

W-RST: Towards a Weighted RST-style Discourse Framework

Patrick Huber*, Wen Xiao*, Giuseppe Carenini

Department of Computer Science

University of British Columbia

Vancouver, BC, Canada, V6T 1Z4

{huberpat, xiaowen3, carenini}@cs.ubc.ca

Abstract

Aiming for a better integration of data-driven and linguistically-inspired approaches, we explore whether RST Nuclearity, assigning a binary assessment of importance between text segments, can be replaced by automatically generated, real-valued scores, in what we call a Weighted-RST framework. In particular, we find that weighted discourse trees from auxiliary tasks can benefit key NLP downstream applications compared to nuclearity-centered approaches. We further show that real-valued importance distributions partially and interestingly align with the assessment and uncertainty of human annotators.

1 Introduction

Ideally, research in Natural Language Processing (NLP) should balance and integrate findings from machine learning approaches with insights and theories from linguistics. With the enormous success of data-driven approaches over the last decades, this balance has arguably and excessively shifted, with linguistic theories playing a less and less critical role. Even more importantly, there are only little attempts made to improve such theories in light of recent empirical results.

In the context of discourse, two main theories have emerged in the past: The Rhetorical Structure Theory (RST) (Carlson et al., 2002) and PDTB (Prasad et al., 2008). In this paper, we focus on RST, exploring whether the underlying theory can be refined in a data-driven manner.

In general, RST postulates a complete discourse tree for a given document. To obtain this formal representation as a projective constituency tree, a given document is first separated into so called Elementary Discourse Units (or short EDUs), representing clause-like sentence fragments of the input

document. Afterwards, the discourse tree is built by hierarchically aggregating EDUs into larger constituents annotated with an importance indicator (in RST called nuclearity) and a relation holding between siblings in the aggregation. The nuclearity attribute in RST thereby assigns each sub-tree either a nucleus-attribute, indicating central importance of the sub-tree in the context of the document, or a satellite-attribute, categorizing the sub-tree as of peripheral importance. The relation attribute further characterizes the connection between sub-trees (e.g. Elaboration, Cause, Contradiction).

One central requirement of the RST discourse theory, as for all linguistic theories, is that a trained human should be able to specify and interpret the discourse representations. While this is a clear advantage when trying to generate explainable outcomes, it also introduces problematic, human-centered simplifications; the most radical of which is arguably the nuclearity attribute, indicating the importance among siblings.

Intuitively, such a coarse (binary) importance assessment does not allow to represent nuanced differences regarding sub-tree importance, which can potentially be critical for downstream tasks. For instance, the importance of two nuclei siblings is rather intuitive to interpret. However, having siblings annotated as “nucleus-satellite” or “satellite-nucleus” leaves the question on how much more important the nucleus sub-tree is compared to the satellite, as shown in Figure 1. In general, it is unclear (and unlikely) that the actual importance distributions between siblings with the same nuclearity attribution are consistent.

Based on this observation, we investigate the potential of replacing the binary nuclearity assessment postulated by RST with automatically generated, real-valued importance scores in a new, **Weighted-RST** framework. In contrast with previous work that has assumed RST and developed

*Equal contribution.

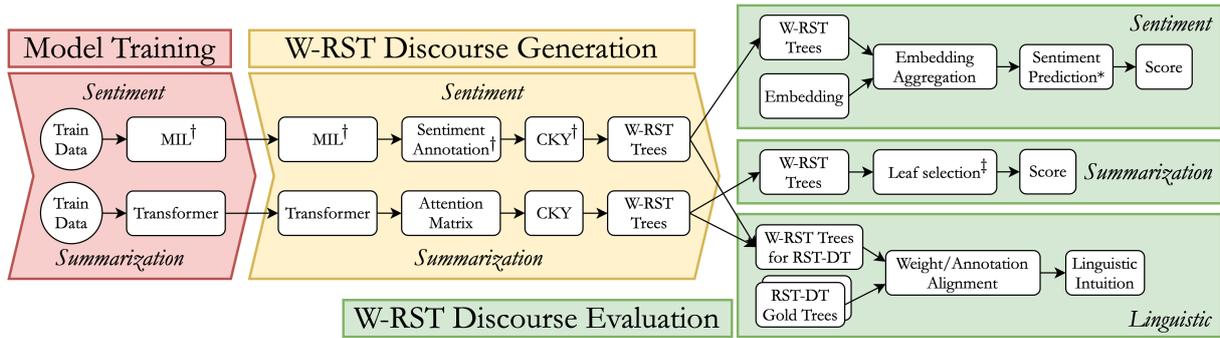


Figure 2: Three phases of our approach to generate weighted RST-style discourse trees. Left and center steps are described in section 3, right component is described in section 4. † = As in Huber and Carenini (2020b), ‡ = As in Marcu (1999), * = Sentiment prediction component is a linear combination, mapping the aggregated embedding to the sentiment output. The linear combination has been previously learned on the training portion of the dataset.

tween summarization and discourse. In particular, in Xiao et al. (2021), we demonstrate that the self-attention matrix learned during the training of a transformer-based summarizer captures valid aspects of constituency and dependency discourse trees.

To summarize, building on our previous work on creating discourse trees through distant supervision, we take a first step towards generating weighted discourse trees from the sentiment analysis and summarization tasks.

3 W-RST Treebank Generation

Given the intuition from above, we combine information from machine learning approaches with insights from linguistics, replacing the human-centered nuclearity assignment with real-valued weights obtained from the *sentiment analysis* and *summarization* tasks¹. An overview of the process to generate weighted RST-style discourse trees is shown in Figure 2, containing the training phase (left) and the W-RST discourse inference phase (center) described here. The W-RST discourse evaluation (right), is covered in section 4.

3.1 Weighted Trees from Sentiment

To generate weighted discourse trees from sentiment, we slightly modify the publicly available code² presented in Huber and Carenini (2020b) by removing the nuclearity discretization component.

An overview of our method is shown in Figure 2 (top), while a detailed view is presented in the left and center parts of Figure 3. First (on the left), we train the Multiple Instance Learning (MIL)

¹Please note that both tasks use binarized discourse trees, as commonly used in computational models of RST.

²Code available at <https://github.com/nlpat/MEGA-DT>

model proposed by Angelidis and Lapata (2018) on a corpus with document-level sentiment gold-labels, internally annotating each input-unit (in our case EDUs) with a sentiment- and attention-score. After the MIL model is trained (center), a tuple (s_i, a_i) containing a sentiment score s_i and an attention a_i is extracted for each EDU i . Based on these tuples representing leaf nodes, the CKY algorithm (Jurafsky and Martin, 2014) is applied to find the tree structure to best align with the overall document sentiment, through a bottom-up aggregation approach defined as³:

$$s_p = \frac{s_l * a_l + s_r * a_r}{a_l + a_r} \quad a_p = \frac{a_l + a_r}{2}$$

with nodes l and r as the left and right child-nodes of p respectively. The attention scores (a_l, a_r) are here interpreted as the importance weights for the respective sub-trees ($w_l = a_l / (a_l + a_r)$ and $w_r = a_r / (a_l + a_r)$), resulting in a complete, normalized and weighted discourse structure as required for W-RST. We call the discourse treebank generated with this approach *W-RST-Sent*.

3.2 Weighted Trees from Summarization

In order to derive weighted discourse trees from a summarization model we follow Xiao et al. (2021)⁴, generating weighted discourse trees from the self-attention matrices of a transformer-based summarization model. An overview of our method is shown in Figure 2 (bottom), while a detailed view is presented in the left and center parts of Figure 4.

We start by training a transformer-based extractive summarization model (left), containing three

³Equations taken from Huber and Carenini (2020b)

⁴Code available at https://github.com/Wendy-Xiao/summ_guided_disco_parser

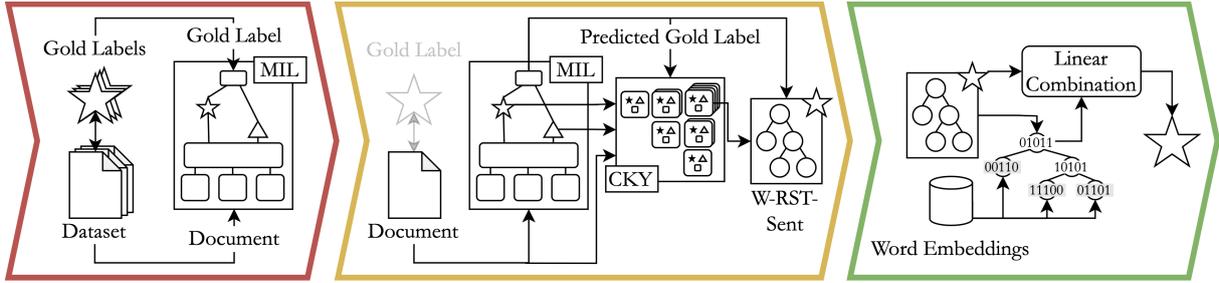


Figure 3: Three phases of our approach. Left/Center: Detailed view into the generation of weighted RST-style discourse trees using the sentiment analysis downstream task. Right: Sentiment discourse application evaluation

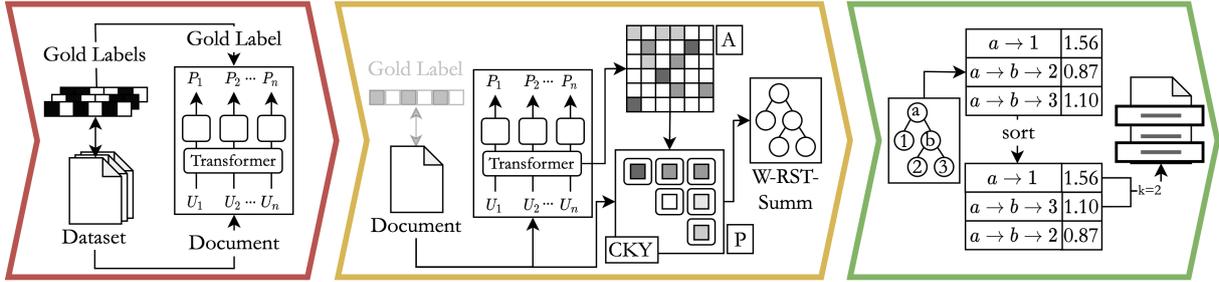


Figure 4: Three phases of our approach. Left/Center: Detailed view into the generation of weighted RST-style discourse trees using the summarization downstream task. Right: Summarization discourse application evaluation

components: (1) A pre-trained BERT EDU Encoder generating EDU embeddings, (2) a standard transformer architecture as proposed in Vaswani et al. (2017) and (3) a final classifier, mapping the outputs of the transformer to a probability score for each EDU, indicating whether the EDU should be part of the extractive summary.

With the trained transformer model, we then extract the self-attention matrix A and build a discourse tree in bottom-up fashion (as shown in the center of Figure 4). Specifically, the self-attention matrix A reflects the relationships between units in the document, where entry A_{ij} measures how much the i -th EDU relies on the j -th EDU. Given this information, we generate an unlabeled constituency tree using the CKY algorithm (Jurafsky and Martin, 2014), optimizing the overall tree score, as previously done in Xiao et al. (2021).

In terms of weight-assignment, given a sub-tree spanning EDUs i to j , split into child-constituents at EDU k , then $\max(A_{i:k,(k+1):j})$, representing the maximal attention value that any EDU in the left constituent is paying to an EDU in the right child-constituent, reflects how much the left sub-tree relies on the right sub-tree, while $\max(A_{(k+1):j,i:k})$ defines how much the right sub-tree depends on the left. We define the importance-weights of the left (w_l) and right (w_r) sub-trees as:

$$w_l = \max(A_{(k+1):j,i:k}) / (w_l + w_r)$$

$$w_r = \max(A_{i:k,(k+1):j}) / (w_l + w_r)$$

In this way, the importance scores of the two sub-trees represent a real-valued distribution. In combination with the unlabeled structure computation, we generate a weighted discourse tree for each document. We call the discourse treebank generated with the summarization downstream information *W-RST-Summ*.

4 W-RST Discourse Evaluation

To assess the potential of W-RST, we consider two evaluation scenarios (Figure 2, right): (1) Apply weighted discourse trees to the tasks of sentiment analysis and summarization and (2) analyze the weight alignment with human annotations.

4.1 Weight-based Discourse Applications

In this evaluation scenario, we address the question of whether W-RST trees can support downstream tasks better than traditional RST trees with nuclearity. Specifically, we leverage the discourse trees learned from sentiment for the sentiment analysis task itself and, similarly, rely on the discourse trees learned from summarization to benefit the summarization task.

4.1.1 Sentiment Analysis

In order to predict the sentiment of a document in *W-RST-Sent* based on its weighted discourse tree, we need to introduce an additional source of information to be aggregated according to such tree. Here, we choose word embeddings, as commonly used as an initial transformation in many models tackling the sentiment prediction task (Kim, 2014; Tai et al., 2015; Yang et al., 2016; Adhikari et al., 2019; Huber and Carenini, 2020a). To avoid introducing additional confounding factors through sophisticated tree aggregation approaches (e.g. TreeLSTMs (Tai et al., 2015)), we select a simple method, aiming to directly compare the inferred tree-structures and allowing us to better assess the performance differences originating from the weight/nuclearity attribution (see right step in Figure 3). More specifically, we start by computing the average word-embedding for each leaf node $leaf_i$ (here containing a single EDU) in the discourse tree.

$$leaf_i = \frac{1}{|leaf_i|} \sum_{j=0}^{j < |leaf_i|} Emb(word_i^j)$$

With $|leaf_i|$ as the number of words in leaf i , $Emb(\cdot)$ being the embedding lookup and $word_i^j$ representing word j within leaf i . Subsequently, we aggregate constituents, starting from the leaf nodes (with $leaf_i$ as embedding constituent c_i), according to the weights of the discourse tree. For any two sibling constituents c_l and c_r of the parent sub-tree c_p in the binary tree, we compute

$$c_p = c_l * w_l + c_r * w_r$$

with w_l and w_r as the real-valued weight-distribution extracted from the inferred discourse tree and c_p, c_l and c_r as dense encodings. We aggregate the complete document in bottom-up fashion, eventually reaching a root node embedding containing a tree-weighted average of the leaf-nodes. Given the root-node embedding representing a complete document, a simple Multilayer Perceptron (MLP) trained on the original training portion of the MIL model is used to predict the sentiment of the document.

4.1.2 Summarization

In the evaluation step of the summarization model (right of Figure 4), we use the weighted discourse tree of a document in *W-RST-Summ* to predict its extractive summary by applying an adaptation of

the unsupervised summarization method by Marcu (1999).

We choose this straightforward algorithm over more elaborate and hyper-parameter heavy approaches to avoid confounding factors, since our aim is to evaluate solely the potential of the weighted discourse trees compared to standard RST-style annotations. In the original algorithm, a summary is computed based on the nuclearity attribute by recursively computing the importance scores for all units as:

$$S_n(u, N) = \begin{cases} d_N, & u \in Prom(N) \\ S(u, C(N)) \text{ s.t.} & \\ u \in C(N) & otherwise \end{cases}$$

where $C(N)$ represents the child of N , and $Prom(N)$ is the promotion set of node N , which is defined in bottom-up fashion as follows: (1) $Prom$ of a leaf node is the leaf node itself. (2) $Prom$ of an internal node is the union of the promotion sets of its nucleus children. Furthermore, d_N represents the level of a node N , computed as the distance from the level of the lowest leaf-node. This way, units in the promotion set originating from nodes that are higher up in the discourse tree are amplified in their importance compared to those from lower levels.

As for the *W-RST-Summ* discourse trees with real-valued importance-weights, we adapt Marcu’s algorithm by replacing the promotion set with real-valued importance scores as shown here:

$$S_w(u, N) = \begin{cases} d + w_N, & N \text{ is leaf} \\ S_w(u, C(N)) + w_N, & \\ u \in C(N) & otherwise \end{cases}$$

Once S_n or S_w are computed, the top-k units of the highest promotion set or with the highest importance scores respectively are selected into the final summary.

4.1.3 Nuclearity-attributed Baselines

To test whether the W-RST trees are effectively predicting the downstream tasks, we need to generate traditional RST trees with nuclearity to compare against. However, moving from weighted discourse trees to coarse nuclearity requires the introduction of a threshold. More specifically, while “nucleus-satellite” and “satellite-nucleus” assignments can be naturally generated depending on the distinct weights, in order to assign the third “nucleus-nucleus” class, frequently appearing in

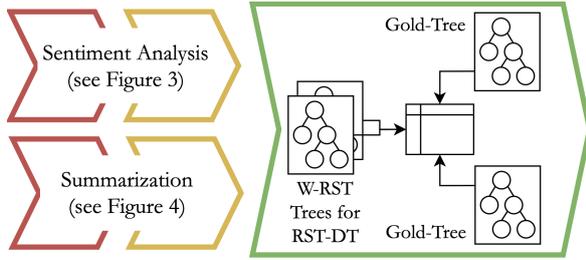


Figure 5: Three phases of our approach. Left: Generation of *W-RST-Sent/Summ* discourse trees. Right: Linguistic evaluation

RST-style treebanks, we need to specify how close two weights have to be for such configuration to apply. Formally, we set a threshold t as follows:

$$\begin{aligned} \text{If: } & |w_l - w_r| < t \quad \rightarrow \quad \text{nucleus-nucleus} \\ \text{Else: If: } & w_l > w_r \quad \rightarrow \quad \text{nucleus-satellite} \\ & \text{If: } w_l \leq w_r \quad \rightarrow \quad \text{satellite-nucleus} \end{aligned}$$

This way, RST-style treebanks with nuclearity attributions can be generated from *W-RST-Sent* and *W-RST-Summ* and used for the sentiment analysis and summarization downstream tasks. For the nuclearity-attributed baseline of the sentiment task, we use a similar approach as for the W-RST evaluation procedure, but assign two distinct weights w_n and w_s to the nucleus and satellite child respectively. Since it is not clear how much more important a nucleus node is compared to a satellite using the traditional RST notation, we define the two weights based on the threshold t as:

$$w_n = 1 - (1 - 2t)/4 \quad w_s = (1 - 2t)/4$$

The intuition behind this formulation is that for a high threshold t (e.g. 0.8), the nuclearity needs to be very prominent (the difference between the normalized weights needs to exceed 0.8), making the nucleus clearly more important than the satellite, while for a small threshold (e.g. 0.1), even relatively balanced weights (for example $w_l = 0.56, w_r = 0.44$) will be assigned as “nucleus-satellite”, resulting in the potential difference in importance of the siblings to be less eminent.

For the nuclearity-attributed baseline for summarization, we directly apply the original algorithm by Marcu (1999) as described in section 4.1.2. However, when using the promotion set to determine which EDUs are added to the summarization, potential ties can occur. Since the discourse tree does not provide any information on how to prioritize those, we randomly select units from the candidates, whenever there is a tie. This avoids exploit-

ing any positional bias in the data (e.g. the lead bias), which would confound the results.

4.2 Weight Alignment with Human Annotation

As for our second W-RST discourse evaluation task, we investigate if the real-valued importance-weights align with human annotations. To be able to explore this scenario, we generate weighted tree annotations for an existing discourse treebank (RST-DT (Carlson et al., 2002)). In this evaluation task we verify if: (1) The nucleus in a gold-annotation generally receives more weight than a satellite (i.e. if importance-weights generally favour nuclei over satellites) and, similarly, if nucleus-nucleus relations receive more balanced weights. (2) In accordance with Figure 1, we further explore how well the weights capture the extend to which a relation is dominated by the nucleus. Here, our intuition is that for inconsistent human nuclearity annotations the spread should generally be lower than for consistent annotations, assuming that human misalignment in the discourse annotation indicates ambivalence on the importance of sub-trees.

To test for these two properties, we use discourse documents individually annotated by two human annotators and analyze each sub-tree within the doubly-annotated documents with consistent inter-annotator structure assessment for their nuclearity assignment. For each of the 6 possible inter-annotator nuclearity assessments, consisting of 3 consistent annotation classes (namely N-N/N-N, N-S/N-S and S-N/S-N) and 3 inconsistent annotation classes (namely N-N/N-S, N-N/S-N and N-S/S-N)⁵, we explore the respective weight distribution of the document annotated with the two W-RST tasks – sentiment analysis and summarization (see Figure 5). We compute an average spread s_c for each of the 6 inter-annotator nuclearity assessments classes c as:

$$s_c = \left(\sum_{j=0}^{j < |c|} w_l^j - w_r^j \right) / |c|$$

With w_l^j and w_r^j as the weights of the left and right child node of sub-tree j in class c , respectively.

⁵We don’t take the order of annotators into consideration, mapping N-N/N-S and N-S/N-N both onto N-N/N-S.

5 Experiments

5.1 Experimental Setup

Sentiment Analysis: We follow our previous approach in Huber and Carenini (2020b) for the model training and W-RST discourse inference steps (left and center in Figure 3) using the adapted MILNet model from Angelidis and Lapata (2018) trained with a batch-size of 200 and 100 neurons in a single layer bi-directional GRU with 20% dropout for 25 epochs. Next, discourse trees are generated using the best-performing heuristic CKY method with the stochastic exploration-exploitation trade-off from Huber and Carenini (2020b) (beam size 10, linear decreasing τ). As word-embeddings in the W-RST discourse evaluation (right in Figure 3), we use GloVe embeddings (Pennington et al., 2014), which previous work (Tai et al., 2015; Huber and Carenini, 2020a) indicates to be suitable for aggregation in discourse processing. For training and evaluation of the sentiment analysis task, we use the 5-class Yelp’13 review dataset (Tang et al., 2015). To compare our approach against the traditional RST approach with nuclearity, we explore the impact of 11 distinct thresholds for the baseline described in §4.1.3, ranging from 0 to 1 in 0.1 intervals.

Summarization: To be consistent with RST, our summarizer extracts EDUs instead of sentences from a given document. The model is trained on the EDU-segmented CNNDM dataset containing EDU-level Oracle labels published by Xu et al. (2020). We further use a pre-trained BERT-base (“uncased”) model to generate the embeddings of EDUs. The transformer used is the standard model with 6 layers and 8 heads in each layer ($d = 512$). We train the extractive summarizer on the training set of the CNNDM corpus (Nallapati et al., 2016) and pick the best attention head using the RST-DT dataset (Carlson et al., 2002) as the development set. We test the trees by running the summarization algorithm in Marcu (1999) on the test set of the CNNDM dataset, and select the top-6 EDUs based on the importance score to form a summary in natural order. Regarding the baseline model using thresholds, we apply the same 11 thresholds as for the sentiment analysis task.

Weight Alignment with Human Annotation:

As discussed in §4.2, this evaluation requires two parallel human generated discourse trees for every document. Luckily, in the RST-DT corpus pub-

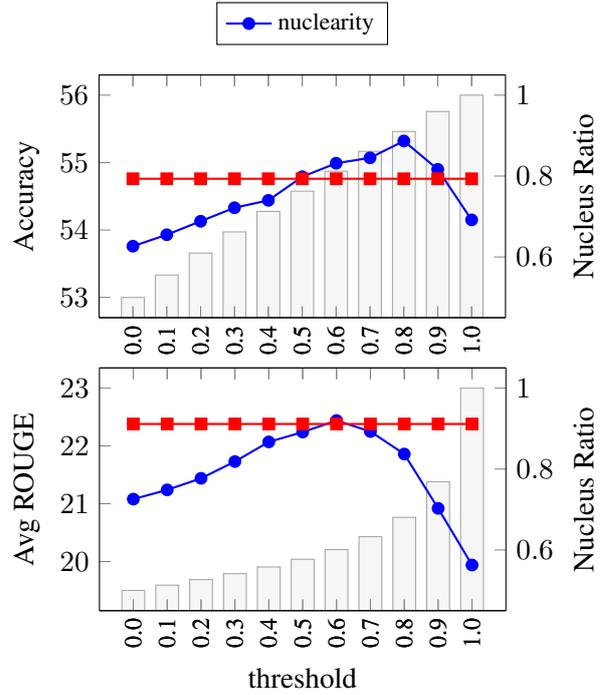


Figure 6: Top: Sentiment Analysis accuracy of the W-RST model compared to the standard RST framework with different thresholds. Bottom: Average ROUGE score (ROUGE-1, -2 and -L) of the W-RST summarization model compared to different thresholds. Full numerical results are shown in Appendix A.

	N-N	N-S	S-N
N-N	273	99	41
N-S	-	694	75
S-N	-	-	172

Table 1: Statistics on consistently and inconsistently annotated samples of the 1,354 structure-aligned subtrees generated by two distinct human annotators.

lished by Carlson et al. (2002), 53 of the 385 documents annotated with full RST-style discourse trees are doubly tagged by a second linguist. We use the 53 documents containing 1,354 consistent structure annotations between the two analysts to evaluate the linguistic alignment of our generated W-RST documents with human discourse interpretations. Out of the 1,354 structure-aligned subtrees, in 1,139 cases both annotators agreed on the nuclearity attribute, while 215 times a nuclearity mismatch appeared, as shown in detail in Table 1.

5.2 Results and Analysis

The results of the experiments on the **discourse applications** for sentiment analysis and summarization are shown in Figure 6. The results for

Sent	N-N	N-S	S-N
N-N	-0.228 (106)	-0.238 (33)	-0.240 (19)
N-S	-	-0.038 (325)	-0.044 (22)
S-N	-	-	-0.278 (115)

Summ	N-N	N-S	S-N
N-N	0.572 (136)	0.604 (42)	0.506 (25)
N-S	-	0.713 (418)	0.518 (36)
S-N	-	-	0.616 (134)

Table 2: Confusion Matrices based on human annotation showing the absolute weight-spread using the Sentiment (top) and Summarization (bottom) tasks on 620 and 791 sub-trees aligned with the human structure prediction, respectively. Cell upper value: Absolute weight spread for the respective combination of human-annotated nuclearities. Lower value (in brackets): Support for this configuration.

sentiment analysis (top) and summarization (bottom) thereby show a similar trend: With an increasing threshold and therefore a larger number of N-N relations (shown as grey bars in the Figure), the standard RST baseline (blue line) consistently improves for the respective performance measure of both tasks. However, reaching the best performance at a threshold of 0.8 for sentiment analysis and 0.6 for summarization, the performance starts to deteriorate. This general trend seems reasonable, given that N-N relations represent a rather frequent nuclearity connection, however classifying every connection as N-N leads to a severe loss of information. Furthermore, the performance suggests that while the N-N class is important in both cases, the optimal threshold varies depending on the task and potentially also the corpus used, making further task-specific fine-tuning steps mandatory. The weighted discourse trees following our W-RST approach, on the other hand, do not require the definition of a threshold, resulting in a single, promising performance (red line) for both tasks in Figure 6. For comparison, we apply the generated trees of a standard RST-style discourse parser (here the Two-Stage parser by Wang et al. (2017)) trained on the RST-DT dataset (Carlson et al., 2002) on both downstream tasks. The fully-supervised parser reaches an accuracy of 44.77% for sentiment analysis and an average ROUGE score of 26.28 for summarization. While the average ROUGE score

Sent	N-N	N-S	S-N
N-N	\emptyset -0.36	\emptyset -0.43	\emptyset -0.45
N-S	-	\emptyset +1.00	\emptyset +0.96
S-N	-	-	\emptyset -0.72

Summ	N-N	N-S	S-N
N-N	\emptyset -0.13	\emptyset +0.13	\emptyset -0.66
N-S	-	\emptyset +1.00	\emptyset -0.56
S-N	-	-	\emptyset +0.22

Table 3: Confusion Matrices based on human annotation showing the weight-spread relative to the task-average for Sentiment (top) and Summarization (bottom), aligned with the human structure prediction, respectively. Cell value: Relative weight spread as the divergence from the average spread across all cells in Table 2. Color: Positive/Negative divergence, \emptyset = Average value of absolute scores.

of the fully-supervised parser is above the performance of our W-RST results for the summarization task, the accuracy on the sentiment analysis task is well below our approach. We believe that these results are a direct indication of the problematic domain adaptation of fully supervised discourse parsers, where the application on a similar domain (Wall Street Journal articles vs. CNN-Daily Mail articles) leads to superior performances compared to our distantly supervised method, however, with larger domain shifts (Wall Street Journal articles vs. Yelp customer reviews), the performance drops significantly, allowing our distantly supervised model to outperform the supervised discourse trees for the downstream task. Arguably, this indicates that although our weighted approach is still not competitive with fully-supervised models in the same domain, it is the most promising solution available for cross-domain discourse parsing.

With respect to exploring the **weight alignment with human annotations**, we show a set of confusion matrices based on human annotation for each W-RST discourse generation task on the absolute and relative weight-spread in Tables 2 and 3 respectively. The results for the sentiment analysis task are shown on the top of both tables, while the performance for the summarization task is shown at the bottom. For instance, the top right cell of the upper confusion matrix in Table 2 shows that for 19 sub-trees in the doubly annotated subset of RST-DT one of the annotators labelled the sub-tree with a nucleus-nucleus nuclearity attribution, while the second annotator identified it as satellite-

nucleus. The average weight spread (see §4.2) for those 19 sub-trees is -0.24 . Regarding Table 3, we subtract the average spread across Table 2 defined as $\varnothing = \sum_{c_i \in C} (c_i) / |C|$ (with $C = \{c_1, c_2, \dots, c_6\}$ containing the cell values in the upper triangle matrix) from each cell value c_i and normalize by $max = \max_{c_i \in C} (|c_i - \varnothing|)$, with $\varnothing = -0.177$ and $max = 0.1396$ across the top table. Accordingly, we transform the -0.24 in the top right cell into $(-0.24 - avg) / max = -0.45$.

Moving to the analysis of the results, we find the following trends in this experiment: (1) As presented in Table 2, the sentiment analysis task tends to strongly over-predict S-N (i.e., $w_l \ll w_r$), leading to negative spreads in all cells. In contrast, the summarization task is heavily skewed towards N-S assignments (i.e., $w_l \gg w_r$), leading to exclusively positive spreads. We believe both trends are consistent with the intrinsic properties of the tasks, given that the general structure of reviews tends to become more important towards the end of a review (leading to increased S-N assignments), while for summarization, the lead bias potentially produces the overall strong nucleus-satellite trend. (2) To investigate the relative weight spreads for different human annotations (i.e., between cells) beyond the trends shown in Table 2, we normalize values within a table by subtracting the average and scaling between $[-1, 1]$. As a result, Table 3 shows the relative weight spread for different human annotations. Apart from the general trends described in Table 2, the consistently annotated samples of the two linguists (along the diagonal of the confusion matrices) align reasonably. The most positive weight spread is consistently found in the agreed-upon nucleus-satellite case, while the nucleus-nucleus annotation has, as expected, the lowest divergence (i.e., closest to zero) along the diagonal in Table 3. (3) Regarding the inconsistently annotated samples (shown in the triangle matrix above the diagonal) it becomes clear that in the sentiment analysis model the values for the N-N/N-S and N-N/S-N annotated samples (top row in Table 3) are relatively close to the average value. This indicates that, similar to the nucleus-nucleus case, the weights are also ambivalent, with the N-N/N-S value (top center) slightly larger than the value for N-N/S-N (top right). The N-S/S-N case for the sentiment analysis model is less aligned with our intuition, showing a strongly negative weight-spread (i.e. $w_l \ll w_r$) where we would have

expected a more ambivalent result with $w_l \approx w_r$ (however, aligned with the overall trend shown in Table 2). For summarization, we see a very similar trend with the values for N-N/N-S and N-N/S-N annotated samples. Again, both values are close to the average, with the N-N/N-S cell showing a more positive spread than N-N/S-N. However for summarization, the consistent satellite-nucleus annotation (bottom right cell) seems misaligned with the rest of the table, following instead the general trend for summarization described in Table 2. All in all, the results suggest that the values in most cells are well aligned with what we would expect regarding the relative spread. Interestingly, human uncertainty appears to be reasonably captured in the weights, which seem to contain more fine grained information about the relative importance of sibling sub-trees.

6 Conclusion and Future Work

We propose W-RST as a new discourse framework, where the binary nuclearity assessment postulated by RST is replaced with more expressive weights, that can be automatically generated from auxiliary tasks. A series of experiments indicate that W-RST is beneficial to the two key NLP downstream tasks of sentiment analysis and summarization. Further, we show that W-RST trees interestingly align with the uncertainty of human annotations.

For the future, we plan to develop a neural discourse parser that learns to predict importance weights instead of nuclearity attributions when trained on large W-RST treebanks. More longer term, we want to explore other aspects of RST that can be refined in light of empirical results, plan to integrate our results into state-of-the-art sentiment analysis and summarization approaches (e.g. Xu et al. (2020)) and generate parallel W-RST structures in a multi-task manner to improve the generality of the discourse trees.

Acknowledgments

We thank the anonymous reviewers for their insightful comments. This research was supported by the Language & Speech Innovation Lab of Cloud BU, Huawei Technologies Co., Ltd and the Natural Sciences and Engineering Research Council of Canada (NSERC). Nous remercions le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) de son soutien.

References

- Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Rethinking complex neural network architectures for document classification. 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 4046–4051.
- Stefanos Angelidis and Mirella Lapata. 2018. Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association for Computational Linguistics*, 6:17–31.
- Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from RST discourse parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2212–2218.
- Lynn Carlson, Mary Ellen Okurowski, and Daniel Marcu. 2002. *RST discourse treebank*. Linguistic Data Consortium, University of Pennsylvania.
- 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.
- Albert Gatt and Emiel Krahrmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.
- Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T Ng, and Bitá Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1602–1613.
- Luke Gessler, Yang Janet Liu, and Amir Zeldes. 2019. A discourse signal annotation system for rst trees. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 56–61.
- Grigorií Guz and Giuseppe Carenini. 2020. Towards domain-independent text structuring trainable on large discourse treebanks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3141–3152.
- Alexander Hogenboom, Flavius Frasinca, Franciska De Jong, and Uzay Kaymak. 2015. Using rhetorical structure in sentiment analysis. *Commun. ACM*, 58(7):69–77.
- Patrick Huber and Giuseppe Carenini. 2020a. From sentiment annotations to sentiment prediction through discourse augmentation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 185–197.
- Patrick Huber and Giuseppe Carenini. 2020b. MEGA RST discourse treebanks with structure and nuclearity from scalable distant sentiment supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7442–7457.
- 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)*, volume 1, pages 13–24.
- Yangfeng Ji and Noah A Smith. 2017. Neural discourse structure for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1005.
- Shafiq Joty, Giuseppe Carenini, and Raymond T Ng. 2015. CODRA: A novel discriminative framework for rhetorical analysis. *Computational Linguistics*, 41(3).
- Dan Jurafsky and James H Martin. 2014. *Speech and language processing*, volume 3. Pearson London.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.
- 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.
- Yang Liu and Amir Zeldes. 2019. Discourse relations and signaling information: Anchoring discourse signals in rst-dt. *Proceedings of the Society for Computation in Linguistics*, 2(1):314–317.
- William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.
- Daniel Marcu. 1999. Discourse trees are good indicators of importance in text. *Advances in automatic text summarization*, 293:123–136.
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Guğlçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290. Association for Computational Linguistics.
- Bitá Nejat, Giuseppe Carenini, and Raymond Ng. 2017. Exploring joint neural model for sentence level discourse parsing and sentiment analysis. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 289–298.

- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltasakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The penn discourse treebank 2.0. *LREC*.
- Rajen Subba and Barbara Di Eugenio. 2009. An effective discourse parser that uses rich linguistic information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 566–574. Association for Computational Linguistics.
- Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566.
- Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432.
- Mark Tarrance. 2015. Understanding planning in text production. *Handbook of writing research*, pages 1682–1690.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.
- 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.
- Wen Xiao, Patrick Huber, and Giuseppe Carenini. 2020. Do we really need that many parameters in transformer for extractive summarization? discourse can help! In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pages 124–134.
- Wen Xiao, Patrick Huber, and Giuseppe Carenini. 2021. Predicting discourse trees from transformer-based neural summarizers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4139–4152, Online. Association for Computational Linguistics.
- Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031. Association for Computational Linguistics.
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.
- 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.
- Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

A Numeric Results

The numeric results of our W-RST approach for the sentiment analysis and summarization downstream tasks presented in Figure 6 are shown in Table 4 below, along with the threshold-based approach, as well as the supervised parser.

Approach	Sentiment Accuracy	Summarization		
		R-1	R-2	R-L
Nuclearity with Threshold				
t = 0.0	53.76	28.22	8.58	26.45
t = 0.1	53.93	28.41	8.69	26.61
t = 0.2	54.13	28.64	8.85	26.83
t = 0.3	54.33	28.96	9.08	27.14
t = 0.4	54.44	29.36	9.34	27.51
t = 0.5	54.79	29.55	9.50	27.68
t = 0.6	54.99	29.78	9.65	27.90
t = 0.7	55.07	29.57	9.45	27.74
t = 0.8	55.32	29.18	9.08	27.32
t = 0.9	54.90	28.11	8.29	26.35
t = 1.0	54.15	26.94	7.60	25.27
Our Weighted RST Framework				
weighted	54.76	29.70	9.58	27.85
Supervised Training on RST-DT				
supervised	44.77	34.20	12.77	32.09

Table 4: Results of the W-RST approach compared to threshold-based nuclearity assignments and supervised training on RST-DT.