RST Discourse Parsing with Second-Stage EDU-Level Pre-training

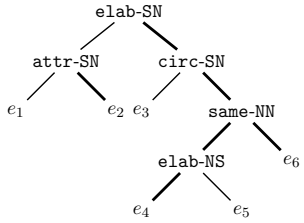
Nan Yu[1] and **Meishan Zhang**[2] and **Guohong Fu**[1,3]* and **Min Zhang**[2]
[1]School of Computer Science and Technology, Soochow University, China
[2]Institute of Computing and Intelligence, Harbin Institute of Technology (Shenzhen), China
[3]Institute of Artificial Intelligence, Soochow University, China
nyu@stu.suda.edu.cn, mason.zms@gmail.com,
{ghfu,minzhang}@suda.edu.cn

Abstract

Pre-trained language models (PLMs) have shown great potentials in natural language processing (NLP) including rhetorical structure theory (RST) discourse parsing. Current PLMs are obtained by sentence-level pre-training, which is different from the basic processing unit, i.e. element discourse unit (EDU). To this end, we propose a second-stage EDU-level pre-training approach in this work, which presents two novel tasks to learn effective EDU representations continually based on well pre-trained language models. Concretely, the two tasks are (1) next EDU prediction (NEP) and (2) discourse marker prediction (DMP). We take a state-of-the-art transition-based neural parser as baseline, and adopt it with a light bi-gram EDU modification to effectively explore the EDU-level pre-trained EDU representation. Experimental results on a benckmark dataset show that our method is highly effective, leading a 2.1-point improvement in F1-score. All codes and pre-trained models will be released publicly to facilitate future studies.[1]

1 Introduction

Discourse analysis based on rhetorical structure theory (RST) has received increasing interest in the natural language processing (NLP) community (Yu et al., 2018; Liu et al., 2019a; Kobayashi et al., 2020; Zhang et al., 2020; Guz and Carenini, 2020; Koto et al., 2021; Zhang et al., 2021), which organizes discourse output through a well-defined tree structure. Figure 1 shows an example of an RST constituent tree, where the leaf nodes are element discourse units (EDUs). Given an EDU sequence, RST discourse parsing aims to automatically construct a hierarchical constituent tree[2].



e_1[CNW Corp. said] e_2[the final step in the acquisition of the company has been completed with the merger of CNW with a subsidiary of Chicago & North Western Holdings Corp.] e_3[As reported,] e_4[CNW agreed to be acquired by a group of investors] e_5[led by Blackstone Capital Partners Limited Partnership] e_6[for $50 a share, or about $950 million.]

Figure 1: An example of RST discourse tree. e_1, e_2, e_3, e_4, e_5, and e_6 are EDUs. `elab`, `attr`, `circ` and `same` are relations. `NS`, `SN`, and `NN` are nuclearities.

The shift-reduce transition-based model has been widely adopted in RST discourse parsing (Yu et al., 2018; Mabona et al., 2019), building the constituent tree incrementally with multiple steps by a sequence of actions. These models take EDU-level features as inputs to score transition actions at each step. Recently, neural network models have achieved state-of-the-art performance for this task by using sophisticated-designed neural modules (Yu et al., 2018; Liu et al., 2019a; Mabona et al., 2019; Zhang et al., 2020; Kobayashi et al., 2020). In particular, the contextualized pre-trained language models (PLMs) such as XLNet (Yang et al., 2020) is able to achieve an impressive performance, resulting in F1-score gains of more than 3 points according to previous studies (Koto et al., 2021; Zhang et al., 2021; Nguyen et al., 2021) and our preliminary findings.

Although great successes have been observed by the contextualized PLMs (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019b), an apparent mismatch in the basic processing units exists between the EDU-level RST parsing and the sentence-level contextualized language modeling, which might be unable to fully explore the pre-training paradigm. Several previous studies have been investigated to address the mismatch between the target tasks

*Corresponding author.
[1]http://github.com/yunan4nlp/E-NNRSTParser
[2]In this study, we focus on the tree construction task, assuming the gold standard EDU as inputs.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4269 - 4280
May 22-27, 2022

and the standard language model pre-training, e.g., SpanBERT (Joshi et al., 2020) for extractive question answering, BART (Lewis et al., 2020), and T5 (Raffel et al., 2020) for sequence-to-sequence (seq2seq) generation, and all these studies achieve improved performances for their target tasks.

In this study, we investigate a second-stage EDU-level pre-training based on the above observation. Concretely, we conduct pre-training from a PLM with two EDU-level tasks in the second stage. The first task is next EDU prediction (NEP), which is inspired by next sentence prediction (NSP) in BERT (Devlin et al., 2019) learning, substituting the sentences with EDUs. The second task is discourse marker prediction (DMP), which is also inspired by the masked language modeling (MLM) in BERT (Devlin et al., 2019) learning, substituting the masked words with the masked discourse markers. To fully utilize contextualized pre-trained representations, we adapt a transition-based neural RST parser that exploits BiEDU representations with regard to the basic encoding unit instead of the standard single-EDU manner.

We conduct experiments on RST discourse treebank (RST-DT) (Carlson et al., 2001) to evaluate the proposed model. First, we derive BiEDU representations directly from PLMs, and thus build a very strong transition-based neural RST parser. Then, we examine the proposed second-stage EDU-level pre-training approach. Experimental results show that the two second-stage pre-training tasks improve RST parsing greatly, and their combination leads to further increases. Our final model achieves the top performance among all the models reported in the literature.

In summary, our contributions are as follows:

- We present a second-stage EDU-level pre-training approach to address the inconsistency between the EDU-level RST parsing and the sentence-level contextualized language modeling, aiming for a better pre-training paradigm for RST parsing.
- We suggest BiEDU-based representations for neural RST parsing to exploit well pre-trained language models more effectively.
- We advance the state-of-the-art RST parsing performance.

2 Second-Stage EDU-Level Pre-training

In this section, we introduce the proposed second-stage EDU-level pre-training approach. It has two EDU-level pre-training tasks, termed by NEP and DMP, respectively. NEP requiries EDU pairs as inputs, and predicts whether each EDU pair is adjacent. DMP requiries EDU sequences as inputs, and predicts the masked discourse marker between two adjacent EDUs.

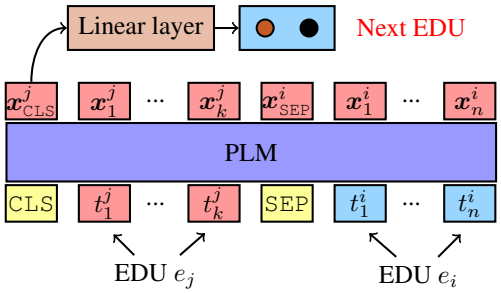


Figure 2: Framework of NEP. The input includes two EDUs, e_j and e_i, which are randomly sampled from the text. They may not be continous.

2.1 Next EDU Prediction (NEP)

NEP is inspired by NSP in BERT (Devlin et al., 2019) learning. NSP is a binary sentence-level classification task, which determines whether two sentences are continous. It integrates rich inter-sentence context features into BERT and thus has a positive effect on several downstream classification tasks, such as PDTB-style discourse relation classification (Shi and Demberg, 2019) and Stanford Natural Language Inference (SNLI) (Bowman et al., 2015). RST parsing involves the classification between two subtrees (a single EDU can also become a subtree), which is highly similar to above downstream tasks. Therefore, we believe that a similar second-stage pre-training task is effective for RST parsing. Considering that the basic inputs of RST parsing are EDUs, we substituting the sentences with EDUs.

We reimplement a SOTA EDU segmenter[3] (Muller et al., 2019) and use it to segment large-scale unlabeled texts. Based on EDU segmentation data, we apply NEP to PLM. Figure 2 shows an overview of NEP. We sample the continuous EDU pairs as positive instances, and the non-continuous EDU pairs as negative instances. It should be noted that these positive and negative instances are sampled on the same scale. When the these instances are ready, we use Equation 4 to pack each EDU pair and calculate its corresponding EDU representation. Then we use a

[3]We also use the RST-DT corpus to train an EDU segmenter, which achieves 96.0% F1-score.

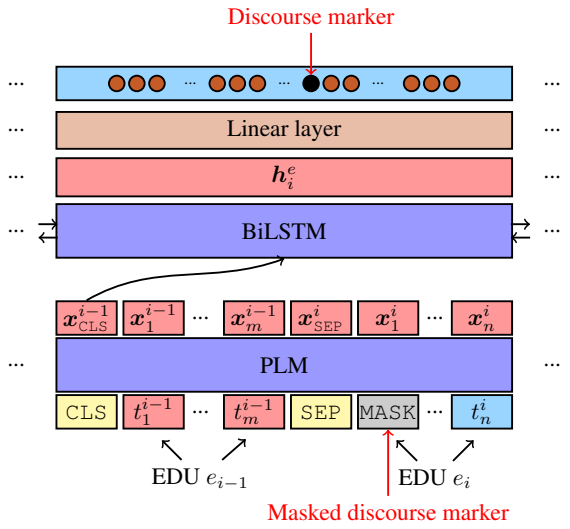


Figure 3: Framework of DMP. The input is an EDU sequence. For convenience, here we only draw two adjacent EDUs e_{i-1} and e_i.

linear layer to calculate the binary score:

$$\boldsymbol{y}^e = \boldsymbol{W}^e \boldsymbol{x}_i^e + \boldsymbol{b}^e \quad (1)$$

where $\boldsymbol{W}^e$ and $\boldsymbol{b}^e$ are the model parameters of the linear layer, and $\boldsymbol{y}^e$ indicates whether the two EDUs are continous. We adopt a cross entropy function as the training objective of NEP.

2.2 Discourse Marker Prediction (DMP)

We further adopt DMP to pre-train PLMs in the second stage based on the following consideration. Pitler et al. (2009) point out that if discourse markers (Schiffrin, 1987) exist in PDTB-style discourse parsing, the classification of discourse relation types become easier. RST parsing aims to classify the relationship between two discourse fragments. By analogy, discourse markers can also make RST parsing easier.

The framework of DMP is shown in Figure 3. The input of DMP is an EDU sequence. We only mask the first word in each EDU that starts with a discourse marker.[4] Then we use Equations 4 and 5 to obtain EDU representations of the masked EDU sequence. Finally, we feed them into a linear layer to calculate the discourse marker score:

$$\boldsymbol{y}^m = \boldsymbol{W}^m \boldsymbol{h}_i^e + \boldsymbol{b}^m \quad (2)$$

where $\boldsymbol{W}^m$ and $\boldsymbol{b}^m$ are the model parameters of the linear layer, and $\boldsymbol{y}^m$ is the score distribution of the discourse markers. We also use a cross entropy function as the training objective of DMP.

[4]We adopt the discourse markers defined by Fraser (2009).

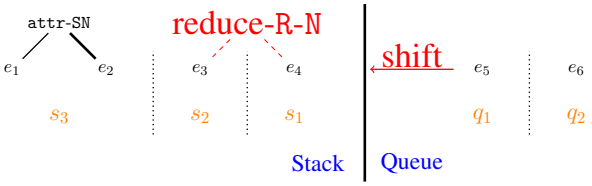


Figure 4: An example to illustrate our transition system. e_1, e_2, e_3, e_4, e_5, and e_6 are EDUs. s_1, s_2, and s_3 are the top three subtrees in the stack. q_1 and q_2 are top two EDUs in the queue. "reduce-R-N" is a candidcate action set, where R and N represent a relation label and a nuclearity label, respectively. "shift" is a gold standard action.

3 Transition-based Neural RST Parser

We adopt a transition-based neural RST parser to evaluate the second-stage EDU-level pre-training approach. The model has two key components, termed by a transition system and a neural network model, respectively. The transtion system, mainly borrowed from Yu et al. (2018), formalizes RST parsing into action sequence predictions, and the neural model yields EDU representations and outputs action sequences.

3.1 Transition System

As shown in Figure 4, our transition system consists of states and actions. A state has two parts, namely a stack stores partially parsed subtrees and a queue stores un-parsed EDUs. The initial state is an empty state, and the final state represents a full RST discourse tree. A action controls the transition of states. There are three kinds of actions:

- A shift action pops the first EDU of the queue and pushes it into the stack. It can only be executed when the queue is not empty.
- A reduce action combines the top two subtrees of the stack into a new subtree with a unclearity label and a relation label. It can only be executed if there are more than two subtrees are in the stack.
- A pop root action pops a full discourse tree from the stack, and the parsing process is completed. It can only be executed when the queue is empty, and only one element is in the stack.

In summary, the transition system converts a tree construction into a sequence of action predictions. By performing the actions, a RST discourse tree is constructed incrementally. Concretely, given the example in Figure 1, we perform actions "shift,

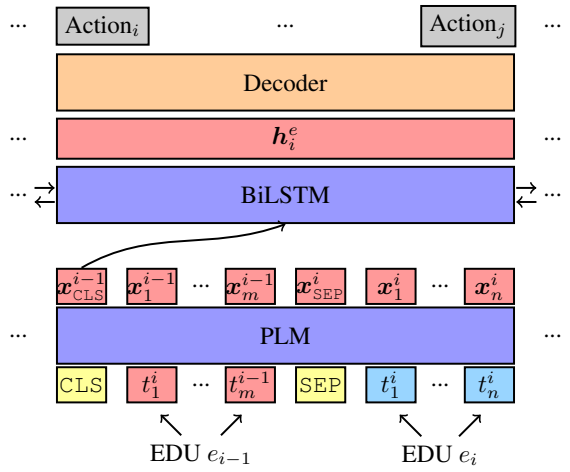


Figure 5: Framework of our neural network model. The input is an EDU sequence. For convenience, here we draw two adjacent EDUs e_{i-1} and e_i.

shift, reduce-attr-SN, shift, shift, shift, reduce-elab-NS, shift, reduce-same-NN, reduce-circ-SN, reduce-elab-SN, pop root" to construct a full RST discourse tree step by step.

3.2 Neural Network Model

The Vanilla Representation We use PLM to encode each text, obtaining single-EDU representations. Concretely, given a text that has been segmented into EDUs $e_1 \cdots e_n$, a special symbol [CLS] is placed at the beginning of each EDU. Then each EDU is tokenized by byte pair encoding (BPE) (Sennrich et al., 2016), and encoded by PLM to obtain contextualized word piece embeddings. Finally, for each EDU, we choose the following representation of [CLS] to represent it:

$$\begin{aligned} e_i &= \texttt{[CLS]}, t_1^i \cdots t_n^i \\ \boldsymbol{x}_{\texttt{CLS}}^i, \boldsymbol{x}_1^i \cdots \boldsymbol{x}_n^i &= \text{PLM}(e_i) \\ \boldsymbol{x}_i^e &= \boldsymbol{x}_{\texttt{CLS}}^i \end{aligned} \tag{3}$$

where [CLS], $t_1 \cdots t_n$ are word pieces, $\boldsymbol{x}_{\texttt{CLS}}^t, \boldsymbol{x}_1^t \cdots \boldsymbol{x}_n^t$ are word piece embeddings, and x_i^e is the single-EDU representation.

Extension with BiEDU The vanilla EDU-based representation exploits the information by treated an EDU as the first segmentation type, leaving its segmentation type unused. Here, we make an extension by using BiEDU representations. Each input unit is packaged by the current EDU as well as the previous EDU jointly, forming as BiEDU. Then [CLS] is placed before the first EDU and [SEP] before the second EDU. We also use BPE to tokenize it and use a PLM for encoding. We still choose the representation of [CLS] to represent each EDU as follow:

$$\begin{aligned} (e_{i-1}, e_i) &= \texttt{[CLS]} \cdots t_m^{i-1}, \texttt{[SEP]} \cdots t_n^i \\ \boldsymbol{x}_{\texttt{CLS}}^{i-1} \cdots \boldsymbol{x}_m^{i-1}, \boldsymbol{x}_{\texttt{SEP}}^i \cdots \boldsymbol{x}_n^i &= \text{PLM}(e_{i-1}, e_i) \\ \boldsymbol{x}_i^e &= \boldsymbol{x}_{\texttt{CLS}}^{i-1} \end{aligned} \tag{4}$$

where [CLS] $\cdots t_m^{i-1}$, [SEP] $\cdots t_n^i$ are tokens, $\boldsymbol{x}_{\texttt{CLS}}^{i-1} \cdots \boldsymbol{x}_m^{i-1}, \boldsymbol{x}_{\texttt{SEP}}^i \cdots \boldsymbol{x}_n^i$ are word piece embeddings, and x_i^e is the BiEDU representation.

BiLSTM Encoding Furthermore, we follow Koto et al. (2021), using BiLSTM to obtain high-level EDU representations:

$$\boldsymbol{h}_1^e \cdots \boldsymbol{h}_u^e = \text{BiLSTM}(\boldsymbol{x}_1^e \cdots \boldsymbol{x}_u^e) \tag{5}$$

where $\boldsymbol{h}_1^e \cdots \boldsymbol{h}_u^e$ are final EDU representations. In addition, we follow Zhang et al. (2021) and Koto et al. (2021), using paragraph features to further enhance the high-level representations.

Decoder The decoder part predicts the next-step action based on a given state. We follow Yu et al. (2018), selecting the three subtrees (s_1, s_2, s_3) at the top of the stack and the first EDU (q_1) in the queue to represent the current state. We calculate the subtree representation by the average of its EDU representations. We concatenate three subtree representations ($\boldsymbol{h}_{s1}$, $\boldsymbol{h}_{s2}$, $\boldsymbol{h}_{s3}$) and an EDU representation ($\boldsymbol{h}_{q1}^e$), and input them into a linear layer to calculate the score distribution of the action:

$$\boldsymbol{y}_i = \boldsymbol{W}_i(\boldsymbol{h}_{s_1} \oplus \boldsymbol{h}_{s_2} \oplus \boldsymbol{h}_{s_3} \oplus \boldsymbol{h}_{q_1}^e) + \boldsymbol{b} \tag{6}$$

where $\boldsymbol{W}_i$, $\boldsymbol{b}$ are model parameters and $\oplus$ is a concatenation operation. During the inference, at each step, we exploit the highest-scored action as the output. When actions are ready, we perform them to construct the coresponding RST discourse tree step by step according to the transition system introduced in Section 3.1.

Training We adopt a cross-entropy loss plus with l_2 regularization term as an objective function to train our RST parser. Given a state, we obtain action scores according to the neural network model and compute the probability of the gold action by softmax. Finally, we feed it into the objective function for loss calculation as follows:

$$\begin{aligned} \boldsymbol{p}_i &= \text{softmax}(\boldsymbol{y}_i) \\ \mathcal{L}(\boldsymbol{\theta}) &= -\log(\boldsymbol{p}_i[a_i^g]) + \frac{\lambda||\boldsymbol{\theta}||_2}{2} \end{aligned} \tag{7}$$

where a_i^g is the gold-standard action of the i-th step, $\boldsymbol{\theta}$ is a set of model parameters of our RST parser, and λ is the l_2 regularization factor. We use Adam algorithm (Kingma and Ba, 2015) to optimize the model parameters of our neural network model.

4 Experiments

4.1 Settings

Datasets To show the proposed model is comparable with previous state-of-the-art systems for RST parsing, we conduct experiments on RST-DT[5] (Carlson et al., 2001). It is a standard benchmark dataset for this task, which is collected from the Wall Street Journal news. It has been divided into training and test sets, which have 347 discourses and 38 discourses, respectively. We randomly select 35 discourses from the training set to develop our model. The original RST-DT contains 78 fine-grained discourse relations. Most of previous studies simplify these fine-grained discourse relations to 18 coarse-grained relations. To facilitate comparison with previous studies, we also use 18 simplified coarse-grained relations.

To show the domain generalization capability of our proposed RST parser to unseen domain articles, we test it on the georgetown university multilayer (GUM) corpus[6]. It contains small-scale articles annotated based on RST in several domains, such as news, fiction, conversations, and etc. For more details, one can refer to their paper (Zeldes, 2017).

The training corpus for second-stage EDU-level pre-training contains unlabeled large-scale collected from a English Wikipedia corpus[7]. Although using a unlabeled news corpus may lead to greater improvements, we find that using a Wikipedia corpus is sufficient to provide new SOTA results.

Evaluation We use the evaluation recommended by Morey et al. (2017), which attaches nuclearity and relation labels to non-leaf trees to eliminate redundant evaluations. The evaluation includes four metrics, termd by Span, Nuclearity, Relation, and Full, respectively. Span evaluates the skeleton of the discourse tree. Nuclearity evaluates the discourse tree with nuclearity labels. Relation evaluates the discourse tree with relation labels. Full evaluates the complete discourse tree with nuclearity and relation labels.

[5] https://catalog.ldc.upenn.edu/LDC2002T07

[6] https://github.com/amir-zeldes/gum

[7] https://dumps.wikimedia.org/enwiki

Hyper-parameters There are several hyper-parameters in our proposed second-stage EDU-level pre-training approach and RST parser. In NEP, the learning rate of PLM is set to 5e-6, and the learning rate of the other model parameters is set to 1e-3. The batch size is set to 50. The maximum norm of gradient clipping is set to 1. The maximum tranining epoch number is set to 10. In DMP, the learning rate of PLM is set to 1e-6, and the learning rate of the other model parameters is set to 1e-4. The batch size is set to 1. The output hidden size of LSTM is set to 200. The settings of maximum training iteration number and the norm of gradient cliping are the same as NEP.

The hyper-parameters of our RST parser are tuned based on the preliminary results on the development set. The hidden size of all neural layers is set to 200. The dropout is set to 0.25. The learning rate of PLM is set to 2e-5, and the learning rate of other model parameters is set to 1e-3. The maximum norm of gradient clipping is set to 1, and the maximum training iteration number is set to 20.

We use *transformers library* (Wolf et al., 2020) to implement PLM and use *PyTorch* (Paszke et al., 2019) to implement other neural network modules.

4.2 Development Results

We conduct several development experiments to show the important factors that influence the performance of our RST parser.

Different Pre-trained Language Models First, we test our proposed RST parser based on several publicly available PLMs such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b), XLNet (Yang et al., 2020), SpanBERT (Joshi et al., 2020), and DeBERTa (He et al., 2020). The max input length of BERT, RoBERTa, SpanBERT, and DeBERTa is 512 tokens. Therefore, we extend them with BiEDU to better exploit these PLMs. Since XLNet has no input length limit, we do not need to apply BiEDU extension to our XLNet RST parser. Table 1 shows the performances of with different PLMs. We find that our BiEDU extension is able to further improve the performances of these PLM-based RST parsers. The SpanBERT RST parser achieves worst performance among these RST parsers. It is probably because that the basic processing units of SpanBERT learning are not matched with RST parsing. The XLNet RST parser achieves the best performance among these RST parser. Therefore, following experiments are

Models	Full (dev)	Full (test)
BERT	49.0	45.2
BERT + BiEDU	51.4	48.9
RoBERTa	50.8	48.0
RoBERTa + BiEDU	51.7	49.5
SpanBERT	41.0	38.8
SpanBERT + BiEDU	42.5	39.2
DeBERTa	48.6	47.0
DeBERTa + BiEDU	49.8	48.1
XLNet	**52.2**	**51.4**

Table 1: Performances of our RST parser with different PLMs.

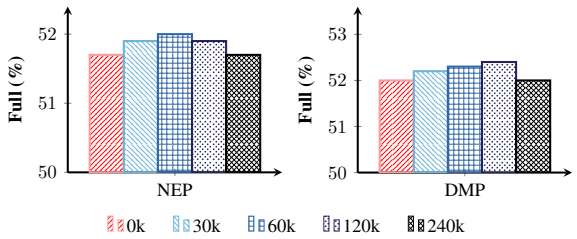


Figure 6: Performances of our RST parser on the development set under second-stage EDU-level pre-training with different sizes of unlabeled articles.

conducted based on the XLNet RST parser.

Unlabeled Article Size We study how the unlabeled articles size in second-stage EDU-level pre-training influences the performance of our RST parser. First, we apply NEP to PLMs. As shown in Figure 6, the performance of our RST parser shows a similar trend when increasing the size of unlabeled articles to perform DMP based pre-training. When the size of the unlabeled articles reaches 30k, the Full metric reaches its peak. Therefore, we use 30k unlabeled articles in NEP.

Then, we adopt DMP to further pre-train the PLM part of our RST parser in the second stage. As can be seen from Figure 6, the performance of our RST parser first increases and then decreases as the size of the unlabeled articles as the size the unlabeled articles gradually increases from 0 to 240k. When the size of the unlabeled articles reaches 120k, the Full metric reaches its peak. Therefore, we use 120k unlabeled articles in DMP. Above experimental results show that we do not need an ultra large-scale unlabeled corpus for our proposed second-stage EDU-level pre-training approach.

4.3 Final Results

As shown in Table 2, we report main results on the RST-DT test set. Our proposed RST parser achieves 73.4 on the Span metric, 63.3 on the Nuclearity metric, 52.4 on the Relation metric, and 51.4 on the Full metric, exceeding most of the previous state-of-the-art systems. When we apply second-stage EDU-level pre-training to XLNet, it achieves 76.4 on the Span metric and 66.1 on the Nucleairty metric, resulting a Full metric improvement 53.5 - 51.4 = 2.1. The Span, nuclearity, and relation metrics have similar tendencies as well. In addition, we implement a top-down RST parser, and also enhance it with using our proposed second-stage EDU-level pre-training approach. We find that the proposed approach is able to improve the performance of top-down RST parser as well.

Models	S	N	R	F
XLNet (transition-based)	73.4	63.3	52.4	51.4
+ NEP + DMP	**76.4**	**66.1**	54.5	53.5
XLNet (top-down)	73.3	62.7	51.9	49.7
+ NEP + DMP	72.9	62.7	52.5	50.5
Feng and Hirst (2014)	68.6	55.9	45.8	44.6
Ji and Eisenstein (2014)	64.1	54.2	46.8	46.3
Joty et al. (2015)	65.1	55.5	45.1	44.3
Surdeanu et al. (2015)	65.3	54.2	45.1	44.2
Li et al. (2016)	64.5	54.0	38.1	36.6
Hayashi et al. (2016)	65.1	54.6	44.7	44.1
Braud et al. (2016)	59.5	47.2	34.7	34.3
Braud et al. (2017)	62.7	54.5	45.5	45.1
Yu et al. (2018)	71.4	60.3	49.2	48.1
Mabona et al. (2019)	67.1	57.4	45.5	45.0
Zhang et al. (2020)	67.2	55.5	45.3	44.3
Nguyen et al. (2021)	74.3	64.3	51.6	50.2
Koto et al. (2021)	73.1	62.3	51.5	50.3
Zhang et al. (2021)	76.3	65.5	**55.6**	**53.8**
Human	78.7	66.8	57.1	55.0

Table 2: Final results of RST parsing on the test set.

We compare our proposed RST parser with previous state-of-the-art systems. Feng and Hirst (2014) propose a linear-chain conditional random field (CRF) parser. Ji and Eisenstein (2014) adopt a statistical transition-based parser with a representation learning. Surdeanu et al. (2015) employ a perceptron and a logistic regression to parse a text. Li et al. (2016) propose a hierarchical neural parser with attention. Joty et al. (2015) propose an intra-sentential and multi-sentential parser. Hayashi et al. (2016) reimplement the HILDA parser (Heilman and Sagae, 2015), using a linear SVM classification to parse a text from the bottom up. Braud et al. (2016) present a BiLSTM RST parser with multi-task learning. Braud et al. (2017) propose a neural greedy parser with cross-lingual recourse. Yu et al. (2018) propose a transition-based neural parser, and further enhance it with hidden-layer vectors extracted from a neural syntax parser. Mabona et al. (2019) propose a generative RST parser with beam search. Zhang et al. (2020) propose a top-

Models	S	N	R	F
Our proposed model	**76.4**	**66.1**	**54.4**	**53.5**
- DMP	74.8	64.3	53.1	52.1
- NEP	75.3	65.0	53.8	52.6
- NEP - DMP	73.4	63.3	52.4	51.4
- NEP - DMP - Para	73.1	62.6	51.9	50.5

Table 3: Ablation study on the test set. "-Para" represents without paragraph features.

down neural parser. Koto et al. (2021) propose a transformer top-down parser with dynamic oracle. Nguyen et al. (2021) propose a seq2seq neural parser based on point network. Koto et al. (2021) propose a sequence labelling parser with dynamic oracle. Zhang et al. (2021) propose a neural top-down parser with adversarial learning. As shown in Table 2, our transition-based XLNet RST parser achieves the best performance among the systems studied on the Span and the Nuclearity metrics. We find that the Relation and the Full metrics of our RST parser are lower than that of Zhang et al. (2021). It is probably because that our proposed second-stage EDU-level pre-training approach only requires predicted EDU segmentation, lacking the information of predicted RST discourse trees.

4.4 Analysis

In this section, we conduct several analysis experiments from different aspects to better understand the proposed RST parser.

Ablation Studies Here we conduct several ablation experiments to examine the effectiveness of our proposed second-stage EDU-level pre-training approach and paragraph features. As shown in Table 3, we find that NEP and DMP are effective for RST discourse parsing. NEP improves our XLNet RST parser by an increase of 52.1 - 51.4 = 0.7 on the Full metric. The tendency of DMP is similar to NEP, obtaining an increase of 52.6 - 51.4 = 0.8 on the Full metric. Our proposed model can be further improved when two EDU-level tasks are applied to XLNet, resulting the Full metric improvement 53.5 - 51.4 = 2.1. In addition, the paragraph features is also effective for RST discourse parsing, which results the overall improvements.

Effect of EDU Segmentation Performance As mentioned earlier, the second-stage EDU-level pre-training approach requires EDU segmentation produced by a supervised EDU segmenter. Predicted EDU segmentation could have errors, which may propagate into RST parsing. Here we examine how

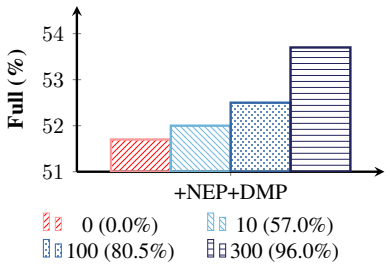


Figure 7: Effect of EDU segmenter performances on our proposed RST parser.

EDUs	Models	S	N	R	F
[1,5]	XLNet	94.3	82.6	70.1	69.5
	+NEP	94.5	83.2	70.9	70.2
[6,10]	XLNet	54.4	37.2	21.8	21.8
	+NEP	57.2	39.0	23.9	23.6
[11,15+]	XLNet	34.6	25.2	19.2	19.2
	+NEP	36.5	26.8	20.7	20.7

Table 4: Performances on the test set with different number of EDUs in spans.

the performance of the supervised EDU segmenter influence the performance of RST parsing. The full EDU segmenter is trained on 300 discourses. We retrain two weaker EDU segmenters on 10 and 100 discourses. Figure 7 shows the RST parsing performances with different EDU segmenters. We find that the EDU segmentation performance influences the RST parsing quality, indicating the importance of correct EDU segmentation.

Analysis by Number of EDUs in Subtrees As mentioned earlier, NEP predicts whether each EDU pair is continous, and it is able to integrate rich inter-EDU context features into PLMs. Therefore, it is expected that the introduce of NEP may bring better improvements for the spans containing more EDUs. As such, here we investigate the benefit by using NEP. Table 4 shows the comparison results. We find that performances are improved significantly when spans contains more EDUs.

Effect of Different Sampling Strategies Forthermore, we examine how different EDU pair sampling strategies influence RST discourse parsing. The training set of NEP is sampled from a large-scale unlabeled corpus. We sample the continuous EDU pairs as the positive instances and the non-continuous EDU pairs as the negative instances. The difficulty of NEP changes depending on how the non-continuous EDU pairs are sampled. Here we compare four strategies of sampling the non-continuous EDU pairs: from a sentence, two adjacent sentences, two sentences in an article, and two

Sampling Strategies	S	N	R	F
From a sentence	74.4	63.9	53.4	52.3
From adjacent sentences	74.9	64.6	53.4	52.2
From a article	74.8	64.3	53.1	52.1
From two articles	73.9	64.2	52.6	51.9
XLNet	73.4	63.3	52.4	51.4

Table 5: Influence of different sampling strategies on our XLNet RST parser.

DMs	Models	S	N	R	F
0	XLNet	95.6	85.8	74.3	73.8
	+DMP	95.7	86.6	75.1	74.5
1	XLNet	88.4	73.1	58.6	57.8
	+DMP	89.8	75.2	61.2	60.3
2	XLNet	76.7	61.8	47.3	47.0
	+DMP	78.5	63.0	45.3	44.8
3	XLNet	61.4	45.1	32.7	32.7
	+DMP	67.5	49.4	35.1	35.1
4+	XLNet	36.5	25.6	18.1	18.1
	+DMP	38.3	27.1	20.0	20.1

Table 6: Performances on the test set with different number of discourse markers in spans. "DMs" indicates the number of discourse markers in spans.

different articles, respectively. Table 5 shows the comparsion results. We find that these sampling strategies do not make difference to RST parsing.

Analysis by Number of Discourse Markers As mentioned earlier, DMP predicts the masked discourse markers of an EDU sequence and discourse markers are essential cues for RST parsing. Therefore, it is expected that the introduce of DMP may bring better performance for the spans containing discourse markers. As such, here we investigate the benefit by adopting DMP. Table 6 shows the performance of our XLNet RST parser with DMP and without DMP. The performances are improved significantly when spans contain discourse markers, which is consistent with our intuitions.

Effect of Different Masking Strategies Then we change the masking strategy in DMP to show how different masking strategies influences RST parsing. We use a random word set to replace the discourse marker set in DMP. The number of random words is the same as the number of discourse markers. Compared with discourse markers, these random words may be unable to offer key cues for discourse parsing. As shown in Table 7, the masking discourse markers strategy leads to performance improvement, and the masking random words strategy leads to slight performance degradation. Therefore, it is thus clear that discourse markers are useful for RST parsing.

Masking Strategies	S	N	R	F
Masking random words	73.2	62.8	51.6	50.6
Maksing discourse markers	75.3	65.0	53.8	52.6
XLNet	73.4	63.3	52.4	51.4

Table 7: Influence of different masking strategies on our XLNet RST parser.

Result on GUM Corpus Finally, we test our proposed RST parser on the GUM corpus (Zeldes, 2017) to show the domain generalization capability of our proposed RST parser. As shown in Table 8, the performance of our XLNet RST parser declines significantly for these out-of-domain articles, especially in conversation and vlog domains. By using our proposed second-stage EDU-level pretraining approach, the performance of the XLNet RST parser can be improved in academic, conversation, textbook, and whow domains significantly, and the performances declines slightly in bio, speech, and vlog domains. Therefore, there is still a lot of room for improvement in the generalization ability of our proposed RST parser.

5 Related Work

RST discourse parsing is an important task in the NLP community, which has been studied since early (Soricut and Marcu, 2003). Early studies adopt statistical models for this task, using human-designed discrete features (Hernault et al., 2010; Feng and Hirst, 2012; Joty et al., 2013; Feng and Hirst, 2014; Heilman and Sagae, 2015; Wang et al., 2017). Recently, several neural network models show great promising for this task (Braud et al., 2016, 2017; Liu and Lapata, 2017; Yu et al., 2018; Mabona et al., 2019; Zhang et al., 2020; Guz and Carenini, 2020). With PLMs such EMLo (Peters et al., 2018), BERT (Devlin et al., 2019), XLM-RoBERTa (CONNEAU and Lample, 2019), and XLNet (Yang et al., 2020), these neural RST parsers report high competitive performances (Liu et al., 2019a; Lin et al., 2019; Liu et al., 2020; Kobayashi et al., 2020; Zhang et al., 2021; Nguyen et al., 2021). We follow the line of these studies, using neural networks to perform RST parsing.

Recently, several studies aim to alleviate the mismatch between pre-trained language models and target tasks. Joshi et al. (2020) use a span masked language modeling to pre-train a language model for extraction question answering. Lewis et al. (2020) propose a pre-training approach for text generation tasks, which maps corrupt documents

Domains	XLNet				XLNet + NEP + DMP			
	S	**N**	**R**	**F**	**S**	**N**	**R**	**F**
academic	65.8	50.7	35.2	34.6	66.6 (+0.8)	52.0 (+0.3)	35.6 (+0.4)	35.0 (+0.4)
bio	57.3	41.3	32.0	31.6	57.2 (-0.1)	41.0 (-0.3)	31.0 (-1.0)	30.6 (-1.0)
conversation	36.8	23.7	12.6	12.2	32.7 (+0.9)	22.0 (-1.7)	12.6 (+0.0)	12.4 (+0.2)
fiction	57.3	41.0	28.1	27.4	57.1 (-0.2)	40.8 (-0.2)	28.1 (+0.0)	27.5 (+0.1)
interview	61.5	43.8	31.5	30.8	61.8 (+0.3)	42.8 (-1.0)	31.5 (+0.0)	30.9 (+0.1)
news	67.4	51.7	40.5	39.5	68.6 (+1.2)	52.5 (+0.8)	40.5 (+0.0)	39.6 (+0.1)
speech	71.5	58.2	45.6	45.6	70.7 (-0.8)	56.9 (-1.3)	44.2 (-1.4)	44.0 (-1.6)
textbook	64.7	51.3	39.8	39.2	65.9 (+1.2)	53.2 (+0.9)	41.8 (+1.0)	41.7 (+1.5)
vlog	47.0	33.2	21.1	20.2	44.7 (-2.3)	31.7 (-1.5)	19.2 (-1.9)	18.5 (-1.7)
voyage	65.0	45.3	31.6	30.6	64.9 (-0.1)	45.5 (+0.2)	31.8 (+0.2)	30.6 (+0.0)
whow	60.1	41.8	27.7	26.8	62.7 (+1.6)	44.5 (+2.7)	28.2 (+0.5)	27.4 (+0.6)

Table 8: Performances on GUM corpus.

to the original. Raffel et al. (2020) propose an unified text-to-text pre-training framework for several NLP tasks. Our work mainly inspired by above studies. In this paper, we propose a second-stage EDU-level pre-training approach to alleviate the mismatching between EDU-level RST parsing and sentence-level language modeling.

There are several studies have shown that pesudo data is useful for RST parsing. Huber and Carenini (2019) use pesudo RST discourse trees to train a RST parser, which generated by distant supervision on a sentiment classification. Kobayashi et al. (2021) improve RST parsing with large-scale sliver agreement subtrees, which is produced by a well trained RST parser. Zhang et al. (2021) train a top-down RST paser with predicted RST discourse trees. Above approaches requires a well trained RST parser to generate pesudo RST discourse trees. In this work, the generation of our pesudo data merely requires an EDU segmenter and discourse markers, without using a well trained RST parser to further generate pesudo RST discourse trees.

6 Conclusion and Future Work

We proposed a second-stage EDU-level pre-training approach for PLM-based RST discourse parser, reducing the mismatch between the EDU-level RST discourse parsing and the pre-training of sentence-level contextualized language modeling. In addition, we extended our RST discourse parser with a light bi-gram EDU modification, finding that it is able to exploit PLMs more effectively. Experiments on RST-DT (Carlson et al., 2001) showed that the proposed approach can bring significantly better performance for RST discourse parsing. We further conducted several experimental analysis to better understand the proposed approach.

The results on the RST-DT (Carlson et al., 2001) and the GUM (Zeldes, 2017) corpora suggest two possibilities for future research. First, although the XLNet RST parser obtains significantly improvements when the second-stage EDU-level pre-training approach is adopted, the Relation and the Full metrics of our RST parser are still lower than the best system. Future research might extend the second-stage EDU-level pre-training task, using pesudo RST discourse trees. Second, the generalization ability of our proposed RST parser needs to be improved in multi-domain scenarios. So in future we may continue to explore the issue of domain adapation in RST parsing on the basis of the second-stage EDU-level pre-training framework.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments, which help to improve the paper. This work was supported by National Natural Science Foundation of China under grants 62076173, U1836222, and 62176180.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Chloé Braud, Maximin Coavoux, and Anders Søgaard. 2017. Cross-lingual RST Discourse Parsing. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 292–304, Valencia, Spain. Association for Computational Linguistics.

Chloé Braud, Barbara Plank, and Anders Søgaard. 2016. Multi-view and multi-task training of RST discourse

parsers. In *Conference on Computational Linguistics (CoLing)*, pages 1903 – 1913, Osaka, Japan.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

Alexis CONNEAU and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Vanessa Wei Feng and Graeme Hirst. 2012. Text-level Discourse Parsing with Rich Linguistic Features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 60–68, Jeju Island, Korea. Association for Computational Linguistics.

Vanessa Wei Feng and Graeme Hirst. 2014. A Linear-Time Bottom-Up Discourse Parser with Constraints and Post-Editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 511–521, Baltimore, Maryland. Association for Computational Linguistics.

Bruce Fraser. 2009. An Account of Discourse Markers. *International Review of Pragmatics*, 1(2):293–320. Publisher: Brill.

Grigorii Guz and Giuseppe Carenini. 2020. Coreference for Discourse Parsing: A Neural Approach. In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pages 160–167, Online. Association for Computational Linguistics.

Katsuhiko Hayashi, Tsutomu Hirao, and Masaaki Nagata. 2016. Empirical comparison of dependency conversions for RST discourse trees. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 128–136, Los Angeles. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. DEBERTA: DECODING-ENHANCED BERT WITH DISENTANGLED ATTENTION.

Michael Heilman and Kenji Sagae. 2015. Fast Rhetorical Structure Theory Discourse Parsing. *arXiv:1505.02425 [cs]*. ArXiv: 1505.02425.

Hugo Hernault, Helmut Prendinger, David A. du Verle, and Mitsuru Ishizuka. 2010. HILDA: A Discourse Parser Using Support Vector Machine Classification. *dad*, 1(3):1–33.

Patrick Huber and Giuseppe Carenini. 2019. Predicting Discourse Structure using Distant Supervision from Sentiment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2306–2316, Hong Kong, China. Association for Computational Linguistics.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation Learning for Text-level Discourse Parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13–24, Baltimore, Maryland. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Shafiq Joty, Giuseppe Carenini, Raymond Ng, and Yashar Mehdad. 2013. Combining Intra- and Multi-sentential Rhetorical Parsing for Document-level Discourse Analysis. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 486–496, Sofia, Bulgaria. Association for Computational Linguistics.

Shafiq Joty, Giuseppe Carenini, and Raymond T. Ng. 2015. CODRA: A Novel Discriminative Framework for Rhetorical Analysis. *Computational Linguistics*, 41(3):385–435.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. *ICLR*.

Naoki Kobayashi, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. 2020. Top-Down RST Parsing Utilizing Granularity Levels in Documents. *AAAI*, 34(05):8099–8106.

Naoki Kobayashi, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. 2021. Improving Neural RST Parsing Model with Silver Agreement Subtrees. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1600–1612, Online. Association for Computational Linguistics.

Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2021. Top-down Discourse Parsing via Sequence Labelling. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 715–726, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Qi Li, Tianshi Li, and Baobao Chang. 2016. Discourse Parsing with Attention-based Hierarchical Neural Networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 362–371, Austin, Texas. Association for Computational Linguistics.

Xiang Lin, Shafiq Joty, Prathyusha Jwalapuram, and M Saiful Bari. 2019. A Unified Linear-Time Framework for Sentence-Level Discourse Parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4200, Florence, Italy. Association for Computational Linguistics.

Linlin Liu, Xiang Lin, Shafiq Joty, Simeng Han, and Lidong Bing. 2019a. Hierarchical Pointer Net Parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1007–1017, Hong Kong, China. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2017. Learning Contextually Informed Representations for Linear-Time Discourse Parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1289–1298, Copenhagen, Denmark. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*. ArXiv: 1907.11692.

Zhengyuan Liu, Ke Shi, and Nancy Chen. 2020. Multilingual Neural RST Discourse Parsing. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6730–6738, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Amandla Mabona, Laura Rimell, Stephen Clark, and Andreas Vlachos. 2019. Neural Generative Rhetorical Structure Parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2284–2295, Hong Kong, China. Association for Computational Linguistics.

Mathieu Morey, Philippe Muller, and Nicholas Asher. 2017. How much progress have we made on RST discourse parsing? A replication study of recent results on the RST-DT. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1319–1324, Copenhagen, Denmark. Association for Computational Linguistics.

Philippe Muller, Chloé Braud, and Mathieu Morey. 2019. ToNy: Contextual embeddings for accurate multilingual discourse segmentation of full documents. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 115–124, Minneapolis, MN. Association for Computational Linguistics.

Thanh-Tung Nguyen, Xuan-Phi Nguyen, Shafiq Joty, and Xiaoli Li. 2021. RST Parsing from Scratch. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1613–1625, Online. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Emily Pitler, Annie Louis, and Ani Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - ACL-IJCNLP '09*, volume 2, page 683, Suntec, Singapore. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Deborah Schiffrin. 1987. *Discourse Markers*. Cambridge University Press.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words

with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Wei Shi and Vera Demberg. 2019. Next Sentence Prediction helps Implicit Discourse Relation Classification within and across Domains. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5790–5796, Hong Kong, China. Association for Computational Linguistics.

Radu Soricut and Daniel Marcu. 2003. Sentence Level Discourse Parsing using Syntactic and Lexical Information. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 228–235.

Mihai Surdeanu, Tom Hicks, and Marco Antonio Valenzuela-Escárcega. 2015. Two Practical Rhetorical Structure Theory Parsers. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 1–5, Denver, Colorado. Association for Computational Linguistics.

Yizhong Wang, Sujian Li, and Houfeng Wang. 2017. A Two-Stage Parsing Method for Text-Level Discourse Analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 184–188, Vancouver, Canada. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv:1906.08237 [cs]*. ArXiv: 1906.08237.

Nan Yu, Meishan Zhang, and Guohong Fu. 2018. Transition-based Neural RST Parsing with Implicit Syntax Features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 559–570, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Amir Zeldes. 2017. The GUM corpus: creating multilayer resources in the classroom. *Lang Resources & Evaluation*, 51(3):581–612.

Longyin Zhang, Fang Kong, and Guodong Zhou. 2021. Adversarial Learning for Discourse Rhetorical Structure Parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3946–3957, Online. Association for Computational Linguistics.

Longyin Zhang, Yuqing Xing, Fang Kong, Peifeng Li, and Guodong Zhou. 2020. A Top-down Neural Architecture towards Text-level Parsing of Discourse Rhetorical Structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6386–6395, Online. Association for Computational Linguistics.