# RST Discourse Parsing as Text-to-Text Generation

Xinyu Hu and Xiaojun Wan

***Abstract*—Previous studies have made great advances in RST discourse parsing through specific neural frameworks or features, but they usually split the parsing process into two subtasks and heavily depended on gold discourse segmentation. In this article, we introduce an end-to-end method for sentence-level RST discourse parsing via transforming it into a text-to-text generation task, which can also be simply applied to document-level parsing. Our method unifies the traditional two-stage parsing and generates the parsing tree directly from the input text through our constrained decoding and postprocessing algorithms, without requiring a complicated model. Moreover, the discourse segmentation can be simultaneously generated and extracted from the parsing tree. Experimental results on the RST Discourse Treebank demonstrate that our proposed method outperforms existing methods in both the tasks of discourse parsing and segmentation. We further carry out ablation studies and more targeted comparisons with traditional patterns to analyze our method in more detail. Considering the lack of annotated data in RST parsing, we also create high-quality augmented data and implement self-training, which further improves the performance of our method.**

***Index Terms*—Discourse parsing, rhetorical structure theory, text generation.**

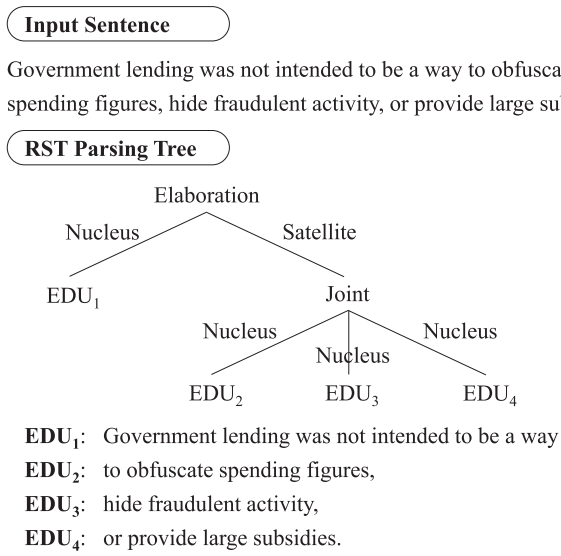



Fig. 1. Example from RST Discourse TreeBank.

## I. Introduction

DISCOURSE parsing involves determining the structure of elementary units forming a discourse and how they are connected with each other. In a coherent text, units are often organized logically and semantically with certain relation. Early studies have demonstrated that discourse parsing can benefit various downstream NLP tasks, including sentiment analysis [1], [2], summarization [3], [4], question answering [5] and machine translation evaluation [6].

RST parsing, based on Rhetorical Structure Theory [7], is one of the most common and influential parsing methods in discourse analysis. According to RST, a text is first segmented into several clause-like units as leaves of the corresponding parsing tree, called elementary discourse units (EDUs). Through certain rhetorical relations among adjacent spans, such as elaboration and joint, underlying EDUs or larger text spans are recursively linked and merged to form their parent nodes, representing the concatenation of them. Finally, a hierarchical tree structure is constructed. Besides rhetorical relations, sibling nodes in the parsing tree contain a kind of nucleus-satellite relations to show who is more central or equal to the discourse structure. Fig. 1 shows an RST parsing tree for a sentence from the RST Discourse TreeBank [8], which is the most common discourse corpus.

In the past, various approaches have been proposed for both document-level and sentence-level RST parsing, which can be mainly divided into bottom-up and top-down methods. Earlier work like transition-based approaches utilized the representation learned through manually-designed features or neural networks to build shift-reduce parsers [9], [10]. The whole parsing tree is gradually built in a sequence of actions, including shift and reduce. Then, benefiting from the development of neural networks, top-down approaches [11], [12], [13], [14] made use of the pointer network [15] to segment text into shorter units recursively until no more units can be generated.

Although many advances have been made in RST parsing, the real performance of existing methods may be far from satisfactory. Most studies before followed the traditional settings to split the parsing process into two stages, namely segmenting EDUs and building parsing trees. They employed their models only on the second stage and treated the gold EDU segmentation as a requisite, which is, however, infeasible in real application scenarios. The segmenting model trained in the first stage can

Manuscript received 5 May 2022; revised 22 May 2023, 6 July 2023, and 24 July 2023; accepted 1 August 2023. Date of publication 18 August 2023; date of current version 5 September 2023. This work was supported in part by the National Key R&D Program of China under Grant 2021YFF0901502, in part by the National Science Foundation of China under Grant 62161160339, in part by the State Key Laboratory of Media Convergence Production Technology and Systems and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zhijian Ou. *(Corresponding author: Xiaojun Wan.)*

The authors are with the MOE Key Laboratory of Computational Linguistics, Wangxuan Institute of Computer Technology, Peking University, Beijing 100871, China (e-mail: huxinyu@pku.edu.cn; wanxiaojun@pku.edu.cn).

Digital Object Identifier 10.1109/TASLP.2023.3306710

generate automatic segmentation as a substitute, but the performance of those parsing methods would drop a lot accordingly. This may be caused by errors in segmenting models transmitting to the parsing stage. Moreover, previous methods relied on additional features or complicated frameworks for different parts of parsing like label prediction, which did not take full advantage of knowledge in the task.

In this article, we focus on sentence-level RST parsing and introduce a simple end-to-end method that can generate the target parsing tree directly from the corresponding text. It is beneficial since sentence-level parsing is essential and serves as a basic step in some document-level parsers [16], [17]. Therefore, the improvement of sentence-level parsing may promote further progress in discourse parsing. Furthermore, our method can be directly applied to document-level parsing and shows superiority compared with previous approaches.

Our proposed method converts RST parsing into a text-to-text generation task by reformulating the parsing tree into a natural language sequence. The information contained in text content, hierarchical structures, and relation labels in the parsing tree can be integrated and learned together by the generation model. This benefits parsing performance, as proved by our ablation study. With constrained decoding and postprocessing algorithms, predicted sequences can be restored and converted back into parsing trees. Experimental results demonstrate that our method outperforms previous work without using gold segmentation. In addition, EDU segmentation can be generated simultaneously during parsing, which has even better performance than most other segmenting models specifically trained on this task. In view of the lack of annotated data in RST parsing, we also attempt to generate high-quality augmented data to obtain extra enhancement.

Our primary contributions are as follows:

- We propose a simple but effective end-to-end approach to sentence-level RST discourse parsing without using gold segmentation and additional auxiliary information, which can be directly applied to document-level parsing.
- Our method generates the parsing tree with the EDU segmentation simultaneously through our constrained decoding and postprocessing algorithms and outperforms previous models on both tasks.
- We obtain high-quality augmented data for RST parsing through our filtering rules and further improve the performance of our model by means of self-training.

## II. Related Work

Discourse parsing describes the hierarchical tree structure of a text and can be used in quality evaluations like coherence and other downstream applications. However, parsing from scratch was difficult, so traditional research usually split the parsing process into two stages to simplify the problem. They evolved into two separate tasks, namely discourse segmentation and RST parsing based on given EDU segmentation. The latter is what most of the existing research on discourse parsing has really been devoted to.

In the past, many methods have been proposed for discourse segmentation, and the performance of the state-of-the-art segmenting model is close to that of a human. SegBot [18] used a pointer network [15] to select the EDU boundaries and [11] introduced a more efficient method with a neural encoder-decoder architecture. Wang et al. [19] regarded the segmentation as sequence tagging based on the BiLSTM-CRF framework. Recently, [20] proposed a transformer-based model with enhanced contextualized word embeddings and hand-crafted features, achieving the best performance in discourse segmentation.

On the other hand, more attention has been paid to discourse parsing at both the sentence level and document level, and some relevant review articles [21], [22] have emerged recently. Previous relevant methods can be mainly divided into two classes: top-down and bottom-up paradigms. In earlier studies, bottom-up methods have been first applied since various kinds of hand-engineered features were mainstream tools and suitable to represent local information. Soricut et al. [23] first proposed a bottom-up CKY-like approach with syntactic and lexical features for sentence-level parsing. Other models with CKY-like algorithms [24], [25], [26], [27] utilized diverse features to learn the probability scores for different subtrees and searched all possible parsing trees to find the most likely one for a text. Although these methods achieved high accuracy, they suffered from slow parsing speed.

Another common bottom-up method is the transition-based parser, which generates the RST parsing tree during a sequence of shift and reduce action decisions. Ji et al. [9] introduced a neural shift-reduce parser with representation learning methods. Wang et al. [16] proposed a two-stage parser based on SVMs with plenty of features. Then [10] trained a transition-based parser with implicit syntactic features from dependency parsing and achieved great improvement. Despite their good efficiency, these methods lack sufficient lookahead guidance for each decision and may not achieve the best result in the long run.

Thanks to the recent advancement of neural methods, it is possible to represent the text effectively in a global view, which promoted top-down parsers. Instead of constructing the whole parsing tree through merging local text spans or subtrees in bottom-up methods, top-down parsers normally segment the original text into shorter units recursively until no more units can be generated, and then the parsing tree has been built. Lin et al. [11] first presented a seq2seq model for sentence-level RST parsing based on pointer networks [15] and [12] improved it with a hierarchical structure to model parent-child and sibling relation. Then [13] extended their methods to document-level RST parsing and treated it as a recursive split point ranking task. Kobayashi et al. [17] constructed subtrees for three granularity levels of text and merged them together to form the whole parsing tree, reducing the difficulty of parsing.

Despite the success of top-down models, most of them still utilized gold EDU segmentation as a necessity and dropped a lot in performance when using automatic segmenting models. Recently, many studies [28], [29], [30], [31], [32], [33], [34], [35], [36] have made sustained progress on document-level

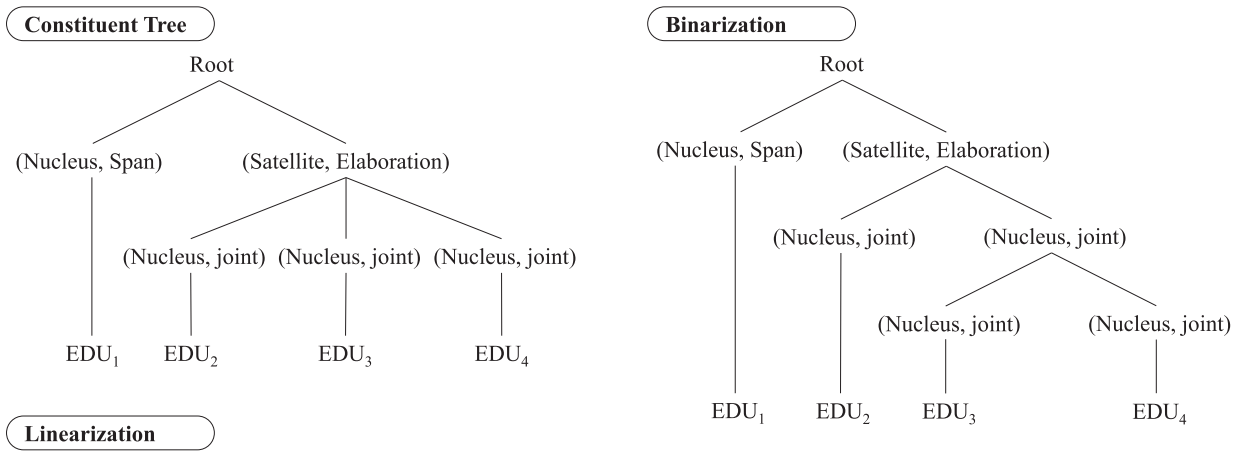



Fig. 2. Process of reformulation for the RST parsing tree from Fig. 1 according to our method.

discourse parsing, but most of them proposed methods based on the gold EDU segmentation. However, it is more practical that the parsing tree should be constructed directly from the input text. And the two-stage process may lead to error propagation from segmenting to parsing. On the other hand, [34] and [14] introduced end-to-end parsing models for document-level and sentence-level parsing, respectively. However, they relied on different and complicated frameworks for structure and label prediction. Our end-to-end approach, instead, transforms RST parsing into a text generation task, eliminating the need for specific frameworks, and improves the performance of both discourse parsing and segmentation tasks.

## III. Our Method

Over the past year, a new paradigm based on powerful pretrained language models has emerged and brought remarkable improvement in many areas. Instead of adapting pretrained language models to different downstream tasks through specific network layers and objective engineering, now in prompt learning [37], downstream tasks are reformulated close to the pretraining tasks. Similar approaches have been applied to parsing tasks like constituent parsing [38], [39]. However, it still remains a great challenge for more complex and longer data structures, like RST parsing trees.

Motivated by the idea above, we propose a method to reformulate the parsing tree into the form of a linear sequence so as to utilize existing seq2seq models. Significantly, our end-to-end model implements the parsing directly from the input texts into the RST trees, avoiding a two-stage process. We show that our new text-to-text task can make great use of the latent knowledge in pretrained models like T5, without additional features or neural frameworks. In addition, we use constrained decoding to ensure well-formed output sequences that can be restored and evaluated through the postprocessing algorithm. Furthermore, our method can also be simply generalized to the tasks of discourse segmentation and document-level discourse parsing without great modifications. The detailed procedures will be introduced in the following sections.

### A. Binarization

In the original RST Discourse TreeBank, RST parsing trees are stored as a set of hierarchically organized text spans together with their nuclearity and relation labels. Daniel Marcu [40] first formally encoded the RST parsing tree in the form of a constituent tree, as shown in Fig. 2, which was followed by the majority of subsequent parsing methods. As in previous studies on the RST-DT, we also construct the constituent tree and then binarize the tree using right-branching. The binarization has been a common assumption [23], [41] and can help reformulate parsing trees more regularly and suitably for training and evaluation since more restrictions are imposed.

The nodes on the tree that have more than one sibling node must contain multinuclear relations like joint, because they are the only ones allowed to link more than two text spans. In view of these sibling nodes sharing the same nuclearity and relation labels, the new binary tree can be built by adding the same intermediate nodes and constructing subtrees recursively. The new constituent tree after the process of binarization can be found in Fig. 2.

### B. Linearization

Then, based on the priority level contained in brackets, we attempt to represent hierarchical architecture by nesting several pairs of brackets. The linearization is carried out from the bottom up according to postorder traversal. We replace each leaf that represents a single EDU with a sequence comprised of a left bracket, text content, a right bracket, and its nuclearity and relation labels. Blank characters are added to each interval between different elements.

As for intermediate nodes, we perform the same process except that the concatenation of new representations of two child nodes serves as the text content. Since the root does not

contain any labels, it simply merges two child nodes with a pair of outermost parentheses. The postorder traversal ensures that intermediate nodes will be processed after their child nodes are updated, and the root is the last one to be considered, resulting in the final linear sequence of the parsing tree.

Different from the linearization method from [42], we reformulate the whole parsing tree instead of each single EDU. Moreover, considering that [43] encouraged the use of the entire input to promote the performance, our linear sequence is designed to contain a complete copy of the corresponding input text. And the full specifications of nuclearity and relation labels are retained to make full use of the latent knowledge since they must be learned during pretraining and can be understood by language models. We have also experimented with simplified forms of reformulated output sequences, but the current method does achieve better performance.

Through these steps, the format of reformulated sequences is unified and normative, with each pair of inner brackets containing text content followed by a nuclearity label and a relation label in turn. And the postorder traversal guides the model to understand the text content before predicting labels, which is in accordance with the way of humans. Besides, we use square brackets in linearization to avoid confusion since the input text itself may contain parentheses. Finally, the target linear sequence of the RST parsing tree in Fig. 1 is shown at the bottom of Fig. 2.

### C. Seq2Seq Training

Since the input and new output of the task are both sequences, RST parsing can thus be trained or finetuned on any generation model as a text-to-text generation task. Mathematically, the model calculates the conditional probability of a target output sequence $y$ and finds the most probable sequence $\hat{y}$, same as other seq2seq problems:

$$\hat{y} = \arg\max_{y} \prod_{n=1}^{|y|} P_\theta \left( y_n \mid y_{<n}, x \right) \tag{1}$$

where $x$ and $\theta$ denote the input sequence and parameters of the model, respectively.

As described in the previous section, our reformulated output sequences are designed to be close to natural language texts. With seq2seq pretrained models like T5 [44], which involve similar text-to-text settings, we suppose the related knowledge can be better transferred to our new RST parsing task. Despite the lack of annotated data in discourse parsing, our method works well without extra complicated frameworks or features. In the meantime, the subtasks of discourse segmentation and prediction of structure and label are all integrated into the single process of text generation, which is superior to other approaches in terms of efficiency. According to our ablation study, the integration of label information has been proven to benefit the parsing performance of seq2seq models.

### D. Constrained Decoding

During the process of inference, a seq2seq model normally generates the target output token-by-token according to the probability distribution. However, since our output sequence is supposed to observe the linearization formats that we designed before, traditional greedy decoding or beam search algorithms may lead to format errors, including wrong content and brackets. To address this problem, we employ the constrained decoding methods [45], [46], [47] to constrain the selection of tokens in each inference step. Specifically, we dynamically modify the candidate vocabulary set in beam search according to the current generated sequence and our designed formats.

Tokens in an output sequence can be classified into seven categories, and their possible successors based on the linearization format are listed as follows:

1) The token <bos>: an open bracket.
2) The token <eos>: none. (It means current inference has been finished.)
3) A nuclearity label: a relation label.
4) A relation label: an open bracket or a close bracket.
5) An open bracket: an open bracket or the next word in the input sequence that has not yet been generated.
6) A close bracket: a nuclearity label or the token <eos>.
7) A word in the input sequence: a close bracket or the next word in the input sequence if it exists.

A part of the constrained decoding process of the example from Fig. 1 can be found in Fig. 3. In addition, we can also control the ending of the generated sequence. When the current token is a close bracket and the content of the input sequence is contained in the current predicted output sequence, we utilize the recursive algorithm introduced in the next subsection to determine whether the whole parsing tree has been restored successfully. The token <eos> will be selected as the next step only if the parsing tree can be constructed.

### E. Postprocessing

In the postprocessing, we employ a recursive algorithm on the generated sequence based on the designed format in Section III-B. The final RST constituent tree is extracted and reconstructed through continually merging bottom text spans until only the root remains. Our postprocessing is concise and model-independent, so it has good transferability and can be easily followed. Although we have employed constrained decoding to ensure the basically correct format of generated sequences, minor format errors like the unmatched number of open brackets are still unavoidable. They may cause potential performance degradation, but the postprocessing just takes charge of the extraction and retains the original performance of the parsing model.

Therefore, our algorithm depends mainly on close brackets, and open brackets are used to determine whether the current close bracket is the boundary of an EDU. Benefiting from the binarization, it is clear that each combination will involve exactly two leaf spans, and each merging action is deterministic. If the current close bracket yields an EDU, then it will be put into the stack; otherwise, the top two text spans in the stack need to be merged. More details are shown in Algorithm 1. The output sequence is finally converted into a set of connected constituents for evaluation with the ground-truth RST parsing tree.

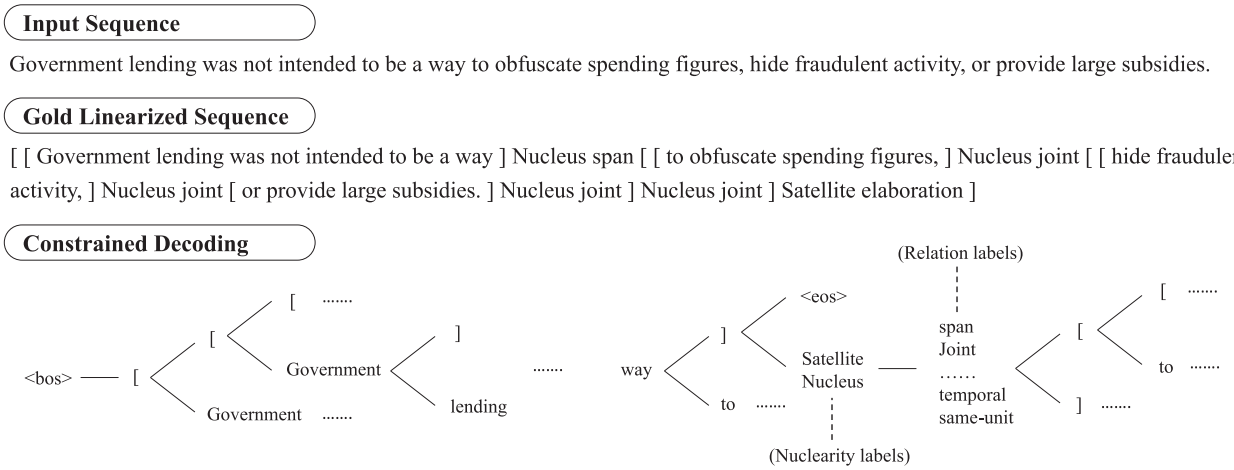



Fig. 3. Partial process of constrained decoding for the example from Fig. 1 based on the linearization format. Solid lines connect the current tokens with the next possible tokens during inference.

**Algorithm 1:** Restoration of the Constituent Tree.

**Input:** Target sequence S, input sentence I
Initialization: T = [], nodes = [], i = 0
Seq_unit = S.split(']'),
$U_k$ = Seq_unit[k].split('['), $0 \leq k <$ len(Seq_unit)
**repeat**
  **if** '[' in Seq_unit[i]**then**
    cur_label = $U_{i+1}$[0]
    cur_text = $U_i$[-1]
    push(nodes, (cur_text, cur_label))
  **else if** len(nodes) > 1 **then**
    ($text_1$, $label_1$) = pop(nodes)
    ($text_2$, $label_2$) = pop(nodes)
    push(T, ($text_1$, $label_1$, $text_2$, $label_2$))
    cur_label = $U_{i+1}$[0]
    cur_text = $text_1$ + $text_2$
    push(nodes, (cur_text, cur_label))
  **end if**
  i = i + 1
**until** I = top(nodes).text
**Output:** T as the set of connected constituents in the constituent tree

### F. Document-Level Parsing

Furthermore, we can directly extend and apply our end-to-end method for sentence-level discourse parsing to the document-level parsing task. Since we focus on sentence-level parsing in this article, we do not propose novel changes and leave them to future work. In document-level parsing, the samples become even longer and always contain a large number of EDUs and complicated structures, making it more difficult and challenging for our method to deal with them.

Based on this point, we employ some modifications and optimizations during the training and inference processes to adapt to the long text. To avoid the cold start situation in many long document tasks [48], we first train a sentence-level parser and then continue to train it for document-level parsing. Since each sentence in the training set of sentence-level parsing is contained within a document in document-level parsing, the model can learn some information from subtrees in advance. Moreover, documents in the training set are assigned different loss weights that are directly proportional to the number of EDUs they contain. It helps those long and difficult samples be paid more attention during training through larger loss weights. The loss function $L$ at document level is calculated as follows:

$$L = \sum_{m=1}^{N} \frac{e_m}{\sum_{i=1}^{N} e_i} \sum_{n=1}^{|y^m|} -\log P_\theta \left(y_n^m \mid y_{<n}^m, x^m\right) \tag{2}$$

where $e_i$ denotes the number of EDUs contained in the $i$th sample $(x^i, y^i)$, and $N$ is the size of the training set.

In the experiments, we choose the pretrained language model with relative positional embeddings, like T5, which allows direct extrapolation to longer sequences. As a result, the long inputs in document-level parsing can be dealt with the same as in the sentence-level task. More details about the pretrained models we use will be explained in Section IV.B. During the inference process, the constrained decoding and beam search are still employed, with some additional discourse tagging rules following the official reference manual. In particular, if an uncompleted output sequence during inference contains any conflicting labels, it will be directly discarded in beam search. And we have also attempted these optimization strategies in sentence-level discourse parsing, but the improvement is not significant. We suppose it is because the sentence-level task is much less complicated and difficult, and contains relatively adequate training data compared with the document-level task.

### G. Discourse Segmentation

Our end-to-end model aims at parsing the input texts directly into the RST trees without going through the task of discourse

**Input Sequence**

Government lending was not intended to be a way to obfuscate spending figures, hide fraudulent activity, or provide large subsidies.

**Segmentation Reformulation**

[ Government lending was not intended to be a way ] [ to obfuscate spending figures, ] [ hide fraudulent activity, ] [ or provide large subsidies. ]

Fig. 4. Our reformulation for the example from Fig. 1 in the task of discourse segmentation.

segmentation. However, we can effortlessly extract the segmentation predictions from the RST sequences generated by our trained parsing model, since the discourse segmentation can be served as part of the parsing task. These results are some kinds of by-products with no extra work.

On the other hand, our reformulation method can also be simply transferred to the discourse segmentation task. Similar to the discourse parsing, we linearize the segmentation with the brackets as the target sequence of the corresponding seq2seq task. Fig. 4 illustrates the linearized output sequence for discourse segmentation, which has no hierarchical structures or label information. And we can independently train our segmentation model with a similar text-to-text pattern and pretrained seg2seq models.

## IV. Experimental Setup

In this section, we introduce the datasets and pretrained language models used in our experiments. The training settings and evaluation metrics are also demonstrated in detail.

### A. Datasets

We implement our experiments on RST Discourse TreeBank (RST-DT) [49], which is the standard dataset also used by other studies. It is the largest available discourse corpus and contains 385 Wall Street Journal English articles selected from the Penn Treebank [50], 347 documents (7673 sentences) for training and 38 documents (991 sentences) for testing.

To construct the dataset for sentence-level RST parsing, we follow the same preprocessing step as [11], [12], [51] to select sentences that consist of several EDUs and form the subtrees of document-level parsing trees. In all, we obtain 7321 sentences for training and 951 for testing, together with their parsing trees for the RST parsing task, which is the same scale as reported in previous studies.

As for discourse segmentation, we obtain two predictions from the direct extraction and the independently trained segmentation model, respectively, as described in Section III-G. In the case of the former, there is no need for an extra training set, while for the latter, we utilize the entire 7673 sentences in the training set, the same as those in the previous work [11] and larger than that of the sentence-level RST parsing. During evaluation, we also keep the test set the same as traditional usage in two settings for fair evaluation, which contains the full 991 sentences. For all the tasks, we randomly select 10% of the training data for hyperparameter tuning. An overview of these datasets is shown in Table I.

TABLE I
The Statistics of Datasets for Different Tasks in Our Experiments

| Dataset | #Training | #Test |
|---|---|---|
| Document-level discourse parsing | 347 | 38 |
| Sentence-level discourse parsing | 7321 | 951 |
| Discourse segmentation | 7673 | 991 |

### B. Models and Settings

In our experiments, we select T5 [44] as the pretrained model. The family of T5 models belongs to the encoder-decoder models that were pretrained on various tasks converted into the text-to-text format, which caters to our method. Furthermore, T5 utilizes relative positional embedding and attention mechanisms, which can handle longer sequences than those during training. Therefore, it is feasible for T5 to deal with long documents in document-level parsing, contrary to pretrained models using absolute positional embedding like BART [52] that do not allow direct extrapolation to longer sequences. We have also attempted the byte-level ByT5 [53] and other generative pretrained models on sentence-level parsing, but they are less effective.

In the training process, we set the batch size to 16, and the maximum input and output sequence length to that of the longest sequence for both sentence-level parsing and discourse segmentation. The training epoch is set to 50 and the warmup rate to 0.1. The Adamw [54] optimizer is used with an initial learning rate of 3e-4 together with the cosine learning rate decay scheduler. The same hyperparameters are used in the document-level parsing, except that the batch size and epoch are modified to 2 and 100, respectively.

During inference, we employ beam search with beam sizes of 24 and 6 in the sentence-level and document-level tasks, respectively, as well as our constrained decoding methods. To achieve stable decoding performance, we average the model parameters over the last five epochs. All the experiments are repeated at least three times with different random seeds, and the average results are reported.

### C. Evaluation Metric

To evaluate the performance of our method in sentence-level RST parsing, we follow the common RST-Parseval metric [40], containing micro-averaged F1-scores of unlabeled (Span) and labeled (Nuclearity and Relation). For fair comparison, we use 18 rhetorical relation labels defined in [8], same as other sentence-level RST parsing studies [11], [12], [14]. As for document-level discourse parsing, we adopt the standard Parseval metric [55] which is considered more reasonable and reliable for the task. An additional label (Full) that indicates the merged label of Nuclearity and Relation is reported. The standard Parseval metric cannot be applied to sentence-level parsing due to sentence boundaries, as mentioned in [29].

TABLE II
RESULTS FOR SENTENCE-LEVEL RST PARSING WITHOUT USING GOLD EDU SEGMENTATION

| Approach | Span | Nuclearity | Relation |
|---|---|---|---|
| Soricut and Marcu [23] | 76.70 | 70.20 | 58.00 |
| Joty *et al.* [51] | 82.40 | 76.60 | 67.50 |
| Lin *et al.* [11] (Pipeline) | 91.14 | 85.80 | 76.94 |
| Lin *et al.* [11] (Joint) | 91.75 | 86.38 | 77.52 |
| Nguyen *et al.* [14] (BERT-large) | 92.02 | 87.05 | 78.82 |
| **Our Method** | | | |
| End-to-end parser (T5-small) | 92.83 | 87.38 | 78.58 |
| End-to-end parser (T5-base) | 93.27 | 88.37 | 80.55 |
| End-to-end parser (T5-large) | **93.54** | **88.97** | **80.90** |

TABLE III
RESULTS FOR OUR PARSING MODELS TRAINED WITHOUT NUCLEARITY OR RELATION LABELS

| Approach | Span | Nuclearity | Relation |
|---|---|---|---|
| End-to-end parser (T5-base) | 93.27 | 88.37 | 80.55 |
| - Nuclearity | 93.10 | / | 80.08 |
| - Relation | 92.99 | 87.96 | / |
| - Nuclearity & Relation | 93.02 | / | / |

TABLE IV
RESULTS FOR DISCOURSE SEGMENTATION

| Approach | Precision | Recall | F1-score |
|---|---|---|---|
| **Human Agreement** | 98.50 | 98.20 | 98.30 |
| Soricut and Marcu [23] | 83.80 | 86.80 | 85.20 |
| Joty *et al.* [51] | 88.00 | 92.30 | 90.10 |
| Li *et al.* [18] | 91.08 | 91.03 | 91.05 |
| Wang *et al.* [19] | 92.04 | 94.41 | 93.21 |
| Lin *et al.* [11] (ELMo-large) | 94.12 | 96.63 | 95.35 |
| Lin *et al.* [11] (Joint) | 93.34 | **97.88** | 95.55 |
| Gessler *et al.* [20] (ELECTRA-large) | **96.80** | 95.92 | 96.35 |
| **Our Method** | | | |
| Trained segmenting model (T5-base) | 95.49 | 97.14 | 96.31 |
| + with full training set | 95.64 | 97.41 | **96.52** |
| Extraction from parsing (T5-base) | 95.58 | 97.00 | 96.29 |

In the task of discourse segmentation, we evaluate the performance only with respect to the intra-sentential segment boundaries and report the results of precision, recall, and micro-averaged F1-score to keep the same with [19].

## V. RESULTS AND EVALUATION

We present the experimental results of our end-to-end method for both RST parsing and discourse segmentation tasks. Further analysis and the construction of our augmented data are demonstrated as well.

### A. Sentence-Level RST Parsing

Since our end-to-end method unifies the traditional two stages of RST parsing, we compare our results with the existing models that also do not make use of gold EDU segmentation [11], [14], [23], [51]. These methods, except [14], utilized extra trained automatic segmenting models to generate imprecise EDU segmentations and then send them to their parsing models to build the parsing tree. Besides the pattern of the two-stage pipeline, [11] proposed jointly training the segmenting and parsing models, and [14] treated segmentation as one additional step in the top-down parsing process. And it needs to be mentioned that we do not report the results of [56] since their statistics of the training set are different from ours and those of previous work.

We experiment with T5 models with different scales, and the results are shown in Table II. The performance of our end-to-end method with T5-base is substantially better than the previous best model, with the improvement of approximately 1.2, 1.3 and 1.7 absolute points in Span, Nuclearity and Relation, respectively. The pretrained language model T5-base has a smaller parameter scale than the BERT-large model used in [14]. The advancement in Nuclearity and Relation illustrates that the integration of labels and input text can be learned more effectively through our reformulation, compared with the traditional form of classification tasks with separate frameworks. Moreover, even with the smallest T5-small model, our end-to-end parser can achieve comparable results to [14], with only a slight lag in Relation. And the stronger T5-large model can further improve the performance, demonstrating the effectiveness of our method.

For further exploration, we conduct an ablation study to separate the label and tree structure predictions, with the results displayed in Table III. It is obvious that without label information, the performance of the model decreases in all aspects. Contrary to the intuition that combining the discourse structure and label prediction will make it more challenging for the model, the information from the discourse structure and label is complementary and able to improve the performance through our end-to-end method.

### B. Discourse Segmentation

Benefiting from our end-to-end method, the EDU segmentation is predicted simultaneously during parsing and can be extracted from the generated parsing tree as by-product results. We evaluate the performance of our segmentation prediction in two settings: extraction from parsing and prediction from the specifically trained segmenting model. Our results are shown in Table IV together with those of previous studies. As described in Section IV-A, our segmenting model with full training set keeps the same training data as previous work, but actually uses more training data than the parsing model. So for better comparison, we additionally train another segmenting model with the same training set as the extraction setting, which only contains 7321 sentences. The corresponding results are shown in the first line of our method in Table IV.

In general, the segmenting model trained on the entire training set performs best with the highest F1-score, which can be attributed to its specific training on the segmentation task. Despite a lower Precision score compared with the optimal result in [20], our segmenting model achieves a higher F1-score with T5-base that has fewer parameters than ELECTRA-large [57]. Moreover, the extraction from parsing yields similar performance to the segmenting model trained on the same dataset. It indicates our end-to-end method improves efficiency and averts performance

TABLE V
RESULTS FOR DOCUMENT-LEVEL RST PARSING WITHOUT USING GOLD EDU SEGMENTATION

| Approach | Span | Nuclearity | Relation | Full |
|---|---|---|---|---|
| Zhang *et al.* [13] | 62.3 | 50.1 | 40.7 | 39.6 |
| Nguyen *et al.* [34] (GloVe) | 63.8 | 53.0 | 43.1 | 42.1 |
| Nguyen *et al.* [34] (XLNet-base) | 68.4 | 59.1 | 47.8 | 46.6 |
| **Our Method** | | | | |
| End-to-end parser (T5-small) | 68.5 | 58.3 | 49.3 | 48.0 |
| - different loss weights | 67.7 | 57.7 | 48.9 | 47.5 |
| - discourse tagging rules | 68.3 | 57.8 | 48.3 | 47.2 |
| End-to-end parser (T5-base) | **71.3** | **61.7** | **52.2** | **50.9** |

TABLE VI
RESULTS FOR SENTENCE-LEVEL RST PARSING WITH GOLD EDU SEGMENTATION OR PIPELINE SETTING

| Approach | Span | Nuclearity | Relation |
|---|---|---|---|
| **Human Agreement** | 95.70 | 90.40 | 83.00 |
| Soricut and Marcu [23] | 93.50 | 85.80 | 67.60 |
| Joty *et al.* [51] | 94.60 | 86.90 | 77.10 |
| Ji and Eisenstein [9] | 93.50 | 81.30 | 70.50 |
| Wang *et al.* [16] | 95.60 | 87.80 | 77.60 |
| Lin *et al.* [11] (ELMo-large) | 96.94 | 90.89 | 81.28 |
| Lin *et al.* [11] (Joint) | 97.44 | 91.34 | 81.70 |
| Nguyen *et al.* [14] (BERT-large) | 97.37 | 91.95 | 82.10 |
| **Our Method** | | | |
| Stage 2 parser (w/ gold EDU) | **97.83** | **92.08** | **82.67** |
| End-to-end parser (w/o gold EDU) | 93.27 | 88.37 | 80.55 |
| Pipeline parser (w/o gold EDU) | 93.11 | 87.37 | 77.70 |

hits in the discourse segmentation subtask, even with more complicated training objectives.

### C. Document-Level RST Parsing

We focus on the setting without using gold edu segmentation, which is in accord with our motivation and end-to-end method, and the relevant experimental results are shown in Table V. The strategy of using different loss weights improves the overall performance of our model, while the discourse tagging rules added mainly benefit the prediction of labels, as expected. Compared with the best performance among previous work, which comes from [34], our end-to-end parser with T5-small achieves better scores in the discourse structure (Span), relation label (Relation) and overall label (Full). However, the XLNet-base model used by [34] contains much more parameters than the T5-small model in our experiments, showing the superiority of our method. Furthermore, we also experiment with the larger T5-base model, and the parser achieves significant improvements of about 3 absolute points on all the metric dimensions.

### D. Pipeline Comparison

Although our method was proposed for end-to-end RST parsing, it can be directly applied to two subtasks in the traditional two-stage process of discourse parsing due to its strong transferability. We have already demonstrated the situation of the first subtask of discourse segmentation in Sections III-G. and V-B. For the second subtask, which involves constructing RST parsing trees from the existing gold EDU segmentation, we similarly reformulate it to a seq2seq task and train the corresponding stage 2 parser. Specifically, the current target output sequence is identical to that of end-to-end parsing, while the input sequence is transformed into the linearized output sequence of the discourse segmentation task. Furthermore, we can concatenate the models for the two subtasks to realize the traditional pipeline setting. In this case, our segmenting model trained for discourse segmentation first predicts the EDU segmentations, and then the stage 2 parser utilizes them to generate the final RST parsing trees.

Experimental results based on T5-base for sentence-level RST parsing are presented in Table VI, along with previous studies that use gold EDU segmentations. With the guidance of gold EDU segmentation, our stage 2 parser shows significant improvement over the end-to-end parser, and also outperforms previous methods. The best improvement is observed in the metric of Span, which may be attributed to the exemption from predicting EDU segmentation, consequently alleviating the complexity of the RST parsing task.

On the other hand, although both our stage 2 parser and segmenting model trained on the discourse segmentation task perform well individually, the pipeline parser consisting of them exhibits inferior performance compared to our end-to-end model. We suppose the significant drops in label prediction, namely Nuclearity and Relation, may be due to the error propagation inherent in the pipeline setting, as mentioned earlier. And it is further supported by the comparison between pipeline and joint models from [11] in Table II. These experimental results demonstrate the superiority of our end-to-end approach in terms of both efficiency and performance.

### E. Data Augmentation

The lack of annotated RST parsing trees has been hindering research on discourse parsing since annotators must be experts in discourse analysis and the manual designed for the annotation is quite complicated. From this point, we intend to expand the training set with the augmented data, which is generated and filtered according to our designed rules. We only consider the sentence-level situation because the performance of current models on document-level parsing is far worse and unsatisfactory than that on sentence-level parsing.

Considering that the RST-DT consists of only a small part of the documents in the WSJ corpus and the rest remain without annotation, we can use them to create silver data, which keeps the same domain as the RST-DT. First, the documents in the WSJ corpus that are not selected for annotation in RST-DT are extracted and split into sentences similarly. We choose three parsers trained by our end-to-end method with different random seeds and utilize them to generate candidate output sequences for each sentence we have selected. In this way, we can get the initial and promiscuous instances for parsing, each with an input sentence and three plausible output sequences.

To obtain high-quality data, we check these sequences according to the format we design in the reformulation. And the rule of annotation for RST parsing is also taken into consideration.

TABLE VII
THE STATISTICS OF ORIGINAL TRAINING SET AND OUR AUGMENTED DATASET FOR THE DISCOURSE PARSING TASK

| Dataset | #Sentence | #Avg EDU | #Avg word |
|---|---|---|---|
| Training set | 7321 | 2.48 | 20.31 |
| Initial silver data | 41387 | 2.79 | 26.77 |
| + filtering rules | 36266 | 2.47 | 24.55 |

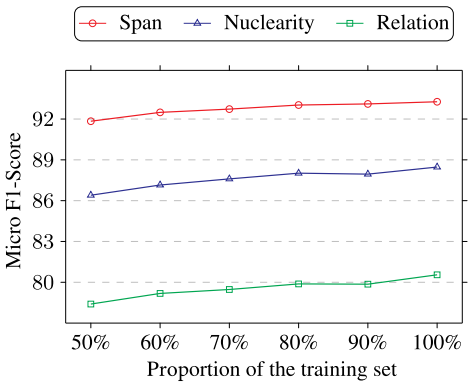



Fig. 5. Performance variation curves with different proportions of the training set.

Considering our constrained decoding methods, we only need to discard the sequences that have mismatched numbers of open brackets. For the rest of the sequences, we employ Algorithm 1 on each of them to restore the constituent information and check whether the nuclearity and relation labels follow the rule of annotation. When nucleus and satellite labels appear together, they should be assigned the label span and another relation label, respectively. And two nucleus labels should use the same relation labels other than the label span.

Through the strategies above, we get those well-formed sequences that follow the labeling rules and have no format errors. If an input sentence still pairs with more than one candidate output sequence, we decide the target sequence via majority voting. The details of our augmented dataset with filtering rules are shown in Table VII. It can be found that the average numbers of EDUs and words in the augmented dataset after filtering approach those of the original training set, which helps to reduce the difference in data distributions between these two datasets.

Before the experiment with augmented data, we explored the influence of the scale of training data. The performance curve in Fig. 5 indicates that more instances indeed continuously promote the performance of the model, and current training data are insufficient. Then we combine the high-quality silver data with the original training set to train our end-to-end parsing model similarly, with T5-base. The results with augmented data can also be found in Table VIII, which show further enhancement in all aspects, particularly in Relation. Considering that the metric of Relation contains 18 labels, the original training set is not large enough for the model to learn sufficient information, so the data augmentation can bring more significant gains. Moreover, we can also extract the discourse segmentation from the predicted

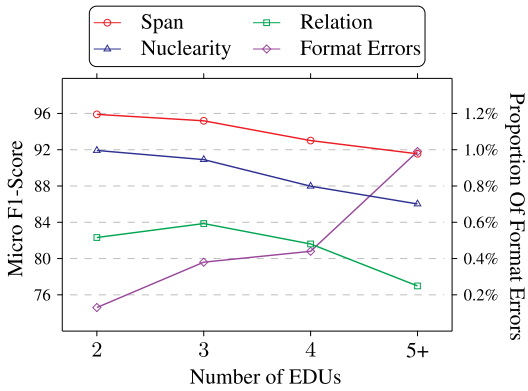



Fig. 6. Performances on Span, Nuclearity and Relation, together with the proportion of instances containing format errors with different numbers of EDUs in instances.

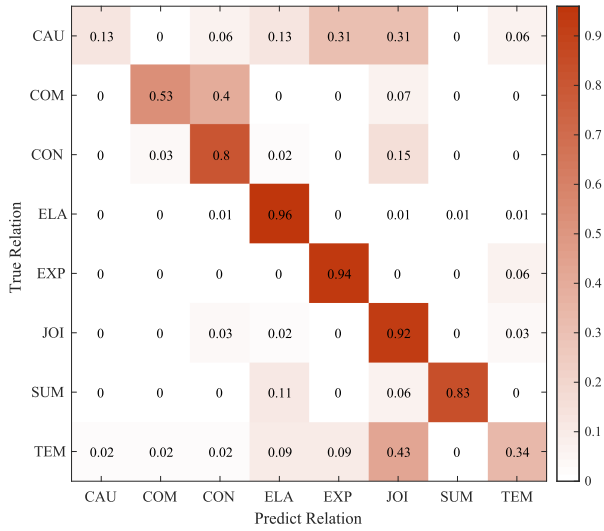



Fig. 7. Confusion matrix for eight semantically similar rhetorical relation labels: Cause(CAU), Comparison(COM), Contrast(CON), Elaboration(ELA), Explanation(EXP), Joint(JOI), Summary(SUM), and Temporal(TEM).

TABLE VIII
RESULTS FOR SENTENCE-LEVEL RST PARSING AND DISCOURSE SEGMENTATION WITH DATA AUGMENTATION

| Sentence-level RST Parsing | Span | Nuclearity | Relation |
|---|---|---|---|
| End-to-end parser | 93.27 | 88.37 | 80.55 |
| + data augmentation | 93.51 | 88.90 | 81.28 |
| **Discourse Segmentation** | **Precision** | **Recall** | **F1-score** |
| Extraction from parsing | 95.58 | 97.00 | 96.29 |
| + data augmentation | 95.86 | 97.11 | 96.48 |

parsing results. Although augmented data similarly improves the performance, it is less effective than that in the parsing task.

### F. Error Analysis and Case Study

In this subsection, we focus on the analysis on sentence-level discourse parsing. In Fig. 6, we show the respective performances of instances with different numbers of EDUs. The

**Input Sentence**

The natural resources development concern said proceeds will be used to repay long-term debt, which stood at 598 million Canadian dollars (US$510.6 million) at the end of 1988.

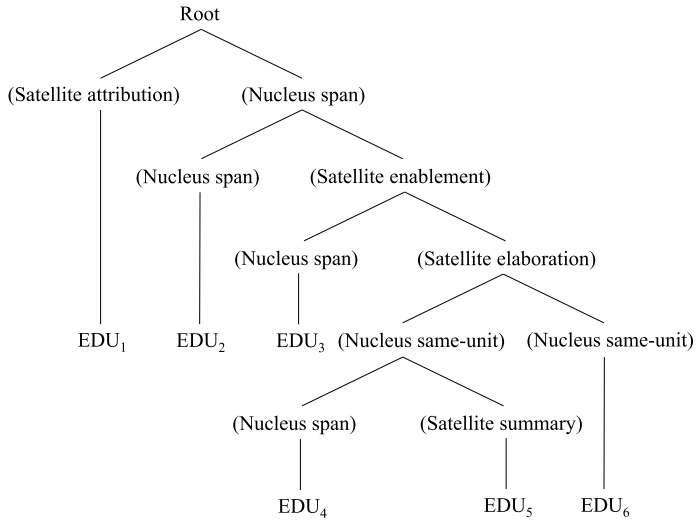
**Output Sequence**

[ [ The natural resources development concern said ] Satellite attribution [ [ proceeds will be used ] Nucleus span [ [ to repay long-term debt, ] Nucleus span [ [ [ which stood at 598 million Canadian dollars ] Nucleus span [ (US$510.6 million) ] Satellite summary ] Nucleus same-unit [ at the end of 1988. ] Nucleus same-unit ] Satellite elaboration ] Satellite enablement ] Nucleus span ]

**Restored Constituents**

(which stood at 598 million Canadian dollars Nucleus span (US$510.6 million) Satellite summary)
(which stood at 598 million Canadian dollars (US$510.6 million) Nucleus same-unit at the end of 1988. Nucleus same-unit)
(to repay long-term debt, Nucleus span which stood at 598 million Canadian dollars (US$510.6 million) at the end of 1988. Satellite elaboration)
(proceeds will be used Nucleus span to repay long-term debt, which stood at 598 million Canadian dollars (US$510.6 million) at the end of 1988. Satellite enablement)
(The natural resources development concern said Satellite attribution proceeds will be used to repay long-term debt, which stood at 598 million Canadian dollars (US$510.6 million) at the end of 1988. Nucleus span)

**Constructed Constituent Tree**




**EDU Segmentation**

**EDU$_1$**: The natural resources development concern said
**EDU$_2$**: proceeds will be used
**EDU$_3$**: to repay long-term debt,
**EDU$_4$**: which stood at 598 million Canadian dollars
**EDU$_5$**: (US$510.6 million)
**EDU$_6$**: at the end of 1988.

**Mistaken Label**


Fig. 8. Example of the output sequence and postprocessing using our method. The red part shows our model correctly predicts the relation label Summary while the other parser mistakenly predicts the label Elaboration. The blue part represents the labels for the text spans before them in the restoration of the constituent tree. The figure of Mistaken Label comes from [34], used for comparison here.

micro F1-scores of Span and Nuclearity drop as the number of EDUs increases, while Relation achieves a low score when the instance only includes two EDUs. We suppose that the increasing difficulty of parsing longer sentences reduces the performance of our method since it remains a challenging problem for the language model to understand long sequences. In addition, short sentences may not contain sufficient information for the model to infer the Relation label, considering that there are 18 relation labels to be identified, while the nuclearity labels only contain two.

The proportion of instances with format errors is actually low, proving the great efficiency of our seq2seq training and tailored constrained decoding, as reported in Fig. 6. And the gradual growth of format errors as the number of EDUs increases shows the difficulty for the model in generating long sequences precisely in keeping with our linearization formats. It can also be proven by the decreasing average EDUs of silver data after the filtering rules. We have attempted to apply the modification of our method for document-level parsing to sentence-level parsing, but the improvement is not obvious. We suppose it is because the difference in text length is not significant in sentence-level parsing. Therefore, it is still challenging but important for future research to explore how to better deal with long sequences in RST parsing since they are the main performance bottleneck.

We also show the confusion matrix for eight semantically similar rhetorical relation labels in Fig. 7, some of which are also mentioned in other studies. Our method fails to effectively distinguish between Temporal and Joint, Comparison and Contrast, but succeeds in Explanation and Elaboration. Fig. 8 shows an instance mistakenly labeled Summary as Elaboration by the other parser [34], but is successfully predicted by our method. We also demonstrate the corresponding output sequence from our method, together with the process of postprocessing, the restored constituent tree, and the extracted EDU segmentation.

### G. Paradigm Discussion

The primary novelty of our work is the paradigm shift of the RST parsing task from traditional top-down or bottom-up stepwise prediction to our text-to-text generation methods. The motivation comes from the text-to-text pretraining mode employed by many pretrained language models, such as T5. They transfer different NLP tasks into the unified text-to-text form so as to train a general model for various tasks through transfer learning, which has achieved great success. In addition, the idea is also similar to the core method of widely used prompt learning [58], where downstream tasks are described in natural language to take better advantage of pretrained language models.

Therefore, we attempt to introduce this paradigm into RST parsing, which is more complex and contains long data structures. We believe the latent knowledge within language models pretrained in similar text-to-text form can be better utilized and improve the training of RST parsing. In our reformulation, the target output sequence contains the complete content of the input text, which can provide more relevant information and enhance the understanding of RST structures for the model. Paolini et al. [43] has also proved that the natural output format can promote the performance of the seq2seq model and encouraged the use of the entire input as context.

Moreover, although our method involves postprocessing for extracting RST trees and other processes, they are inherent components of the new text-to-text paradigm. And we actually avoid the corresponding processing steps in traditional approaches. For example, the previous SOTA system [14] in top-down mode required the labeled span and splitting representations for recursive parsing, together with a specific inference algorithm to generate RST trees. Furthermore, our efficient end-to-end parsing model also eliminates the need for complicated and multi-step frameworks in previous work. The experimental results demonstrate that our parsing model indeed achieves superior performance to traditional approaches, and the new paradigm is promising for RST parsing.

## VI. Conclusion

In this article, we propose a simple but effective end-to-end method for RST parsing to generate the parsing tree directly from the input text. We convert RST parsing into text-to-text generation by reformulating each parsing tree into an equivalent linear sequence. Benefiting from the latent knowledge in pretrained language models, our method does not require additional features or complicated frameworks and can simultaneously perform discourse segmentation during parsing. Moreover, due to its strong transferability, our method can be easily applied to document-level RST parsing and two subtasks in the traditional two-stage process of parsing. Experimental results show that our method exhibits superior performance on these tasks. Furthermore, we create high-quality augmented data to alleviate the lack of annotated RST parsing trees and further improve the performance of our model. In future research, we will explore how to better deal with the long sequences and the lack of annotation data in discourse parsing.

## Acknowledgment

The authors appreciate the anonymous reviewers for their helpful comments.

## References

[1] L. Polanyi and M. van den Berg, "Discourse structure and sentiment," in *Proc. IEEE 11th Int. Conf. Data Mining Workshops*, 2011, pp. 97–102.
[2] P. Bhatia, Y. Ji, and J. Eisenstein, "Better document-level sentiment analysis from RST discourse parsing," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 2212–2218.
[3] A. Louis, A. K. Joshi, and A. Nenkova, "Discourse indicators for content selection in summarization," in *Proc. 11th Annu. Meeting Special Int. Group Discourse Dialogue*, 2010, pp. 147–156.
[4] S. Gerani, Y. Mehdad, G. Carenini, R. T. Ng, and B. Nejat, "Abstractive summarization of product reviews using discourse structure," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1602–1613.
[5] P. Jansen, M. Surdeanu, and P. Clark, "Discourse complements lexical semantics for non-factoid answer reranking," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 977–986.
[6] S. R. Joty, F. Guzmán, L. Màrquez, and P. Nakov, "Discourse structure in machine translation evaluation," *Comput. Linguistics*, vol. 43, no. 4, pp. 683–722, 2017.
[7] W. C. Mann and S. A. Thompson, *Rhetorical Structure Theory: A Theory of Text Organization*. Los Angeles, CA, USA: Inf. Sci. Inst., Univ. Southern California, 1987.
[8] L. Carlson and D. Marcu, "Discourse tagging reference manual," Inf. Sci. Inst., Univ. Southern California, Los Angeles, CA, USA, ISI Tech. Rep. ISI-TR-545, 2001.
[9] Y. Ji and J. Eisenstein, "Representation learning for text-level discourse parsing," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 13–24.
[10] N. Yu, M. Zhang, and G. Fu, "Transition-based neural RST parsing with implicit syntax features," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 559–570.
[11] X. Lin, S. R. Joty, P. Jwalapuram, and S. Bari, "A unified linear-time framework for sentence-level discourse parsing," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 4190–4200.
[12] L. Liu, X. Lin, S. R. Joty, S. Han, and L. Bing, "Hierarchical pointer net parsing," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 1007–1017.
[13] L. Zhang, Y. Xing, F. Kong, P. Li, and G. Zhou, "A top-down neural architecture towards text-level parsing of discourse rhetorical structure," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6386–6395.
[14] T. Nguyen, X. Nguyen, S. R. Joty, and X. Li, "A conditional splitting framework for efficient constituency parsing," in *Proc. Annu. Meeting Assoc. Comput. Linguistics Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 5795–5807.
[15] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2692–2700.
[16] Y. Wang, S. Li, and H. Wang, "A two-stage parsing method for text-level discourse analysis," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 184–188.
[17] N. Kobayashi, T. Hirao, H. Kamigaito, M. Okumura, and M. Nagata, "Top-down RST parsing utilizing granularity levels in documents," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 8099–8106.
[18] J. Li, A. Sun, and S. R. Joty, "SEGBOT: A generic neural text segmentation model with pointer network," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 4166–4172.
[19] Y. Wang, S. Li, and J. Yang, "Toward fast and accurate neural discourse segmentation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 962–967.
[20] L. Gessler, S. Behzad, Y. J. Liu, S. Peng, Y. Zhu, and A. Zeldes, "DisCoDisCo at the DISRPT2021 shared task: A system for discourse segmentation, classification, and connective detection," in *Proc. 2nd Shared Task Discourse Relation Parsing Treebanking (DISRPT 2021)*, Association for Computational Linguistics, Nov. 2021, pp. 51–62.
[21] W. Song and L. Liu, "Representation learning in discourse parsing: A survey," *Sci. China Technol. Sci.*, vol. 63, no. 10, pp. 1921–1946, 2020.
[22] J. Li, M. Liu, B. Qin, and T. Liu, "A survey of discourse parsing," *Front. Comput. Sci.*, vol. 16, no. 5, 2022, Art. no. 165329.

[23] R. Soricut and D. Marcu, "Sentence level discourse parsing using syntactic and lexical information," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol.*, 2003, pp. 228–235.
[24] H. Hernault, H. Prendinger, D. A. duVerle, and M. Ishizuka, "HILDA: A discourse parser using support vector machine classification," *Dialogue Discourse*, vol. 1, no. 3, pp. 1–33, 2010.
[25] S. R. Joty, G. Carenini, R. T. Ng, and Y. Mehdad, "Combining intra- and multi-sentential rhetorical parsing for document-level discourse analysis," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2013, pp. 486–496.
[26] V. W. Feng and G. Hirst, "A linear-time bottom-up discourse parser with constraints and post-editing," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 511–521.
[27] J. Li, R. Li, and E. H. Hovy, "Recursive deep models for discourse parsing," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 2061–2069.
[28] Y. J. Liu and A. Zeldes, "Why can't discourse parsing generalize? A thorough investigation of the impact of data diversity," in *Proc. Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2023, pp. 3104–3122.
[29] N. Kobayashi, T. Hirao, H. Kamigaito, M. Okumura, and M. Nagata, "A simple and strong baseline for end-to-end neural RST-style discourse parsing," in *Proc. Conf. Empirical Methods Natural Lang. Process.*2022, pp. 6725–6737.
[30] N. Yu, M. Zhang, G. Fu, and M. Zhang, "RST discourse parsing with second-stage EDU-level pre-training," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 4269–4280.
[31] J. Fiacco, S. Jiang, D. Adamson, and C. Rosé, "Toward automatic discourse parsing of student writing motivated by neural interpretation," in *Proc. 17th Workshop Innov. Use NLP Building Educ. Appl.*, 2022, pp. 204–215. [Online]. Available: https://aclanthology.org/2022.bea-1.25
[32] F. Koto, J. H. Lau, and T. Baldwin, "Top-down discourse parsing via sequence labelling," in *Proc. Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2021, pp. 715–726.
[33] Z. Liu, K. Shi, and N. Chen, "DMRST: A joint framework for document-level multilingual RST discourse segmentation and parsing," in *Proc. 2nd Workshop Comput. Approaches Discourse*, 2021, pp. 154–164. [Online]. Available: https://aclanthology.org/2021.codi-main.15
[34] T. Nguyen, X. Nguyen, S. R. Joty, and X. Li, "RST parsing from scratch," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2021, pp. 1613–1625.
[35] A. Shen, F. Koto, J. H. Lau, and T. Baldwin, "Easy-first bottom-up discourse parsing via sequence labelling," in *Proc. 3rd Workshop Comput. Approaches Discourse*, 2022, pp. 35–41. [Online]. Available: https://aclanthology.org/2022.codi-1.5
[36] L. Zhang, F. Kong, and G. Zhou, "Adversarial learning for discourse rhetorical structure parsing," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 3946–3957.
[37] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Comput. Surv.*, vol. 55, pp. 1–35, 2023.
[38] L. Liu, M. Zhu, and S. Shi, "Improving sequence-to-sequence constituency parsing," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 4873–4880.
[39] D. Fernández-González and C. Gómez-Rodríguez, "Enriched in-order linearization for faster sequence-to-sequence constituent parsing," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4092–4099.
[40] D. Marcu, "The rhetorical parsing of unrestricted texts: A surface-based approach," *Comput. Linguistics*, vol. 26, no. 3, pp. 395–448, 2000.
[41] D. duVerle and H. Prendinger, "A novel discourse parser based on support vector machine classification," in *Proc. Annu. Meeting Assoc. Comput. Linguistics Int. Joint Conf. Natural Lang. Process.*, 2009, pp. 665–673.
[42] C. Braud, B. Plank, and A. Søgaard, "Multi-view and multi-task training of RST discourse parsers," in *Proc. Int. Conf. Comput. Linguistics*, 2016, pp. 1903–1913.
[43] G. Paolini et al., "Structured prediction as translation between augmented natural languages," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–26.
[44] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, 2020, Art. no. 140.
[45] C. Hokamp and Q. Liu, "Lexically constrained decoding for sequence generation using grid beam search," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1535–1546.
[46] M. Post and D. Vilar, "Fast lexically constrained decoding with dynamic beam allocation for neural machine translation," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2018, pp. 1314–1324.
[47] P. Chen, N. Bogoychev, K. Heafield, and F. Kirefu, "Parallel sentence mining by constrained decoding," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 1672–1678.
[48] Z. Sun et al., "Rethinking document-level neural machine translation," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 3537–3548.
[49] L. Carlson, D. Marcu, and M. E. Okurovsky, "Building a discourse-tagged corpus in the framework of rhetorical structure theory," in *Proc. Annu. Meeting Special Int. Group Discourse Dialogue Workshop*, 2001, pp. 1–10.
[50] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a large annotated corpus of English: The Penn Treebank," *Comput. Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.
[51] S. R. Joty, G. Carenini, and R. T. Ng, "A novel discriminative framework for sentence-level discourse analysis," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, 2012, pp. 904–915.
[52] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880.
[53] L. Xue et al., "ByT5: Towards a token-free future with pre-trained byte-to-byte models," *Trans. Assoc. Comput. Linguistics*, vol. 10, pp. 291–306, 2022.
[54] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–19.
[55] M. Morey, P. Muller, and N. Asher, "A dependency perspective on RST discourse parsing and evaluation," *Comput. Linguistics*, vol. 44, no. 2, pp. 197–235, 2018.
[56] Y. Zhang, H. Kamigaito, and M. Okumura, "A language model-based generative classifier for sentence-level discourse parsing," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 2432–2446.
[57] K. Clark, M. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–18.
[58] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Comput. Surv.*, vol. 55, no. 9, 2023, Art. no. 195.

**Xinyu Hu** received the B.S. degree in 2021 from the Department of Computer Science and Technology, Peking University, Beijing, China, where he is currently working toward the Ph.D. degree with the Wangxuan Institute of Computer Technology. His main research interests include discourse parsing, machine translation, and long text modeling.

**Xiaojun Wan** received the B.S. degree in information science from the Department of Information Management, Peking University (PKU), Beijing, China, in 2000, and the M.S. and Ph.D. degrees in computer science from the Department of Computer Science and Technology, PKU in 2003 and 2006 respectively. He is currently a Professor with the Wangxuan Institute of Computer Technology (WICT, formerly known as the Institute of Computer Science and Technology), PKU. He has many publications in major international conferences and journals, including ACL, EMNLP, NAACL, SIGIR, AAAI, IJCAI, WWW, CIKM, TOIS, Computational Linguistics, *Journal of the Association for Information Science and Technology*, *Information Processing and Management*, and *Information Sciences*. His research interests include natural language processing and deep learning, and is recently interested in exploring NLG evaluation, faithfulness and safety of LLMs, and cross-modal generation. He was an Editorial Board Member of the *Journal of Computer Science and Technology* during 2020–2023, *Natural Language Engineering* during 2019–2023, and Computational Linguistics during 2016–2018. He was the PC Chair of EMNLP-IJCNLP 2019, a Senior PC Member of IJCAI (2016/2018–2020/2022–2023) and AAAI (2019-2023), and the Area Chair of ACL, NeurIPS, IJCAI, EMNLP, NAACL, EACL, and IJCNLP.