Toward Automatic Discourse Parsing of Student Writing Motivated by Neural Interpretation

James Fiacco
Language Technologies Institute
Carnegie Mellon University
jfiacco@cs.cmu.edu

Shiyan Jiang
Language Technologies Institute
North Carolina State University
sjiang24@ncsu.edu

David Adamson
Turnitin
dadamson@turnitin.com

Carolyn P. Rosé
Language Technologies Institute
Carnegie Mellon University
cprose@cs.cmu.edu

Abstract

Providing effective automatic essay feedback is necessary for offering writing instruction at a massive scale. In particular, feedback for promoting coherent flow of ideas in essays is critical. In this paper we propose a state-of-the-art method for automated analysis of structure and flow of writing, referred to as Rhetorical Structure Theory (RST) parsing. In so doing, we lay a foundation for a generalizable approach to automated writing feedback related to structure and flow. We address challenges in automated rhetorical analysis when applied to student writing and evaluate our novel RST parser model on both a recent student writing dataset and a standard benchmark RST parsing dataset.

1 Introduction

Automatic writing feedback technologies (e.g., MI Write (Palermo and Wilson, 2020), Criterion (Burstein et al., 2003), Coh-Metrix (McNamara et al., 2010), Writing Pal (Roscoe and McNamara, 2013), and Revision Assistant (West-Smith et al., 2018)) show promises in helping students to develop writing skills at scale. One challenging area where these technologies meet is in providing feedback for improving coherence of student essays (Cotos, 2011; Fiacco et al., 2019b). Efforts have been made to address the challenge of providing structural level feedback via automatically extracting discourse structure from essays (Burstein et al., 2003). Extracting hierarchical discourse structure and organization from documents has been shown to be valuable for numerous applications including text categorization, authorship attribution, and automatic essay feedback (Feng and Hirst, 2014b; Jiang et al., 2019).

A popular approach to analysis of the structure of writing that leverages principles of the dependency-based hierarchical nature of text and is common across genres is the discourse analytic framework known as Rhetorical Structure Theory (RST, described in section 3.1) (Mann and Thompson, 1988). RST holds the promise of providing specific structural writing feedback for free-form essays (Burstein et al., 2001). However, RST parsing has remained a challenging task due to the dearth of annotated data and the challenges of decision making for discourse relations based on local context (Mabona et al., 2019). This paper builds on the same theoretical foundation using a Neural Network Based RST parser as a means for automation. Specifically, we propose a novel neural approach to automated RST analysis that improves over the best previously published approach from the field of Language Technologies. In particular, of existing neural architectures for RST parsing, neural transition based parsers have been making headway (Yu et al., 2018; Mabona et al., 2019), however, at their core, transition parsers make parsing decisions locally. While they use recurrent models to construct their stacks and buffers, in practice, recurrent models have been shown to primarily to use very near context (Khandelwal et al., 2018). This is a limitation for discourse parsing where knowledge about the document as a whole may provide essential context for judging relations.

We therefore propose and evaluate two improvements to the neural transition parser paradigm that provide better performance, both on standard RST parsing and on student writing by utilizing the limited data more efficiently:

1. By adding a co-task of predicting the most nuclear unit of the RST tree, we can increase the model's performance with the intuition that it may incentivize the model to maintain a broader document context that it can use for predicting individual tree spans and nuclear-

Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022), pages 204 - 215
July 15, 2022 ©2022 Association for Computational Linguistics

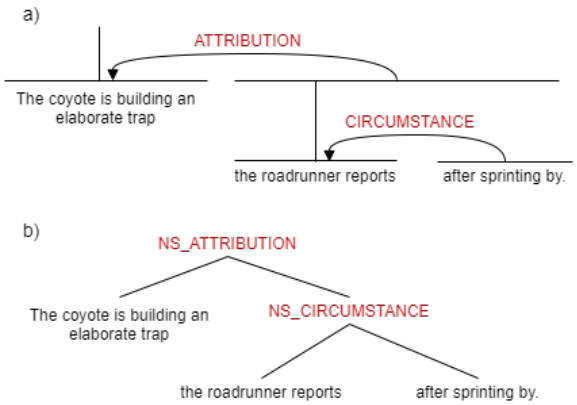

Figure 1: Example RST tree fragment with nuclearity and relations. a) The traditional depiction of an RST tree structure. b) The RST tree form corresponding to the labeled attachment decisions of (a).

ity.

2. By selectively introducing parser states from a previously trained parser into a new model during training, we can guide the training of the new model towards better performance on less structured writing.

The first improvement builds on the general concept of multitask learning in NLP (Bingel and Søgaard, 2017; Peng et al., 2017) and the intuition that a topic-like sentence, as a common key component in many writing assessments and rubrics (Aull, 2015), may provide important contextual information to aid local parser decision-making. The second improvement suggests a potential for a reflective form of neural network learning related neural component reuse that grows out of state-of-the-art work in neural network interpretation.

In the following sections, we evaluate our parsing model on both the standard English RST Discourse Treebank (RST-DT) (Carlson et al., 2003) and a more recent RST dataset of student writing (Jiang et al., 2019).

2 Rhetorical Structure Theory

Rhetorical Structure Theory decomposes a document into basic units of analysis called elementary discourse units (EDU) that can be combined through rhetorical relations between units into larger composite units (Mann and Thompson, 1988). Thus, the rhetorical relations combine to build a hierarchical tree structure that represents the overall structure of the document (Figure 1a). Each relation has one (mononuclear) or more (multinuclear) nuclei where a nucleus is an essential span which, if deleted, would leave the remaining text incoherent. Mononuclear relations have satellites that are related to the nucleus by means of a rhetorical relation. They play a supporting role, and are therefore not necessary for coherence of the document. Each node of the tree represents a relation tuple $\langle S, N, R \rangle$ where S is the span, N is the direction of nuclearity, and R is the relation label. This is more readily seen in Figure 1b which depicts an alternate representation of the RST tree structure.

RST has a long history (Mann and Thompson, 1988), and its original formulation continues to be treated as authoritative. However, for some types of writing, especially student writing, additional and combined relations have been proposed in order to bring the set of used relations in line with the writing practices that are applicable to the corpus (Jiang et al., 2019).

3 Related Work

This paper makes its fundamental contribution to work on automated feedback for student writing by expanding analysis capabilities that lay a foundation for a new form of support. Our technical contribution is grounded within the field of neural network modeling, contributing to work on neural approaches to Rhetorical structure analysis and leveraging approaches originating in the area of neural model interpretation.

An effective method for performing discourse parsing has been to utilize techniques from syntactic parsing and applying them at the document level. While RST parsing research has more frequently seen parsers influenced by another approach referred to as constituency parsing, it was shown that using techniques pioneered for dependency parsing could be as or more effective (Morey et al., 2018). As methods for RST parsing moved from those that rely on discourse markers and hand-coded rules (Marcu, 2000; LeThanh et al., 2004) to those that rely on deep learning (Li et al., 2014; Ji and Eisenstein, 2014; Braud et al., 2017), many of the improvements have been through techniques from syntactic parsing (Soricut and Marcu, 2003; Luong et al., 2013). In a similar way, our work builds on past RST parsers using neural transition parsing (Yu et al., 2018; Mabona et al., 2019). We extend this work by leveraging another area of neural network research, namely neural network interpretation, in order to yield a reflective form of learning

that improves performance by leveraging lessons learned in an earlier stage of the training, as in a stage-based regression.

Neural pathways (Fiacco et al., 2019a) refer to a method for pinpointing sets of a model's neurons that function together in groups. These groups of neurons are referred to as pathways because they cut across architectural layers and allow representation of the flow of activation through a network, potentially from input all the way to output. For our application, we follow the original authors and use PCA (Hotelling, 1933) for this step as the resulting factor loadings (DeCoster, 1998) can then be used to determine which neurons belong to each pathway, and that forms the basis for our pruning approach. The remaining stages of this approach are not used in the work reported here but offer opportunities for promising follow up work.

In offering an abstraction over the details of a neural model, this approach offers the possibility of identifying portions of learned networks that can be dissected from the network as a whole and then reused as pre-packaged basic functionality within a more complex model learned at a later stage. Thus, we seek to harvest components pretrained on a simpler dataset to aid in learning a more robust model later on a more challenging dataset. While a deep dive into the differences between the learned functions of an RST parser trained on a relatively clean standard dataset and one trained solely on a noisier student writing dataset is beyond the scope of this paper, we will demonstrate that this work provides inspiration for development of what we will refer to as a neural pruning method that protects important simple generalizations while enabling accounting for complex special cases as well, and to represent an awareness of the difference between these in the final decision making.

4 A Corpus of Student Writing

In this section, we first offer more understanding about RST and then describe a corpus of student writing that has been annotated with RST.

4.1 Applying Rhetorical Structure Theory to Student Writing

Since we are using a neural approach, annotated data is necessary for training. The English RST Discourse Treebank is a common benchmark dataset for RST parsing. It includes 385 articles from the Wall Street Journal (Carlson et al., 2003), consti-

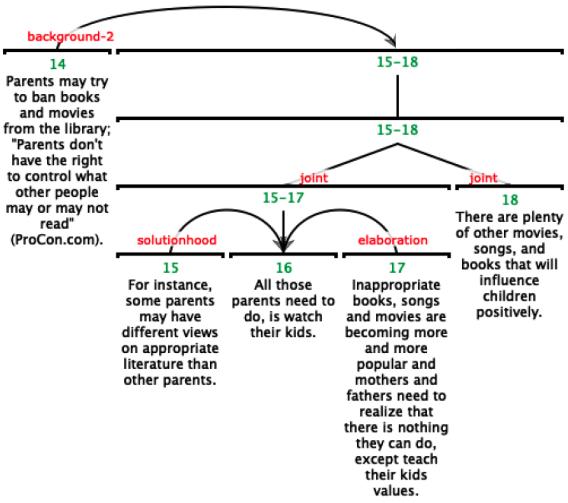

Figure 2: Example RST tree of a fragment of student writing.

tuting approximately 180,000 words of texts and covering a wide range of topics, such as finance and arts. These articles were created by professional writers, and are thus typically well-written, consistently structured, and copy-edited. (Palmer et al., 2010).

We also consider an RST corpus of less-polished student writing (Jiang et al., 2019). The corpus consists of 274 essays collected from Turnitin Revision Assistant (West-Smith et al., 2018), responding to standards-aligned formative writing tasks (Valencia and Wixson, 2001). These tasks cover a range of genres, including literary analysis, historical analysis, argumentative, and informative writing. For example, one writing task asks the student write an essay to the head of the school board, to argue whether competitive sports are more helpful or harmful to young people. These essays are drawn from a diverse set of secondary classrooms across the United States, representing a broad range of writing skills and student backgrounds. We hold out 25 documents as a development set, and 28 documents as a test set.

4.2 Comparison of Datasets

As we bridge between work on the original corpus and the student writing corpus, we must consider differences in properties. In addition to unconventional grammar and usage, many developing student essays lack clear cohesion or structure. These issues may make the modeling task more challenging than with the relatively clean RST-DT dataset. Common organizational issues in the corpus include (1) essays lacking transitional phrases

(e.g., "However," or "In conclusion"), or transition words used inappropriately; (2) pronoun reference ambiguity; (3) paragraphs where the topic sentence is not clearly indicated, or where there are multiple main ideas (and sometimes contradictory ideas) in one paragraph; (4) sentences not presented in a logical progression. These areas of focus for developing writers are also highlighted in the literature (de Jong and Harper, 2005). Ambiguous and weakly structured essays may indicate an opportunity for automated feedback, but they also pose challenges for the parsing task.

The prevalence of the JOINT relation captures some of the difference between RST-DT and the Turnitin corpus. JOINT indicates a lack of rhetorical relations between nuclei. It indicates that there is no relation that could describe the connection between sentences (Jiang et al., 2019). In newspaper articles, this lack of connection is very rare. However, in student essays the lack of coherent rhetorical relations is common because of the wide range of experience among developing writers.

4.3 Designing Feedback from RST Relations

Three veteran secondary English teachers provided feedback and commentary on the structure and flow of 18 essays from the Turnitin dataset. Their comments reveal a handful of organizing principles and focal points for structure-driven feedback that provide guidance on how an RST style analysis could form the foundation for automated feedback.

In particular, almost all of the suggestions for improvement highlight a lack of connection or a break in flow between units of the essay. Some of these comments addressed breaks between consecutive sentences within a paragraph, for example "Strange jump in focus here... The rest of the intro does not lead to this statement naturally," and "Immediate departure from the initial question in sentence 1." Other comments, in contrast, deal specifically with the logical flow between whole paragraphs: "Transitions between paragraphs are relatively non-existent and make for pretty large jumps from one topic to another" and "To keep the organizational structure clear, this needs a more explicit connection to the introduction and thesis, including attention to the two distinct texts."

These comments, anchored to sentences or paragraphs in the student texts, roughly correspond to the locations of JOINT relations in the gold RST annotations. For example, Figure 2 is part of a gold RST annotation of student-generated essays. This essay has five paragraphs. The subtree (sentence 14-18) is a part of the third paragraph arguing that parents should guide children in evaluating "inappropriate" books, instead of pushing libraries to ban them. While sentence 18 is related to the overall argument in this paragraph, the connection between sentence 18 and other sentences is not clear. Potential automated feedback could be: "There may be ideas in this sentence that don't clearly relate to the paragraph's focus. Connect these ideas to the paragraph's main point by adding transition words, or consider whether this sentence should be revised or removed." This example shows that identifying the missing link (referring to the relation of JOINT) holds the promise of triggering meaningful revision actions. As our previous studies suggested that teachers viewed the structure of a developing essay as an archipelago of internally cohesive text islands[cite book chapter], we seek to validate RST's suitability to represent this segmentation. Using the locations of teacher comments as gold-truth segment boundaries using WindowDiff (Pevzner and Hearst, 2002) and Beeferman's P_k (Beeferman et al., 1999). Both WindowDiff and P_k range from 0 to 1 where a lower value indicates a lower probability that a given sentence is assigned to an incorrect segment, in practice a value of 0.2 to 0.4 is considered reasonable in state-of-the-art systems (Badjatiya et al., 2018). We observe a mean WindowDiff of 0.31 and P_k of 0.34 between these teacher-reviewed essays and the RST JOINT annotations. This suggests a plausible upper bound on an RST parser's ability to identify these critical boundaries.

5 Improving and Validating RST Parsing for Student Writing

In this section, we begin with and then improve on the best previously published approach in automating RST analysis for writing. Transition parsers are common among state-of-the-art models for discourse parsing with RST in the past several years. Their power lies in their ability to make strong local decisions about the next action the parser must take given an embedding that, because of recurrent neural models, has the capacity to contain features from the whole document. However, recurrent neural networks often do not in practice retain sufficient context for long range dependencies (Bahdanau et al., 2014; Khandelwal et al., 2018). We

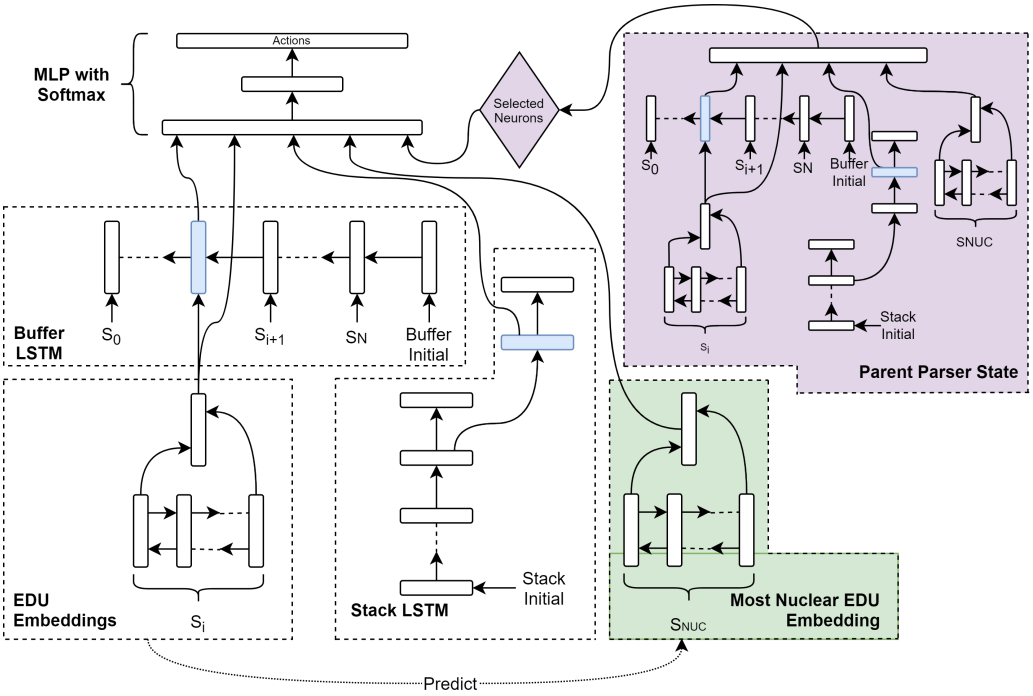

Figure 3: Diagram of neural transition parser model architecture for RST parsing augmented with our changes (shaded purple and green). The parent parser state (purple) has the same basic architecture as the rest of the diagram with the exception of having another parent parser state component. The dotted line from EDU Embedding to Most Nuclear EDU Embedding (green) indicates choice made by the model for which EDU to use.

address this by providing an additional embedding for the predicted most nuclear sentence of the document to provide a reference point for the parsing decisions. Furthermore, inspired by neural interpretation techniques, we further augment the model with a two stage parsing approach that allows the second stage of the model to learn from mistakes made by the first.

5.1 Neural Transition Parsing Model

The model presented in this work is based on a recurrent neural network based RST parser (Yu et al., 2018). For the benefit of the reader this subsection provides an overview of the base model, however, for a full mechanical description see their paper. Our augmentations of the model follow in the remaining subsections.

The model constructs a neural representation that is used to decide whether to make a SHIFT or REDUCE action analogous to those in a simple LR-parser (Knuth, 1965). Furthermore, the model maintains a neural analogue to a stack and buffer to track progress through the parse, which is illustrated in the unshaded regions of Figure 3.

EDU Embedding: Each sentence in the document is embedded using a BiLSTM over word embeddings for each word in the EDU. The final states of the forward and backward LSTMs are used as the EDU representation.

Dependency Parse Embedding: In addition to the embedding generated by the BiLSTM, an embedding of syntactic information was included (Braud et al., 2017; Mabona et al., 2019). The information was integrated via concatenating the produced arc embedding from the dependency parse obtained from a strong neural dependency parser (Dozat and Manning, 2017) with the output from the BiLSTM above.

Buffer: The buffer is an LSTM that inputs each EDU embedding from the end of the document to the beginning. Each state is stored in memory such that it can be accessed sequentially as items are removed from the buffer. Each state of the buffer is therefore an aggregate representation of all of the EDUs from the current EDU to the end of the document.

Stack: The stack is a Stack LSTM (Swayamdipta et al., 2016). The stack state is updated via the result of an MLP given the two stacks states popped off the stack during a REDUCE action procedure. If an item is popped off the stack, the stack state is updated to the output state of the LSTM of the

previous cell.

Action and Relation Prediction: At each timestep the parser either predicts a SHIFT action or one of the many REDUCE actions. Each REDUCE action has an associated relation label and predicting the correct REDUCE action amounts to choosing the correct relation for the current subtree. The prediction is made by a multi-layer perceptron (MLP) that is provided a concatenation of the EDU embedding, the current neural state of the buffer, the current neural state of the stack, and additional neural representations that will be described in depth in the next sections. The input layer to the MLP will be referred to as the parser state at a given time. For each action, a deterministic procedure is executed in line with the transition parsing paradigm. In the case where there is only one possible action, the model is forced to use that action without choice.

5.2 Most Nuclear EDU Embedding

To provide the model a reference for making parsing decisions for a given document, we include in the parser state an EDU embedding of the predicted most nuclear EDU. Formally, we consider the most nuclear EDU the leaf node of the RST tree that is reached when, starting at the root node, one follows the direction of nuclearity at each branch. For multinuclear nodes, we arbitrarily take the left branch. In Figure 1, the most nuclear EDU would be "The coyote is building an elaborate trap."

The most nuclear EDU S_{NUC} is selected by the model by choosing the EDU with the maximum score computed by an MLP given the EDU embedding and choosing the highest scoring sentence. This can be formalized as:

$$S_{NUC} = \arg\max_{\mathbf{s} \in S} W \cdot \mathbf{s}$$

Where S is a set of all of the sentence EDU computed by the neural transition parser.

The most nuclear EDU embedding is constructed via a BiLSTM in much the same manner as the EDU embeddings in the neural transition parser. This BiLSTM has its own set of learned parameters, though it uses shared word embeddings as those used for the EDU embeddings.

Because there is only one predicted most nuclear EDU for a document, the effective training samples for this embedding is equal to the number of documents in the training set rather than the number of EDUs. Because of this, it is necessary to restrict the size of the embedding to prevent overfitting. Furthermore, the error from the RST parsing task cannot backpropagate to W through the argmax so we include a separate error signal for predicting the correct most nuclear EDU. The most nuclear EDU of a document can be trivially obtained from the gold trees.

5.3 Parent Parser State

Recent work has shown there is evidence that neural models may be learning general heuristics and memorizing exceptions to those heuristics that increase performance on a given task (Fiacco et al., 2019a). Assuming this is the case, we attempt to exploit this behavior to offload some of the complexity of learning the RST discourse parsing task into multiple phases of training. A fully trained parent model, which includes all of the features in the previous sections, is executed concurrently to the child model and a subset of the parser state of the parent model is concatenated with the parser state of the child model.

The parser state for the parent model is updated along with the child model using the action chosen by the child model, though with its own stack and buffer representations. This ensures that even if the parent and child models diverge in their predicted actions, the parser states are consistent. Maintaining this consistency is important for the neural transition parser as the representation of the stack can contain a representation of a larger segment of the document than just a single EDU.

Neuron Selection via Pathways: For datasets with noisy data, we prune the parser state from the parent model to only use the dimensions of the state that correspond to the neurons that are part of the neural pathways that explain the most variance of the model. The intuition for this pruning is that the groups of neurons that explain the largest amount of variance in the model will regularize the model via eliminating overfitted parameters.

These neurons are obtained by extracting the parser state for each training instance and constructing an activation matrix with the dimensions of the parser state as columns and the training instances as rows. A PCA is performed over the matrix, and the subset of resulting factors that cumulatively explain more than a tunable threshold of the variance are chosen as the subset of pathways of interest. For each selected factor, the factor loadings of each neuron are computed and the N neurons with the

highest loadings are added to the set of neurons to be transferred. The value of N can be tuned by optimizing performance on a validation set.

5.4 Training

There are three phases to the training of the model: parent model training, neuron selection, and child model training. The procedure for training the parent and child models are identical except for the usage of the parent neurons as features for the child model. The neuron selection phase is only applicable for the noisier Turnitin data and is described in the Parent Parser State section.

There are three objectives that are optimized using negative log likelihood loss during the model training. The first training objective (L_m) is predicting the most nuclear EDU at the document level. The second objective (L_n), at the action level, is to predict the nuclearity of each relation given the parser state. This objective affects how the model composes the embeddings when combining via a REDUCE action. The final training objective, (L_a), is to choose the correct action given the parser state. We do not fine tune the embedding from the dependency parser during training. The third phase of training follows the same procedure as the first phase with selected neurons from the parent parser state included. The final loss for a document is described as:

$$L = \alpha_m L_m + \alpha_n \sum_A L_n + \alpha_a \sum_A L_a$$

where A is the set of all actions required for the parse and each α is a scaling factor that can be tuned for each loss.

For noisy datasets, an additional step is required for the training procedure; the neurons that will be used by the child model must be selected. This is performed by computing the neural pathways of the parent model using the parser state via PCA. The pathways that explain the most variance are chosen and the heaviest loaded neurons on those pathways are selected. During training, no gradient is passed back to the parent model so the neuron selection process need not be continuous nor differentiable. Training the child model thereby uses the parser state of the parent model as though it were a fixed input.

6 Experiments

We provide three quantitative evaluations of our method: first, in order to compare our parser to previous RST parsers, we train and evaluate our parser on the English RST-DT corpus. Second, we provide an ablation study of the added components of our model along with the model we used as a base. The ablation study uses the same test set as the first experiment, so results are directly comparable. Lastly, we train another version of our model on the Turnitin dataset, which has a very different set of properties when compared to the RST-DT corpus. This last set of experiments is designed to test the ability of the model to handle unpolished, less structured text. The model is compared to the strongest baseline from the RST-DT corpus retrained on the Turnitin dataset.

6.1 Evaluation Metrics

The evaluations of this work follow the setup described by a recent metric enhancement for RST (Morey et al., 2017) and, for consistency, only compare to models that were included in that replication study or use the same evaluation method. The reason for this restriction is that it was found that RST Parseval, the previous standard evaluation metric, artificially raised scores and had been used inconsistently (Morey et al., 2017). Our models are therefore evaluated using micro-averaged F1 scores on labeled attachment decisions for the four standard metrics: span attachments (S), span attachments with nuclearity (N), span attachments with relations (R), and span attachments with both nuclearity and relation labels (F).

6.2 Implementation Details

The models were implemented using the DyNet neural network toolkit (Neubig et al., 2017). Training was performed on a NVIDIA GTX 1080. Early stopping was performed based on the F1 scores of the model without an oracle on the development set, with a patience of 3. The ADAM optimizer (Kingma and Ba, 2014) is used for training with a learning rate of 0.001. Dropout (Srivastava et al., 2014) is used for regularization and a dropout of 0.3 is applied to each hidden layer. All tunable α hyperparameters were left at 1.

For the RST parsing models, word embeddings for both the parent and child models were randomly initialized with 128 dimensional vectors. Each LSTM in the parent model had 256 dimensions while in the child model, each LSTM had 512 dimensions. For neuron selection, the 16 neurons with the highest factor loadings from the PCA were chosen for each pathway that explained more than

	F1 Scores			
Model	*Span*	*Nuclearity*	*Relation*	*Full*
JI & EISENSTEIN (2014)(JI AND EISENSTEIN, 2014)*	64.1	54.2	**46.8**	**46.3**
FENG & HIRST (2014)(FENG AND HIRST, 2014A)*	68.6	55.9	45.8	44.6
LI ET AL. (2016) (LI ET AL., 2016)*	64.5	54.0	38.1	36.6
BRAUD ET AL. (2016) (BRAUD ET AL., 2016)*	59.5	47.2	34.7	34.3
BRAUD ET AL. (2017)(BRAUD ET AL., 2017)*	62.7	54.5	45.5	45.1
MABON ET AL. (2019) (MABONA ET AL., 2019)	67.1	57.4	45.5	45.0
ZHANG ET AL. (2020)(ZHANG ET AL., 2020)	67.2	55.5	45.3	44.3
OUR MODEL	**71.7**	**60.3**	44.5	44.3
-DEPENDENCY PARSE EMBEDDINGS	71.2	58.4	43.6	43.6
-PARENT PARSER STATE	70.2	57.2	43.0	42.9
-MOST NUCLEAR EDU EMBEDDINGS	68.4	57.2	42.7	42.4
TRANSITION PARSER ONLY	67.2	53.7	39.9	39.8

Table 1: RST-DT test set micro-averaged F1 scores for labeled attachment decisions for our model with varying components removed. Parsers from previous work are reported as they appear in their original publication, with the exception of those marked with an * where the reported results come from the replication study with the improved metric (Morey et al., 2017).

1% of the model variance. The number of dimensions for the PCA was tuned to explain 90% of the variance in neuron activations.

The dependency parser was pretrained on Universal Dependencies v1 (Nivre et al., 2016) derived from the Penn Treebank 3 (Marcus et al., 1999) using version 3.9.2 of the Stanford Universal Dependency Converter. Word embeddings and label MLP dimensions were set to 64 while the recurrent layers and the arc MLP layers were set to 128. Choice of optimizer, dropout, and early stopping criteria were the same for the dependency parser pretraining.

7 Evaluation

7.1 Parsing Results

Table 1 shows the performance across parsers on the labeled attachments metrics for the RST-DT test set. We include reported metrics for several models beyond the best baseline in order to provide a comprehensive view of recent work in the field, including other neural based models. The best version of our model gains a 4.5% increase in F1 score for the span metric (S) and a 7.9% increase in F1 score for combined span and nuclearity metric (N) in comparison with the Feng Hirst (Feng and Hirst, 2014a) model, the next best model for those metrics. The increase was gained with a competitive, albeit 2.8% lower span and relation metric (R).

Furthermore, we achieve these results with only the dependency parser as external data. Pretrained embeddings of any kind were not required for either the dependency parser nor the final RST parser and were found to not contribute empirically. Using pretrained GloVe embeddings (Pennington et al., 2014) do not significantly improve the performance over random initialization.

7.2 Ablation

We evaluated the model with key components removed to evaluate the effects of each of those components on the final performance of the model. The components ablated were the dependency parser embedding, the most nuclear EDU embedding, and the parent parser state. These results are presented in the lower section of Table 1.

From the results we see that the largest contributor to our model's performance was the inclusion of the most nuclear EDU co-task without which, the parser does not outperform the previous state-of-the-art on any metric. The parent model's parser state as a feature for action and relation prediction had the next largest effect with the span and nuclearity metric (N) falling to the same level as when the most nuclear EDU embedding was not used. Lastly, the syntactic information carried in the dependency parser embedding contributed the least, but still had a significant effect on all metrics.

We also present the performance of the base model, our implementation of the base neural transition parser (Yu et al., 2018) with the same settings

	F1 Scores			
Model	*Span*	*Nuclearity*	*Relation*	*Full*
RST-DT				
JI & EISENSTEIN (2014)(JI AND EISENSTEIN, 2014)*	64.1	54.2	**46.8**	**46.3**
OUR MODEL	**71.7**	**60.3**	44.5	44.3
OUR MODEL (W/ NEURON SELECTION)	70.6	59.7	44.4	44.3
Turnitin Corpus				
JI & EISENSTEIN (2014)(JI AND EISENSTEIN, 2014)*	**56.1**	**33.4**	1.2	1.1
OUR MODEL	44.1	22.9	14.0	12.4
OUR MODEL (W/ NEURON SELECTION)	47.6	28.4	**18.0**	**17.0**

Table 2: Test set micro-averaged F1 scores for labeled attachment decisions for our model on the RST-DT corpus and the Turnitin dataset. The models were evaluated on each dataset both with and without pruning the parent parser state (W/ NEURON SELECTION).

as each of the other models from the ablation study. While it has competitive performance to prior work on the span only metric (S), all of the metrics are considerably lower than the final model. All ablation conditions were significantly different from the final model with $p < 0.05$.

7.3 Model Robustness with Neuron Selection

As our goal is to facilitate automatic essay feedback with RST, we evaluated our model, as well as the best performing model for predicting Relations, on the Turnitin dataset to test the ability of each model to handle the less consistently structured student writing data. Table 2 shows a comparison of the model performance on both the RST-DT corpus and the Turnitin dataset. For each dataset, we include versions of our model that use neuron selection as described in the Parent Parser State section and without. Each model was trained on the RST-DT dataset and fine-tuned on the Turnitin Corpus. All models saw significant degradation of performance on the student writing data as compared to the Wall Street Journal articles. Our model variations both has significantly ($p < 0.001$) less loss of performance for Relation prediction compared to the previous best performing model. Our model that used the neuron selection significantly ($p < 0.001$) increased performance on the Turnitin dataset compared to the model without.

Qualitatively, the JOINT relation was the most problematic for each parser as it was being considerably over-generated despite being only the 5th most common relation type. For variably structured writing such as student essays, understanding these conditions would likely go the furthest for improving RST parsing performance.

8 Conclusion

We presented two principal augmentations to neural transition parsers for RST that resulted in a 7.9% increase in span prediction and a 4.5% increase in nuclearity prediction. These improvements were made while remaining competitive on relation prediction, though no improvement was observed for that metric. Furthermore, we evaluated our model on an alternate, noisier dataset. We found that on this dataset our model had more accurate relation predictions than past approaches from the inclusion of a neuron selection step between the training of parent and child models in a boosting-like neural ensemble enhancement.

For future work, we want to empirically verify that the prediction of structural breaks (JOINT relations) in student writing align with teacher-identified organization feedback. This can enable automated essay feedback on the absence of structure, providing support where it's needed most. Furthermore, conveying the necessary information contained within RST trees to students and teachers provides an additional rich area of inquiry. It is worthwhile to further explore how prospective users respond to the technological instruction support to facilitate students' ability to locate places for revision and teachers' ability to integrate the automated feedback into their instruction.

Acknowledgements

This work was supported in part by NSF grant DRL 1949110 and funding from the Schmidt foundation.

References

Laura Aull. 2015. Connecting writing and language in assessment: Examining style, tone, and argument in the us common core standards and in exemplary student writing. *Assessing writing*, 24:59–73.

Pinkesh Badjatiya, Litton J Kurisinkel, Manish Gupta, and Vasudeva Varma. 2018. Attention-based neural text segmentation. In *European Conference on Information Retrieval*, pages 180–193. Springer.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine learning*, 34(1):177–210.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169.

Chloé Braud, Ophélie Lacroix, and Anders Søgaard. 2017. Cross-lingual and cross-domain discourse segmentation of entire documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 237–243, Vancouver, Canada. Association for Computational Linguistics.

Chloé Braud, Barbara Plank, and Anders Søgaard. 2016. Multi-view and multi-task training of rst discourse parsers. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1903–1913.

Jill Burstein, Daniel Marcu, Slava Andreyev, and Martin Chodorow. 2001. Towards automatic classification of discourse elements in essays. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pages 98–105.

Jill Burstein, Daniel Marcu, and Kevin Knight. 2003. Finding the write stuff: Automatic identification of discourse structure in student essays. *IEEE Intelligent Systems*, 18(1):32–39.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2003. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Current and new directions in discourse and dialogue*, pages 85–112. Springer.

Elena Cotos. 2011. Potential of automated writing evaluation feedback. *Calico Journal*, 28(2):420–459.

Ester J de Jong and Candace A Harper. 2005. Preparing mainstream teachers for english-language learners: Is being a good teacher good enough? *Teacher Education Quarterly*, 32(2):101–124.

Jamie DeCoster. 1998. Overview of factor analysis.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *ICLR 2017*.

Vanessa Wei Feng and Graeme Hirst. 2014a. 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.

Vanessa Wei Feng and Graeme Hirst. 2014b. Patterns of local discourse coherence as a feature for authorship attribution. *Literary and Linguistic Computing*, 29(2):191–198.

James Fiacco, Samridhi Choudhary, and Carolyn Rose. 2019a. Deep neural model inspection and comparison via functional neuron pathways. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5754–5764.

James Fiacco, Elena Cotos, and Carolyn Rose. 2019b. Towards enabling feedback on rhetorical structure with neural sequence models. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 310–319.

Harold Hotelling. 1933. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

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.

Shiyan Jiang, Kexin Yang, Chandrakumari Suvarna, Pooja Casula, Mingtong Zhang, and Carolyn Rose. 2019. Applying rhetorical structure theory to student essays for providing automated writing feedback. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 163–168.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 284–294.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Donald E Knuth. 1965. On the translation of languages from left to right. *Information and control*, 8(6):607–639.

Huong LeThanh, Geetha Abeysinghe, and Christian Huyck. 2004. Generating discourse structures for written texts. In *Proceedings of the 20th international conference on Computational Linguistics*, page 329. Association for Computational Linguistics.

Jiwei Li, Rumeng Li, and Eduard Hovy. 2014. Recursive deep models for discourse parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2061–2069.

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.

Minh-Thang Luong, Michael C Frank, and Mark Johnson. 2013. Parsing entire discourses as very long strings: Capturing topic continuity in grounded language learning. *Transactions of the Association for Computational Linguistics*, 1:315–326.

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.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Daniel Marcu. 2000. The rhetorical parsing of unrestricted texts: A surface-based approach. *Computational linguistics*, 26(3):395–448.

Mitchell P Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. 1999. Penn treebank-3. *Linguistic Data Consortium, LDC99T42, University of Pennsylvania*.

Danielle S McNamara, Max M Louwerse, Philip M McCarthy, and Arthur C Graesser. 2010. Coh-metrix: Capturing linguistic features of cohesion. *Discourse Processes*, 47(4):292–330.

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.

Mathieu Morey, Philippe Muller, and Nicholas Asher. 2018. A dependency perspective on RST discourse parsing and evaluation. *Computational Linguistics*, 44(2):197–235.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Corey Palermo and Joshua Wilson. 2020. Implementing automated writing evaluation in different instructional contexts: A mixed-methods study. *Journal of Writing Research*, 12(1).

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103.

Hao Peng, Sam Thomson, and Noah A Smith. 2017. Deep multitask learning for semantic dependency parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2037–2048.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Lev Pevzner and Marti A Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36.

Rod D Roscoe and Danielle S McNamara. 2013. Writing pal: Feasibility of an intelligent writing strategy tutor in the high school classroom. *Journal of Educational Psychology*, 105(4):1010.

Radu Soricut and Daniel Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 149–156. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Swabha Swayamdipta, Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2016. Greedy, joint syntactic-semantic parsing with stack lstms. *arXiv preprint arXiv:1606.08954*.

Sheila W. Valencia and Karen K. Wixson. 2001. Commentary: Inside english/language arts standards: What's in a grade? *Reading Research Quarterly*, 36(2):202–217.

Patti West-Smith, Stephanie Butler, and Elijah Mayfield. 2018. Trustworthy automated essay scoring without explicit construct validity. In *AAAI Spring Symposia*.

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.

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. *arXiv preprint arXiv:2005.02680*.