document with \( N \) sentences and \( Y = \{Y_1, Y_2, \ldots, Y_N\} \) as the target document with the same number of sentence. We assume that each source sentence \( X_i \) is aligned with the target sentence \( Y_i \), where \( i \in [1, N] \).

---

\(^{1}\)It is analogous to GroupAttn in G-Transformer (Bao et al., 2021). G-Transformer is the state-of-the-art document-level NMT model, which encode the local and global attention by GroupAttn and GlobalAttn respectively.

---

2.1 Dependency RST Tree

In the RST framework, the discourse structure of the text is represented in a constituent tree structure. The dependency RST tree is the dependency perspective of the RST tree (Li et al., 2014; Morey et al., 2018), which is helpful in text summarization (Xu et al., 2020a; Hirao et al., 2013). The way to get the dependency RST tree of the input document is shown in Figure 2. In step ①, the document is split into contiguous, adjacent and non-overlapping text spans called Elementary Discourse Units (EDUs), a sub-sentence phrase unit originating from RST. In step ②, non-terminal
nodes are composed of two or more adjacent EDUs merged upwards to form an RST tree. When merging, the more semantically important unit is called the "nucleus", and the other units that modify the "nucleus" are called "satellite". The arc from π3 to π2 means π2 is a "satellite" modifying the "nucleus" π3. The "nucleus-satellite" relationship can be further refined into a variety of rhetorical relationships such as elaboration and background. Finally, in step 3, the RST tree is converted to its dependency perspective (Li et al., 2014).

2.2 Trees for source and target documents
We use third-party tools to get the dependency RST tree for the source document. The details can be found in Section 4.2. However, the target document cannot obtain the dependency RST tree during decoding. The one-to-one correspondence between sentences in the source and target document makes the inter-sentence relationship between the two documents consistent. Thus we can obtain a coarse-grained dependency RST tree for the target document. As shown in Figure 3, the tree for the target document can be converted by the tree of the source document by merging the EDUs in the same sentence and ignoring the inner-sentence relations. Zhang et al. (2021) empirically shows no EDU across sentences, and all EDUs in the same sentence should be exactly covered by a complete subtree. This ensures that leaf nodes could be sentences and the target document tree still forms a tree.

3 RST-Transformer
Our RST-Transformer extends from the Transformer NMT (Vaswani et al., 2017) architecture. Figure 4(a) provides an overview of the proposed model. The encoder and decoder are composed of \( N_s \) Sentence layers and \( N_t \) RST layers, respectively. According to Jawahar et al. (2019), lower layers of Transformer catch more local syntactic relations, while the higher layers represent long-distance relations. Based on this finding, we use the RST layer on the top layers for cross-sentence interaction and the Sentence layer on the lower layers for sentence-level information. Section 3.1 and Section 3.2 describes the Sentence and RST layers, respectively. Section 3.3 introduces three proposed methods to infuse local and global information in the RST layer.

3.1 Sentence Layer
Our sentence layer is the Transformer (Vaswani et al., 2017) considering sentence boundary. Roughly speaking, Transformer consists of two sublayers: self-attention network and feed-forward network. We integrate the sentence boundary information into the self-attention module. The attention function discussed maps a query and a set of key-value pairs to an output. The self-attention module produces representations by applying the attention function to each pair of tokens from the input sequence. Given a text representation \( H \in \mathcal{R}^{T \times d} \):

\[
Q = HW_Q, K = HW_K, V = HW_V \tag{2}
\]

\[
\text{Attn}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V \tag{3}
\]

where the projection matrices \( \{W_Q, W_K, W_V\} \in \mathcal{R}^{d \times d_k} \) are trainable parameters, \( T \) denotes the length of input tokens and \( d, d_k \) are the embedding size and hidden size, respectively.

The self-attention network can capture global contextual features. But too much information may cause training failure on the document-level translation task (Bao et al., 2021). Thus, we update Equation 3 by sentence boundary guiding, naming sentence attention (SentAttn). In addition to input \( Q, K \) and \( V \), the sentence mask \( M_s \in \mathcal{R}^{T \times T} \) is also involved:

\[
\text{SentAttn}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} + M_s \right) V \tag{4}
\]

where the sentence mask \( M_s \) is an indicator that detects whether two tokens are in the same sentence. Specifically, \( M_s \) gives negative infinity for the token pairs not in the same sentence to make softmax close to 0. Note that the SentAttn is same as the GroupAttn in Bao et al. (2021). For every two tokens \( x_i, x_j \) in the source document, we have:
3.2 RST Layer

Our RST layer has two attention modules and one feed-forward module. One attention module is SentAttn defined in Equation 4, which captures local information. The other is RST attention (RSTAttn), used to capture the discourse structure enhanced global information.

Similarly, we update Equation 3 using the dependency RST tree. We involve RST mask $M_{rst}$ into it:

$$RSTAttn(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} + M_{rst} \right) V$$

where the RST mask $M_{rst}$ is an indicator. If the EDUs where two tokens $x_i$ and $x_j$ located are directly related (parent or child) in the dependency RST tree or same, $M_{rst}(i, j)$ is set to 0 and negative infinity otherwise:

$$M_{rst}(i, j) = \begin{cases} 0 & x_i, x_j \text{ in same EDU or related,} \\ -\infty & \text{otherwise.} \end{cases}$$

The upper half of Figure 4(b) shows an example of $M_{rst}$. The dependency RST tree of the source and target document can be get from the Section 2.1.
3.3 Fusion Methods for SentAttn and RSTAttn

To investigate what kind of structure is better for information fusion between SentAttn and RSTAttn, we propose three different RST-Transformer structures: Serial, Parallel and Mix. Their structural diagram is shown in Figure 4(c).

Given the source document representation $H \in \mathbb{R}^{T \times T}$, we can get the $Q$, $K$, $V$ same as Equation 2:

- **Serial** connects SentAttn and RSTAttn and is in series:

$$H'_s = \text{SentAttn}(Q, K, V),$$

$$\{Q'_s, K'_s, V'_s\} = H'[W'_s, W'_K, W'_V]$$

$$H' = \text{RSTAttn}(Q'_s, K'_s, V'_s),$$

where the projection matrices $\{Q'_s, K'_s, V'_s\}$ are trainable parameters.

- **Parallel** uses a gate-sum module (Zhang et al., 2016; Tu et al., 2017) to combine RSTAttn and SentAttn:

$$H'_{st} = \text{RSTAttn}(Q, K, V),$$

$$g = \text{sigmoid}([H'_s, H'_st]W + b),$$

$$H' = H'_s \odot g + H'_{st} \odot (1 - g),$$

where the $H'_s$ obtained by Equation 8, which is same as Serial, $W$ and $b$ are linear projection parameters, and $\odot$ denotes element-wise multiplication.

- **Mix** integrates the Parallel and Serial structures. We obtain $H'_s$ and $H'_{st}$ by Equation 8-10, which are same as Serial. Then we combine $H'_s$ and $H'_{st}$ to obtain the $H'$ by a gate-sum module (Equation 12-13), which is same as Parallel.

4 Experiments

We compare our RST-Transformer with sentence-level Transformer and previous document-level NMT models on both non-pretraining and pretraining settings. We conduct experiments on three English-German (En-De) datasets. Following Liu et al. (2020), we calculate case-sensitive sentence BLEU (s-BLEU) and document BLEU (d-BLEU) as the metrics.

4.1 Datasets

Following previous work (Maruf et al., 2019), we use three En-De datasets as the benchmark to evaluate our method, which comes from three different domains: TED is transcriptions of TED talks from IWSLT 2017, News comes from News Commentary v11, Europarl is extracted from Europarl v7. The statistic of these datasets can be found in Table 1. We tokenize and truecase the sentences with MOSES (Koehn et al., 2007) tools, applying BPE (Sennrich et al., 2016) with 32000 merging operations. We split documents into instances with up to 512 tokens and ensure the integrity of the sentence.

4.2 Implement Details

We train our models on 4 GPUs of A100.

**Hyper-parameters** We use the Adam (Kingma and Ba, 2015) optimizer to train the models. The training strategy is the same as Bao et al. (2021), except for 1) the learning rate is 3e-4, 2) the batch size is limited to 4096, 8192, 8192 for TED, News and Europarl, respectively. We search the batch size in [4096, 8192] and the learning rate in [1e-4, 3e-4, 5e-4]. We determine the number of updates/steps automatically by early stop on validation set. $N_a$ is set to 4 and $N_b$ is set to 2 for all experiments. Following previous work (Bao et al., 2021; Miculicich et al., 2018), we train the models in two stages. First we optimize the parameters for a Transformer model with sentence-level data. Then we use the Transformer to initialize our RST-Transformer and continuous train on the document-level data.

**RST Discourse Parsing** We first apply NeuralEDUSeg (Wang et al., 2018) to obtain EDUs. Then, we use StageDP (Wang et al., 2017b) to parse the segmented document. After obtaining the RST tree, we apply the algorithm proposed by Li et al. (2014) to convert the RST tree to its dependency perspective.

4.3 Baselines

We compare our model with the following NMT models:

- **Transformer** is the base configuration Transformer NMT model in Vaswani et al. (2017) trained on...
<table>
<thead>
<tr>
<th>Model</th>
<th>TED s-BLEU</th>
<th>TED d-BLEU</th>
<th>News s-BLEU</th>
<th>News d-BLEU</th>
<th>Europral s-BLEU</th>
<th>Europral d-BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer</td>
<td>24.80</td>
<td>-</td>
<td>25.10</td>
<td>-</td>
<td>31.37</td>
<td>-</td>
</tr>
<tr>
<td>HAN</td>
<td>24.58</td>
<td>-</td>
<td>25.03</td>
<td>-</td>
<td>28.60</td>
<td>-</td>
</tr>
<tr>
<td>SAN</td>
<td>24.42</td>
<td>-</td>
<td>24.84</td>
<td>-</td>
<td>29.75</td>
<td>-</td>
</tr>
<tr>
<td>Hybrid Context</td>
<td>25.10</td>
<td>-</td>
<td>24.91</td>
<td>-</td>
<td>30.40</td>
<td>-</td>
</tr>
<tr>
<td>Flat-Transformer</td>
<td>24.87</td>
<td>-</td>
<td>23.55</td>
<td>-</td>
<td>30.09</td>
<td>-</td>
</tr>
<tr>
<td>G-Transformer</td>
<td>25.12</td>
<td>27.17</td>
<td>25.47</td>
<td>27.08</td>
<td>32.39</td>
<td>34.08</td>
</tr>
<tr>
<td>RST-Transformer-Serial</td>
<td>25.55</td>
<td>27.75</td>
<td>25.83</td>
<td>27.38</td>
<td>32.43</td>
<td>34.16</td>
</tr>
<tr>
<td>RST-Transformer-Parallel</td>
<td>25.57</td>
<td>27.75</td>
<td>26.05</td>
<td>27.68</td>
<td>32.64</td>
<td>34.33</td>
</tr>
<tr>
<td>RST-Transformer-Mix</td>
<td>25.61</td>
<td>27.84</td>
<td>26.43</td>
<td>28.00</td>
<td>32.87</td>
<td>34.64</td>
</tr>
<tr>
<td>BART fine-tuned on sentence</td>
<td>27.78</td>
<td>-</td>
<td>29.90</td>
<td>-</td>
<td>31.87</td>
<td>-</td>
</tr>
<tr>
<td>BART fine-tuned on doc</td>
<td>-</td>
<td>28.29</td>
<td>-</td>
<td>30.49</td>
<td>-</td>
<td>34.00</td>
</tr>
<tr>
<td>G-Transformer + BART</td>
<td>28.06</td>
<td>30.03</td>
<td>30.34</td>
<td>31.71</td>
<td>32.74</td>
<td>34.31</td>
</tr>
<tr>
<td>RST-Transformer-Serial + BART</td>
<td>27.96</td>
<td>29.95</td>
<td>30.35</td>
<td>31.60</td>
<td>32.94</td>
<td>34.49</td>
</tr>
<tr>
<td>RST-Transformer-Parallel + BART</td>
<td>28.29</td>
<td>30.26</td>
<td>30.71</td>
<td>32.01</td>
<td>33.01</td>
<td>34.59</td>
</tr>
<tr>
<td>RST-Transformer-Mix + BART</td>
<td>28.26</td>
<td>30.23</td>
<td>30.89</td>
<td>32.29</td>
<td>33.14</td>
<td>34.63</td>
</tr>
</tbody>
</table>

Table 2: Case-sensitive BLEU scores on En-De translation. "\(\dagger\)" indicates statistically significant (Koehn, 2004) over the state-of-the-art G-Transformer at \(p < 0.05\).

4.4 Results

Table 2 shows the overall results on three datasets. We find that our three fusion methods outperform the base model and previous document-level NMT models. Especially the RST-Transformer-Mix achieves the best results at all three datasets. The RST-Transformer-Mix can achieve improvements of 1.18 average s-BLEU scores over the context-agnostic Transformer. Therefore, the Mix is the most effective method of information fusion. It is worth noting that our RST-Transformer-Parallel model is better than G-Transformer on average by 0.63 s-BLEU scores and 0.71 d-BLEU scores, respectively. As mentioned above, G-Transformer can be regarded as the RST-Transformer-Parallel model without dependency RST tree. It proves that the RST structure can provide effective information.

There is an active topic about document-level MT using pretraining. We use mBART25 (Liu et al., 2020) to initialize our RST-Transformer and finetune it with learning rate \(3e^{-4}\). Taking advantage of sequence-to-sequence pretraining, the result of RST-Transformer-Mix+BART is 2.46 s-BLEU scores better than the RST-Transformer-Mix on average. Compared to the document-level state-of-the-art pretrained model G-Transformer+BART, our best model gives 0.55 s-BLEU score improvement on the News dataset. It shows the discourse structure information also enhances performance although in well-pretrained settings.

5 Analysis

In this section, we investigate our RST-Transformer to reveal its strengths and weaknesses in terms of (1) training without the Transformer initialization, (2) decoding without dependency RST tree, (3) changes in document phenomena of translations, (4) a case study. We use the Mix structure for...
Table 3: s-BLEU on different model size. "Trans." is Transformer, "G-Trans" is G-Transformer, "our-Mix" is our RST-Transformer-Mix.

<table>
<thead>
<tr>
<th>Size</th>
<th>Method</th>
<th>TED</th>
<th>News</th>
<th>Europarl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>Trans</td>
<td>0.42</td>
<td>0.45</td>
<td>30.40</td>
</tr>
<tr>
<td></td>
<td>G-Trans</td>
<td>23.53</td>
<td>23.55</td>
<td>32.18</td>
</tr>
<tr>
<td></td>
<td>our-Mix</td>
<td>24.28</td>
<td>24.12</td>
<td>32.33</td>
</tr>
<tr>
<td>Big</td>
<td>Trans</td>
<td>0.72</td>
<td>0.33</td>
<td>27.33</td>
</tr>
<tr>
<td></td>
<td>G-Trans</td>
<td>23.29</td>
<td>22.22</td>
<td>32.04</td>
</tr>
<tr>
<td></td>
<td>our-Mix</td>
<td>23.99</td>
<td>23.97</td>
<td>33.25</td>
</tr>
<tr>
<td>Large</td>
<td>Trans,</td>
<td>0.23</td>
<td>0.31</td>
<td>1.27</td>
</tr>
<tr>
<td></td>
<td>G-Trans</td>
<td>6.23</td>
<td>13.68</td>
<td>31.51</td>
</tr>
<tr>
<td></td>
<td>our-Mix</td>
<td>13.22</td>
<td>20.04</td>
<td>31.99</td>
</tr>
</tbody>
</table>

5.1 Training From Scratch

Most of the current document-level NMT models are initialized by a pretrained sentence-level Transformer because this strategy can improve the performance. Bao et al. (2021) also shows the current transformer model prefers to stick around local minima without the initialization. The "Trans" rows in Table 3 show the s-BLEU score of Transformer trained on document-level datasets in different model sizes. Here the Base, Big model size is the same as the base, large configurations in (Vaswani et al., 2017) and the Large size is the same setting of BART large model (Lewis et al., 2020).

As the table shows, we have a better performance on different model sizes. The performance drops slightly when increasing the model size from Base to Big. Further to the Large model, there is no sharp drop either. On the Large setting, the performance gap between ours and the G-Transformer reaches +6.99/+6.36 on the two small datasets (TED and News). The results indicate that our model can better prevent training failure.

5.2 Decoding without RST Tree

When decoding using RST-Transformer, an additional step is required to obtain the dependency RST tree of the source document. This is not expected in practical application. In order that our model can still model long-distance dependencies without the help of discourse information, we added a hyper-parameter $\text{drop}_{\text{rst}}$ in the training process. It means the RST attention mask $M_{\text{rst}}$ is set to all zero with probability $p = \text{drop}_{\text{rst}}$.

Figure 5 shows the results of RST-Transformer-Mix training with $\text{drop}_{\text{rst}} \in [0, 0.3, 0.5, 0.7, 1.0]$.

When $\text{drop}_{\text{rst}} = 0$, our RST-Transformer-Mix can reach the best result with the help of the dependency RST tree. But it is failed without the tree. Once considering the absence of dependency RST tree during training ($\text{drop}_{\text{rst}} > 0$), our model can quickly adapt to this situation. Especially when $\text{drop}_{\text{rst}} = 0.5$, the RST-Transformer-Mix is better than the current state-of-the-art model even without the help of discourse structure. That means we only need to integrate discourse information during the training phase and the model could learn to model long-dependency itself.

5.3 Discourse Phenomena

We examine whether our models can capture discourse phenomena by evaluating our model on the consistency testsets (Voita et al., 2019). The consistency testsets evaluate the discourse phenomena include deixis, lexicon consistency, ellipsis (inflection and verb phrase) in English-Russian. Each testset contains contrastive examples consisting of a positive translation with correct discourse phenomenon and negative translations with incorrect phenomena. The goal is to determine whether a model is more likely to generate a correct translation than the incorrect variation. We follow the training setting proposed by Voita et al. (2019) and use both 6M sentence pairs and 1.5M document pairs from OpenSubtitles2018 (Lison et al., 2018) to train our model. As the results are shown in Table 5, our RST-Transformer-Mix obtains the best results.

5.4 Case Study

Table 4 shows an example of the deixis problem in En-De document-level MT, similar to Figure 1.
Table 4: An example of deixis problem in En-De. Given the context $S_{-1-2}$ and $S_{-2}$, the phrase "the other" in $S_{1-3}$ should be translate to "die andere" as shown in $T_{1-3}$ because the "domes" it refers to is a feminine word in German.

<table>
<thead>
<tr>
<th>Method</th>
<th>deixis</th>
<th>lexical</th>
<th>Ellipsis</th>
</tr>
</thead>
<tbody>
<tr>
<td>sent</td>
<td>50.0</td>
<td>45.9</td>
<td>40.7</td>
</tr>
<tr>
<td>concat</td>
<td>83.5</td>
<td>47.5</td>
<td>76.4</td>
</tr>
<tr>
<td>MCN</td>
<td>61.3</td>
<td>46.1</td>
<td>48.3</td>
</tr>
<tr>
<td>G-Transformer</td>
<td>89.9</td>
<td>83.6</td>
<td>-</td>
</tr>
<tr>
<td>RST-Transformer-Mix</td>
<td>91.7</td>
<td>83.6</td>
<td>79.1</td>
</tr>
</tbody>
</table>

Table 5: Accuracy (%) of consistency testsets.

6 Related Work

Structure-aware NMT is a well-studied topic in the sentence-level translation and the structure information is proven to be useful. Many approaches claim performance improvements by using treebank syntax (Sennrich and Haddow, 2016; Eriguchi et al., 2016; Bastings et al., 2017; Aharoni and Goldberg, 2017). Other approaches incorporate syntactic information in NMT models relatively indirectly (e.g. multi-task learning (Luong et al., 2015; Nadejde et al., 2017; Eriguchi et al., 2017; Hashimoto and Tsuruoka, 2017)). Unlike these syntactically-aware NMT methods mentioned above, Marcheggiani et al. (2018) and Song et al. (2019) investigate semantic role labeling and abstract meaning representation on NMT by GCN (Kipf and Welling, 2016) and GRN (Zhang et al., 2018) respectively.

On the contrary, it is still the mainstream practice to treat documents as plain text in document-level machine translation (Voita et al., 2018; Wang et al., 2019; Miculicich et al., 2018; Maruf et al., 2019; Xu et al., 2020b; Ma et al., 2020; Zheng et al., 2020; Liu et al., 2020; Bao et al., 2021). Only little work has done with structured information. Xu et al. (2020b) proposed a graph-based approach where graphs are constructed according to inter-sentential (adjacency, dependency) and intra-sentential (lexical consistency, coreference) relations. Compared with their work, we use the dependency RST tree supported by linguistic theory and introduce the tree by attention mask. Xiaomian and Chengqing (2020) and Chen et al. (2020) also enrich input word embeddings with path embeddings based on RST trees. Unlike their methods, we integrate RST trees into the encoder and decoder of the conventional transformer model via an attention mask.

7 Conclusion

In this paper, we present a novel discourse structured information enhanced approach for document-level NMT. We modify the Transformer architecture to integrate the dependency RST tree of the source document by attention mask in encoder and decoder. Experiments show that our method gives state-of-the-art results compared to existing models under pretraining and non-pretraining settings. Our further analysis also shows that our model achieves nearly the same results even when decoding without the discourse information.
References


