RST Discourse Structure Improves Story Ending Generation

陳 宏 [1,3]　西田 典起 [1]　朱 中元 [1]　岡崎　直観 [2,3]　中山 英樹 [1,3]
[1] 東京大学　大学院情報理工学系研究科
[2] 東京工業大学 情報理工学院 情報工学系
[3] 産業技術総合研究所人工知能研究センター
[1] {chen, nishida, shu, nakayama}@nlab.ci.i.u-tokyo.ac.jp
[2] okazaki@c.titech.ac.jp

1 Introduction

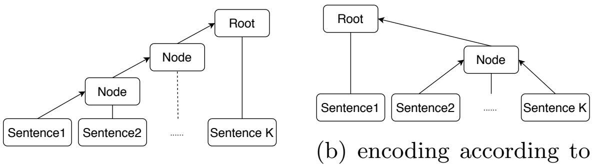
Story ending generation is a crucial task for automatic story generation, which completes a story with a reasonable ending sentence based on given contexts. A typical example can be found as follows:

Context	Gina misplaced her phone at her grandparents. It wasn't anywhere in the living room. She realized she was in the car before. She grabbed her dad's keys and ran outside.
Ending	She found her phone in the car.

Conventionally, an encoder-decoder sequence generation model is applied for predicting the ending sentence. Here, the encoder turns the source sentences into fixed-length vectors, then the decoder produces the resultant sequence with the help of attention mechanism.

Two existing works can be found in literature on this task. Each work puts an unique assumption on the contextual structure of sentences in a story. The most common modelling approach [14, 5] is to concatenate the contextual sentences, forming a long sequence and encode it. In this approach, all contextual sentences can be directly accessed by the decoder. Therefore, the sentences are considered equally important. To capture the order and the relationships between adjacent sentences, Guan et al. [4] represents context clues by incrementally encoding contextual sentences from left to right to build a context vector. The decoder then observes the context vector containing information from all previous sentences. In this approach, latter sentences are considered as more important. Fig. 1a shows the implicit discourse structure in this case.

Story Cloze Test is a task that given contextual sentences, one has to select one correct ending from given candidates. Srinivasan et al. [11] found that the model using only the last sentence in the context achieves comparable performance with the models using full context.



(a) left-to-right encoding

(b) encoding according to RST discourse structure

Figure 1: Two different encoding strategies. Root nodes denote the feature of contextual sentences. Each node is an intermediate feature encoded so far.

To investigate whether their finding is also appropriate in story ending generation, we conducted a preliminary experiment, which will be detailed in Section 5.2. As a result, we find that the first sentence almost always contain topical or background information helpful for generating a better ending. Based on the preliminary results, we consider the topic sentence as the most important context.

Therefore, to generate more coherent endings, we propose to find the importance of each sentence in the context. Discourse parsing is such a promising method to automatically uncover coherence structures (i.e., trees) for multiple sentences. It has been successfully applied in down-stream tasks such as question answering [1], and automatic essay scoring [8]. Rhetorical Structure Theory (RST) [7] is one of the most widely accepted theories of discourse structure. Figure 1b illustrates an encoding strategy according to RST-based discourse structure. The hierarchy of sentences represents their relative importance [6], which we consider to be useful for story ending generation.

Our contribution can be summarized as two folds. First, we propose a simple and efficient encoding strategy that extends Transformer [12] to exploit RST-based discourse structures for encoding contextual sentences in a story. Second, we perform em-

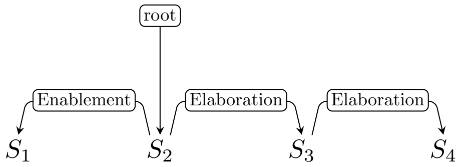


Figure 2: An example of RST discourse dependency structure.

pirical evaluations that confirm the effectiveness of RST discourse structures to producing better story endings.

2 Related Work

Story Ending Generation is the task to complete a story, which has been approached by several neural based methods in the recent years. The ROCstory corpus often used in this task was designed for Story Cloze Test [9]. In this test, the model needs to understand the logic, find clues and compare two candidates. However, story ending generation is far more challenging than making choices because the model needs to generate both resonable and fluent sentence.

Early attempt [14] relied on reinforcement learning method on copy mechanism in order to enhance the probability of selecting the word from the context. Recently, Gupta et al. [5] aimed to generate more interesting sentences with different conditions including sentiment, length and keyphrase. Guan et al. [4] represented context clues by incremental encoding, and leverages common-sense knowledge by multi-source attention to effectively capture the coherence and logic of story.

3 Rhetorical Structure Theory

Rhetorical Structure Theory (RST) [7] assumes that a tree structure can be derived for coherent text. In RST, leaf nodes are clauses-level non-overlapping text segments (or elementary discourse units, EDUs), and consecutive text segments are combined by rhetorical relations recursively to form lager segments. The rhetorical relations have two components: Nucleus and Satellite. A nucleus corresponds to the more salient segment of the relation, while a satellite corresponds to more supporting one.

In this paper, we employ RST discourse dependency structures [13], such as the one in Figure 2, for story ending generation. To obtain RST parses for context stories in our experiments, we used an off-the-shelf RST parser [3].

4 Proposed Encoder Model

We propose a novel context encoding strategy based on RST discourse dependency graphs for story ending generation. Fig. 3 gives an overview of the encoder model architecture, which contains two components. The first component is the original word-level Transformer encoder. The second component applies K layers self-attention mechanism for encoding sentence-level information according to a specific tree structure.

Word-Level Encoder In this paper, we use S_1, S_2,...,S_N to denote contextual sentences, where each sentence S_i may have different number of tokens. In the first stage, a regular Transformer encoder is applied to encode the words in each sentence into word-level feature vectors as shown in Fig. 3 as the first layer of vectors.

Sentence-Level Encoder Similar to BERT [2], We use the word feature corresponding to the start token $\langle \text{sos} \rangle$ as the feature vector for each sentence. Now, we have N sentence-level feature vectors. Next, we feed these sentence vectors into K self-attention layers. We apply binary masks constructed according to the tree structure.

In detail, when computing the weighted summarization in the attention, each S_i is assigned a mask $m_i \in \{0,1\}^N$. Each mask is an N-dimensional binary vector. An example of m_i can be $[1,0,0,0]^\top$. The value m_{ij} is 1 when S_i depends on S_j based on the RST discourse dependency tree. For the example tree in Fig. 2, all mask $\mathbf{m} = m_1, \ldots, m_N$ forms

$$\mathbf{m} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

When applying self-attention for K iterations, we denote the hidden state for S_i in k-th iteration as h_i^k. In each iteration, the hidden state h_i^k is updated as the following weighted summarization:

$$h_i^{k+1} = \sum_{j=1}^{N} w_{ij}^k h_j^k \tag{2}$$

$$w_{ij}^k = \frac{\exp(f(h_i^{k-1}, h_j^{k-1}))}{\sum_{j=1}^{N} \exp(f(h_i^{k-1}, h_j^{k-1}))} \tag{3}$$

$$f(h_i^k, h_j^k) = \begin{cases} \text{Score}(h_i^k, h_j^k), & \text{if } m_{ij} = 1 \\ -\infty, & \text{otherwise} \end{cases} \tag{4}$$

where w_{ij}^k is the attention weight that matches S_j to S_i in the k-th iteration. The score function Score$(\cdot)$

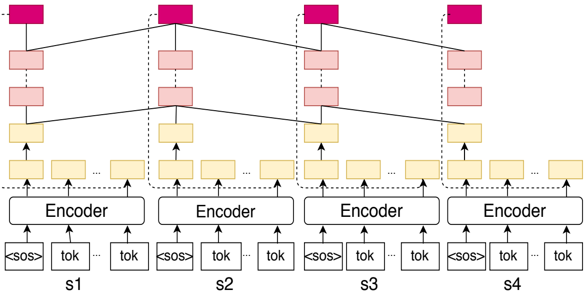


Figure 3: Architecture of proposed encoder model.

gives the unnormalized weight, which is same as the original self-attention model. In this way, leaf node sentences can only attend to itself, while a non-leaf nodes can take in account sentence-levels features from their children. Ideally, the resultant sentence-level feature maintains the hierarchical information according to the tree structure.

After updating the feature vector h_i for each sentence, we merge it back to the word-level feature vectors with simple addition. Therefore, in this modelling approach, the Transformer decoder still observes word-level feature vectors, however, enhanced by the sentence-level encoder.

5 Experiment

5.1 Dataset and Settings

ROCStory dataset was designed for Story Cloze Test. We follow the previous work [4, 5], reforming the dataset by using the right ending candidate as target. The training corpus contains 98,162 five-sentence stories. First four sentences are the context (input), and the last sentence is the ending sentence (output). Besides, we have 1,874 stories in the test dataset and for each story, one gold ending sentence is given.

We use GloVe.6B as word vectors. The vocabulary size is set to 20,000 and the word vector dimension is 300. For the transformer model, the self attention layer and the heads are both set to 6. These settings were applied to all models.

5.2 Preliminary Experiment

As aforementioned, we conduct an experiment to roughly analyze the importance of each sentence in the context. Since there are four sentences in each context in ROCStory dataset, we train a Transformer model using only the first sentence, the second sentence, the third sentence and the fourth sentence separately as input. Table 2 shows that the first sentence is more important among all sentences.

This founding confirms that to generate a coherence ending, the topic sentence is necessary so that RST discourse dependency structure is valuable in this task because the head in the RST discourse dependency structure is always the topic sentence.

5.3 Baselines

In this experiment, we set several baselines including the original Transformer and our proposed model with different tree structures.
Transformer (T): Sentences are concatenated into a long sentence and trained by regular Transformer.
Transformer* (T*): With the aforementioned model, all the masks are set to 1, thus each sentence contributes equally to the model.
Transformer Random (T (rand)): In order to prove that the RST structure is helpful, we assign random mask to each context.
Transformer Last (T (last)): Similar to the previous work [4], each sentence is encoded to its next sentence.
Transformer First (T (first)): The sentences all connect to the first sentence based on the expectation that the first sentence always provide topic, which helps generating more coherent ending.
Transformer RST (T (RST)): The model is equipped with RST discourse dependency structure which is our proposal.

5.4 Results

We conduct the automatic evaluation on the 1,874 stories in the test dataset. We adopt BLEU, METEOR, ROUGE-L, CIDEr and perplexity (PPL) to evaluate the generation performance. Except for the Perplexity score, higher scores are better. To further evaluate the quality of generated stories, Sagarkar et al. [10] proposed to evaluate overall, relevance and interestingness score with trained models. We also adopt their models to evaluate our stories.

As shown in Table 1, our RST based transformer model achieves the best score on perplexity (PPL), which means our proposed model assigns higher probability to the gold sentence. Moreover, we achieve a 0.67% improvement from the second place in the CIDEr score which means this model contains more consensus of how people writes the ending. In the metrics evaluated by Sagarkar's model, we get the best performance on overall and relevance.

Table 3 shows examples of the generated ending with different models. Please note that as both T(first) and T(RST) consider the first sentence as root, they generate similar sentences. The results reveals that using different tree structure can indeed influence the output.

Model	BLEU@1	BLEU@2	Meteor	ROUGE@L	CIDEr	PPL	O	R	I
T	0.2275	0.0848	0.0982	0.2429	0.3586	9.89	5.845	6.013	**5.180**
T*	0.2310	0.0846	0.0985	0.2450	0.3638	9.62	5.824	6.003	5.126
T(last)	0.2463	0.0900	0.0993	**0.2564**	0.3687	9.67	5.820	6.020	5.098
T(first)	0.2391	0.0912	**0.1021**	0.2525	0.3877	9.55	5.761	5.917	5.112
T(random)	0.2265	0.0829	0.0976	0.2395	0.3584	9.78	5.875	6.046	5.124
T(RST)	**0.2576**	**0.0952**	0.0938	0.2547	**0.3941**	**9.51**	**5.880**	**6.160**	5.174

Table 1: Automatic and neural based (O, R, I) results. O, R, and I respectively denote Overall score, Relevance score and Interestingness score [10].

S	B@2	Meteor	R@L	CIDEr	PPL
1	0.0637	**0.0846**	0.2176	**0.2613**	**11.39**
2	0.0527	0.0707	0.2110	0.1479	13.36
3	0.0573	0.0739	0.2155	0.1719	13.36
4	**0.0668**	0.0755	**0.2237**	0.1949	12.81

Table 2: Automatic results in the preliminary experiment. S denotes the index of the sentence.

Context	Peter wished to show his daughter his favorite Christmas song. He played her a video of the song. Next, he let her listen to the sound track. The last time he decided to sing her the song.
T	He loved it so much that he gave it to his daughter.
T(*)	peter was so happy that he cried tears of joy.
T(last)	Peter loved the song so much that she cried.
T(first)	Peter's daughter loved the song.
T(rand)	He was so happy when he gave her the song to his daughter.
T(RST)	She loved the song.
Gold	Peter's daughter enjoyed the music.

Table 3: Examples generated by different models.

6 Conclusion

In this paper, we leverage the RST discourse structure for story ending generation task. The results show that the RST discourse structure helps the model to capture the importance of sentences. However, several limitations shall be concerned. First, due to that RST was not originally designed based on the story structure, ideally, we need to develop new relation labels and structure theories to fit the task. Second, according to recent works, models are weakened when facing longer stories. We conjecture that in the long story generation, RST structure further helps the model to understand the context. We leave these points for our future work.

Acknowledgements

This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

[1] A. R. Akula. A novel approach towards building a generic, contextual and portable nlidb system. 2015.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *ACL*, 2019.

[3] V. W. Feng and G. Hirst. Text-level discourse parsing with rich linguistic features. In *ACL*, 2012.

[4] J. Guan, Y. Wang, and M. Huang. Story ending generation with incremental encoding and commonsense knowledge. In *AAAI*, 2019.

[5] P. Gupta, V. B. Kumar, M. Bhutani, and A. W. Black. Writerforcing: Generating more interesting story endings. In *ACLW*, 2019.

[6] A. Louis, A. Joshi, and A. Nenkova. Discourse indicators for content selection in summarization. In *SIGDIAL'10*, 2010.

[7] W. C. Mann and S. A. Thompson. Rhetorical Structure Theory: Towards a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[8] E. Miltsakaki and K. Kukich. Evaluation of text coherence for electronic essay scoring systems. *Natural Language Engineering*, 10(1):25–55, 2004.

[9] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *NAACL-HLT*, 2016.

[10] M. Sagarkar, J. Wieting, L. Tu, and K. Gimpel. Quality signals in generated stories. In **SEM*, 2018.

[11] S. Srinivasan, R. Arora, and M. Riedl. A simple and effective approach to the story cloze test. In *ACL*, 2018.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[13] Y. Yoshida, J. Suzuki, T. Hirao, and M. Nagata. Dependency-based discourse parser for single-document summarization. In *EMNLP*, 2014.

[14] Y. Zhao, L. Liu, C. Liu, R. Yang, and D. Yu. From plots to endings: A reinforced pointer generator for story ending generation. In *NLPCC*. Springer, 2018.