What's Hard in RST Parsing? Predictive Models for Error Analysis

Anonymous ACL submission

Abstract

Despite recent advances in Natural Language Processing (NLP), hierarchical discourse parsing in the framework of Rhetorical Structure Theory remains challenging, and our understanding of the reasons for this are as yet limited. In this paper we examine and model some of the factors associated with parsing difficulties in previous work: the existence of implicit discourse relations, challenges in identifying long-distance relations, out-of-vocabulary items, and more. In order to assess the relative importance of these variables, we also release two annotated English test-sets with explicit correct and distracting discourse markers associated with gold standard RST relations. Our results show that as in shallow discourse parsing, the explicit/implicit distinction plays a role, but that long-distance dependencies are the main challenge, while lack of lexical overlap is less of a problem, at least for in-domain parsing. Our final model is able to predict where errors will occur with an accuracy of 76.28% for the bottom-up parser and 76.61% for the Our final model is able to predict where errors will occur with an accuracy of 76.28% for the bottom-up parser and 76.61% for the top-down parser.

1 Introduction

Powered by pretrained language models, recent advancements in NLP have led to rising scores on a myriad of language understanding tasks, especially at the sentence level. However, at the discourse level, where analyses require reasoning over multiple sentences, progress has been slower, with generalization to unseen domains remaining a persistent problem for tasks such as coreference resolution (Zhu et al., 2021) and entity linking (Lin and Zeldes, 2021).

One task which remains particularly challenging is hierarchical discourse parsing, which aims to identify the connections between propositions in a text or conversation, classify their functions, and

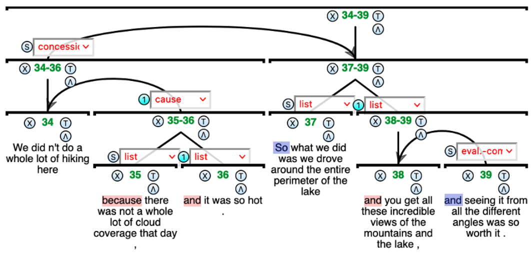

Figure 1: An RST analysis of a *vlog* excerpt. Tokens highlighted in red are discourse markers associated with relations in the tree, while tokens highlighted in blue are distractors, with no corresponding relation.

form a recursive tree structure, which indicates the locally most prominent elementary discourse unit (EDU) in each tree or subtree. For example, Figure 1 shows a tree in the most popular hierarchical discourse formalism, Rhetorical Structure Theory (RST, Mann and Thompson 1988), in which the list of units 37–38 is the most prominent (being pointed to by other units directly or indirectly), and discourse relation labels such as CAUSE are identified using edge labels.

There is by now substantial evidence showing that even for a high resource language like English, state-of-the-art (SOTA) neural discourse parsers, whether employing a top-down or a bottom-up architecture, do not perform well across domains (Atwell et al., 2021, 2022; Yu et al., 2022), with some crucial tasks, such as predicting the most prominent Central Discourse Unit (CDU) of each document, performing at just 50% (Liu and Zeldes, 2023). At the same time, we do not have a good understanding of what exactly prevents good performance—is it the fact that some relations are **well-marked** (for example, most CONTINGENCY relations are marked by the discourse marker (DM) *if*, but most EVALUATION relations lack a common marker)? Conversely, is the **presence of distracting markers** not associated with the correct relation (e.g. an additional temporal marker such as *then* inside a unit with a non-temporal function)?

Alternatively, is it the difficulty in identifying high-level relations, between groups of multiple sentences or paragraphs, compared to less tricky intra-sentential relations between clauses? Or is it just the prevalence of out-of-vocabulary (OOV) items in test data?

In this paper, we would like to systematically evaluate the role of these and other factors contributing to errors in English RST discourse parsing. Our contributions include:

- Annotation and evaluation of the `dev`/`test` sets of the English RST-DT (Carlson et al., 2003) and GUM datasets (Zeldes, 2017), for explicit relation markers, as well as distracting markers not signaling the correct relation;
- Parsing experiments with two different SOTA architectures to examine where degradation happens;
- Development and analysis of multifactorial models predicting where errors will occur and ranking importance for different variables;
- Qualitative and quantitative error analysis.

Our results reveal that while explicit markers and distractors do play a role, the most significant predictor of difficulty is inter-sentential status and the specific relation involved. At the same time, our error analysis indicates that distractors often correspond to true discourse relations which are not included in the gold-standard tree, but may be included in alternative trees produced by other annotators. In addition, we find that OOV rate plays only a minor role, that architecture choice is presently not very important, and that genre continues to matter even when all other factors are known. All code and data are available at `anonymized URL`.

2 Related Work

2.1 Discourse Structure in Discourse Parsing

Discourse parsing is the task of identifying the coherence relations that hold between different parts of a text. Regardless of discourse frameworks or formalisms, identifying intra-sentential, inter-sentential, or inter-paragraph discourse relations may pose different levels of difficulty to parsers due to their various characteristics and levels of explicitness (e.g. Zhao and Webber 2021; Dai and Huang 2018; Muller et al. 2012). Intuitively, this becomes increasingly important for discourse parsing in a hierarchical framework such as RST, where long-distance relations are more frequent.

Researchers have therefore been considering ways of dealing with long-distance relations for nearly twenty years, starting with the structure-informed model proposed by Sporleder and Lascarides (2004) to tackle local and global discourse structures such as paragraphs. Other multi-stage parsing models, for example, as developed by Joty et al. (2013, 2015), have taken into account the distribution and associated features of intra-sentential and inter-sentential relations, achieving competitive results for English document-level parsing.

Later models expanded on these approaches by incorporating paragraph information to better capture high-level document structures. For instance, Liu and Lapata (2017) proposed a neural model leveraging global context, enabling it to capture long-distance dependencies and achieving SOTA performance. Active research continues on developing multi-stage parsing algorithms aiming at capitalizing on structural information at the sentence or paragraph-levels (Wang et al., 2017; Kobayashi et al., 2020; Nishida and Nakayama, 2020).

2.2 Explicit and Implicit Relations in RST

Unlike in hierarchical RST parsing, work on shallow discourse parsing in the framework of the Penn Discourse Treebank (PDTB, Prasad et al. 2014), in which relations apply between spans of text without forming a tree, has long distinguished explicitly and implicitly marked discourse relations. Explicit relations are signaled by connectives such as ‘but’ or ‘on the other hand’, while implicit ones lack such marking. It is well-established that shallow parsing of explicit discourse relations is substantially easier due to the availability of connective signals, which, although not unambiguous, narrow down likely senses for relations. For example, the best systems from Knaebel (2021) achieved an F1 score of 62.75 on explicit relations and an F1 score of 40.71 on implicit relations for Section 23 of WSJ using PDTB v2 (Prasad et al., 2008).

RST datasets used in hierarchical discourse parsing do not make such a distinction, in part because RST trees include very high-level relations between entire sections of documents, which are less likely to be marked by such items. As a result, such a distinction is not available, meaning that we are in the dark regarding the prevalence and importance of such markers for RST parsing.

We are aware of two prior works analyzing connectives for RST data: the RST Signalling Corpus

(RST-SC, Das et al. 2019) analyzes each relation in the English RST-DT dataset, indicating which relations were signaled by a DM (DMs roughly include the same items as PDTB connectives; see Webber et al. (2019) and Das and Taboada (2014) for complete inventories of markers). However, the data is limited to newswire material and does not provide an alignment of analyses to actual tokens, limiting the possibilities for model building (i.e. we only know whether a DM was present somewhere, not which token in the text it was or in which exact EDU it appeared). It also does not indicate whether DMs were present which ***did not*** signal the relation in the tree (i.e. distractors). Although previous efforts targeted DM tokens in RST-DT (Liu and Zeldes, 2019) and the study of signals beyond newswire texts (Liu, 2019), no previous study has examined the role of DMs in RST parsing.

Stede and Neumann (2014) enriched an RST corpus of German with token-aligned explicit connectives and the relations they signal, making it possible to investigate their positions and potentially the presence of distracting connectives. However, the annotations were not mapped to the RST relations in the corpus, making exact inferences again tricky, and the size of the corpus (32K tokens) precludes training high quality neural models. This corpus too is limited to the newspaper domain, which also motivates us to annotate genre-rich data, described in the next section.

Finally we note that data in other frameworks, including not only PDTB but also SDRT (Segmented Discourse Representation Theory, Asher and Lascarides 2003), contains multiple concurrent discourse relations, providing information about the presence of competing or distracting relations. However, SDRT data does not include connective annotations, and apart from the coverage of RST-SC of overlapping data from the Wall Street Journal (WSJ) in PDTB, there is no way to extract a mapping between connectives and RST relations in any existing dataset (for attempts at aligning PDTB and RST-DT, see Demberg et al. 2019).

In this paper, we therefore begin by creating hand-annotated data associating exact DM tokens with RST-style relations, or indicating their status as distractors, not associated with any relation in the gold tree. These latter DMs are especially interesting, since they could indicate that some parser errors are not exactly errors, instead corresponding to concurrent relations not present in the gold trees.

3 Data

To examine the role of explicit vs. implicit relations in parsing errors, we first need to know which relations were explicitly signaled. To that end, we use PDTB's methodology to define explicit connectives,[1] and annotate data from the two largest RST corpora for English, covering the `test` set of RST-DT[2] (Carlson et al., 2003) and the `test` and `dev` sets of GUM (Zeldes, 2017) , with 1) **discourse markers** (including 'distractor' DMs) and 2) **associated relations**, thereby attaching DMs to each relation they signal, or no relation. Table 1 gives an overview of the data.

	RST-DT	**GUM v9**
# of docs	385	213
train/dev/test	*347 / – / 38*	*165 / 24 / 24*
# of toks	203,352	203,780
# of EDUs	21,789	26,310
# of genres	1	12
# of relation labels	78	32
# of relation classes	17	15
# of relation instances	18,630	23,451

Table 1: Overview of the Largest English RST Corpora.

Inter-Annotator Agreement To assess the reliability and quality of the human annotations, we conduct an inter-annotator agreement study on the `test` set of RST-DT and report average mutual F1 scores. The use of RST-DT can also facilitate some comparisons between the PDTB and RST frameworks as a number of documents from the WSJ section of the Penn Treebank (Marcus et al., 1993) were annotated in both PDTB v3 and RST-DT. In total, we double-annotated 38 documents, divided to overlap among three annotators. For DMs, the average F1 score was 95.2, and for associated relation, the average F1 score given a DM was 96.7. These scores indicate a high agreement between annotators for both tasks.

Automatic Parses In order to study of parsing errors from different architectures, we select two SOTA-performing parsers to obtain automatic parses: a BOTTOM-UP one from Guz and Carenini (2020), using their best `SpanBERT-NoCoref` setting, and a TOP-DOWN one from Liu et al. (2021). Following recommendations by Morey et al. (2017), we use the more stringent original

[1]RST papers often use the term DM without clear inventories; from this point on we will use 'DM' for brevity, but strictly adhere to the PDTB English inventory.

[2]RST-DT has no established separate `dev` set.

Parseval metric on binary trees. Table 2 shows reproduced 5-run average scores on both `test` sets.

corpora	GUM v9			RST-DT		
metrics	S	N	R	S	N	R
BOTTOM-UP Guz and Carenini (2020)	70.4	57.7	49.9	76.5	65.9	54.8
TOP-DOWN Liu et al. (2021)	71.9	58.9	51.7	76.5	65.8	54.8

Table 2: Parsing Performance on GUM v9 and RST-DT `test` with Gold Segmentation (5 run average).

It is clear that scores of both architectures are neck and neck, which raises questions on whether, beyond numeric scores, they find similar or different data difficult.

4 Analysis

Strictly speaking, the types of errors that top-down and bottom-up parsers make are not identical: while bottom-up, and in particular shift-reduce parsers see analyzed preceding discourse units, grouped in a stack, and remaining discourse units in an upcoming queue, top-down parsers analyze a domain of ungrouped tokens to be split and determine the optimal split point and label for each decision. Because we want to analyze what promotes errors both across and for each architecture, we adopt an output-centric view, analyzing EDUs at which parsers do and do not make errors based on their properties in the completed gold vs. predicted tree. At the same time, we do not want our results to be swayed by coincidental variations in neural models, which can have far-reaching consequences due to cascading errors. Instead, we train five models in each architecture: if only one model fails to predict a relation, it may not be very hard, while 4–5 errors would be indicative of genuinely hard relations.

Additionally, since models ultimately confronts different inputs as a result of such cascaded decisions, we will use a dependency representation of both the gold and predicted RST trees, following the dependency conversion as defined by Li et al. (2014). Although RST uses constituent discourse trees, focusing on each EDU and its dependencies will make it possible to make meaningful comparisons across models, and to intuitively understand how challenging EDUs are at any point in each document, regardless of whether or not they head large constituent structures. In Section 4.2 we will also incorporate the spanned domain of each head EDU's constituent block as an additional feature to assess the role of block size in predicting errors.

4.1 Explicit vs. Implicit Relations

Table 3 shows the distribution of explicit or unmarked relations across the genres in the dev/test sets of GUM v9 and in comparison to RST-DT's test set, for each relation class and overall. The results for RST-DT are consistent with previous work, with 17.0% of test data relations being marked, similarly to the 18.2% identified by Das and Taboada (2017) for the entire corpus (but not anchored to specific tokens). An examination of distributions by genre in GUM reveals some differences, highlighted in Table 3, with ***vlog*** exhibiting the most explicit relations, and ***conversation*** the fewest, raising the possibility that it may be more challenging for parsers. And in fact, Liu and Zeldes (2023) point to *conversation* as the worst-performing genre at all metric levels using an older version of the corpus (v8), which had less *conversation* data compared to GUM v9.

	# of explicit	explicit prop.	# of implicit	implicit prop.	# of distractor	distractor prop.
GUM v9	1198	21.7%	4332	78.3%	174	3.1%
RST-DT	398	17.0%	1948	83.0%	81	3.5%
academic	73	16.1%	380	83.9%	13	2.9%
bio	66	18.4%	292	81.6%	11	3.1%
conversation	100	12.9%	674	**87.1%**	23	3.0%
fiction	116	23.7%	374	76.3%	15	3.1%
interview	80	20.2%	317	79.8%	8	2.0%
news	73	18.1%	331	81.9%	7	1.7%
reddit	147	28.3%	373	71.7%	20	3.8%
speech	84	19.1%	356	80.9%	9	2.0%
textbook	95	21.3%	352	78.7%	9	2.0%
vlog	180	**35.8%**	323	64.2%	38	**7.6%**
voyage	69	22.4%	239	77.6%	9	2.9%
whow	115	26.4%	321	73.6%	12	2.8%
mean	99.8	21.9%	361	78.1%	14.5	3.1%

Table 3: Distribution of Explicit and Implicit Relations as well as EDUs with Distracting DMs in RST-DT `test` and `dev+test` of GUM v9.

Looking at the presence of 'distractor' connectives, which are not associated with one of the gold relations in the tree, we see that ***vlog*** is the most prone to such cases, again raising the question of whether these may pose a problem for parsers, which may identify **a possibly correct relation that is not prioritized by the gold tree**. This situation appears to be infrequent in the WSJ data from RST-DT, which has only 81 such cases (3.5%). Taking a closer look at the types of distractors across genres in GUM, we see that the most frequent types are 'and', 'but', and 'so', which are highly ambiguous and common in conversational data such as the *vlog* and *conversation* genres.

Regarding the most and least explicitly signaled relation classes in GUM v9, Table 4 reveals that CONTINGENCY is the most explicitly marked class due to the use of the DM 'if', and that the least explicitly signaled classes are ATTRIBUTION and ORGANIZATION. The former is almost always signaled by speech verbs and the latter mostly by document layout and graphical features in written texts, or by back-channeling in conversation data. It is also worth noting that instances of EVALUATION, RESTATEMENT, and TOPIC (used predominantly for question-answer pairs) are mostly ***not*** signaled by a discourse marker.

relation class	**# of explicit**	**explicit prop.**	**# of implicit**	**implicit prop.**
ROOT	0	0.0%	48	100.0%
ADVERSATIVE	222	55.5%	178	44.5%
ATTRIBUTION	0	0.0%	292	**100.0%**
CAUSAL	131	53.5%	114	46.5%
CONTEXT	143	31.8%	306	68.2%
CONTINGENCY	99	**91.7%**	9	8.3%
ELABORATION	64	5.8%	1049	94.2%
EVALUATION	4	1.7%	231	98.3%
EXPLANATION	44	12.5%	308	87.5%
JOINT	409	37.2%	689	62.8%
MODE	52	45.2%	63	54.8%
ORGANIZATION	0	0.0%	331	**100.0%**
PURPOSE	21	10.7%	176	89.3%
RESTATEMENT	6	3.8%	150	96.2%
SAME-UNIT	1	0.3%	289	99.7%
TOPIC	2	2.0%	99	98.0%

Table 4: Distribution of Explicit and Implicit Relations across Relation Classes in `dev+test` of GUM v9.

With these descriptive statistics in hand, we can examine each parser's performance on explicit/implicit relations, as well as on EDUs with a distracting DM in either the source or target of the relation (we must consider both ends, since many DMs can mark either a source or target such as 'but' and 'so'). Figure 2 shows the density of relations incurring between 0 and 5 attachment errors (disregarding labels) in each architecture for GUM, broken down by whether a DM marks the relation (top) and whether a distracting DM is present (bottom). The figure reveals several important facts: firstly, DMs are unsurprisingly associated with fewer errors (t=−7.29, D=0.23, p<0.0001), with lack of connectives affecting top-down models slightly more severely (χ^2=3.95, ϕ=0.14, p<0.05). Secondly, lack of distractors is associated with having fewer errors (t=5.0718, D=0.37, p<0.0001), and this is more pronounced for the bottom-up architecture, but the difference between architectures is not significant here.

Although it seems obvious that explicitness will

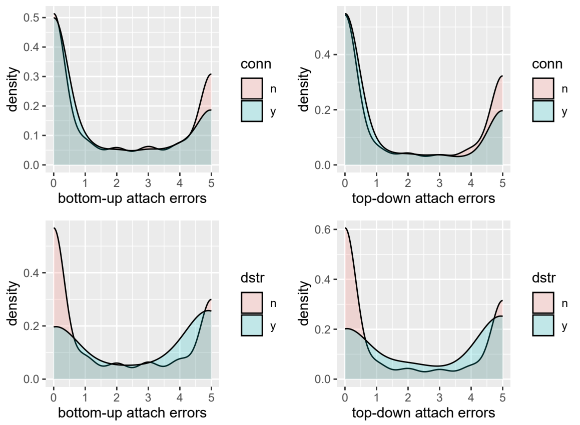

Figure 2: Attachment Error Count Density with and without DMs or Distractors for Each Architecture in `dev+test` of GUM v9.

facilitate parsing and that distractors should be harmful, it is an open question whether such markers will remain important once we know about other factors known to cause problems, such as OOV items, EDU text length, and intra-sentential status. To compare these, we construct several regression models predicting the number of errors. Because the distribution of error numbers is U-shaped, as shown in Figure 2, we use mixed effects Beta regression with a random effect for document identity, and re-scale the number of **attachment or relation errors** to the range 0–1, where 1 means the maximum of 5 model errors. Table 5 shows the significance of each predictor in each model.[3]

Looking first at GUM on the left, Table 5 shows that, when given only DMs and distractors, both features are significant in predicting errors above a per-document random effect baseline, for both architectures. In other words, predicting implicit relations is unsurprisingly harder in RST, just as it is for PDTB-style shallow discourse parsing, and distractors make things even harder.

However, adding the subordination feature (middle two models for GUM v9), which indicates whether an EDU is in a subordinate clause (and therefore likely to have an intra-sentential relation), removes the significance of the presence of a DM (but not of distractors). This suggests DMs are less important in predicting errors (or lack thereof) than intra-sentential status. Adding some more predictors, a fuller model with EDU length, OOV rate (the percentage of lexical items not seen during training per EDU), and genre does not remove the

[3]Significance for `genre`, a multi-nominal feature, is computed via a likelihood ratio test comparing the model with and without this predictor.

corpus	GUM v9						RST-DT					
architecture	bot-up	top-down	bot-up	top-down	bot-up	top-down	bot-up	top-down	bot-up	top-down	bot-up	top-down
`dm`	<.001***	<.001***	0.059	0.074	0.003**	0.005**	0.988	0.002**	0.244	<.001***	0.445	<.001***
`distractor`	<.001***	<.001***	<.001***	<.001***	<.001***	<.001***	<.001***	<.001***	<.001***	<.001***	<.001***	<.001***
`subord`			<.001***	<.001***	<.001***	<.001***			<.001***	<.001***	<.001***	<.001***
`length`					<.001***	<.001***					<.001***	<.001***
`oov`					0.115	0.262					0.944	0.563
`genre`					<.001***	<.001***						

Table 5: Results of the Regression Models for GUM v9 and RST-DT from both Architectures.

significance of subordination status, and shows that OOV rate is not a significant predictor in this setting. The more complex models with 6 features also restore some significance for DMs, albeit to a lesser degree than other predictors.

Moving to RST-DT, we see a similar pattern, except for a surprising difference between architectures: in the mixed effects model, presence of a DM is ***not*** a significant predictor for the bottom-up architecture, while it is significant for top-down. This pattern is repeated across all sets of features on the right side of Table 5. For RST-DT, since we do not have gold syntactic dependency trees, we use gold intra-sentential relation status to represent the `subord` feature. This feature remains highly significant in all models across architectures. Finally, adding all the features to the right-most models (excluding `genre`, since RST-DT is all newswire), OOV rate again fails to reach significance, while all other features are significant, except for DMs for the bottom-up architecture models.

These numbers suggest several things: first and most important, while DMs may be somewhat important, some representation of intra-sentential status is the more robust predictor of parsing errors. This effect persists even if we know about other plausible features, such as EDU length and OOV rate. This observation fits with the line of work mentioned above on multi-stage models for RST parsing, which attempt to learn separate models for intra-sentential and inter-sentential or inter-paragraph models (e.g. Kobayashi et al. 2020). Although joint models can perform well on all levels regardless, we can confirm that there are substantial differences between these types.

In terms of architecture differences, results for RST-DT suggest more sensitivity to DMs for top-down models, but this result is not reproduced in GUM. Finally, all models are sensitive to distractors, which raises questions about the nature of this sensitivity—what kinds of errors are parsers making, and more specifically are they predicting relations corresponding to distractor DMs? We address these questions in the next sections.

4.2 Predicting Parsing Errors

The results in the previous section quantify the importance of different characteristics of discourse relations in promoting errors, and the relative difficulty of implicit relations in SOTA RST parsing.

However, the linear model comparing the significance of explicit DMs, distractors, and features such as EDU length or OOV rate is rather naive and leaves out a variety of potentially relevant properties of subtrees, such as total number of attached discourse units (which could contribute to ambiguity), or the gold relation to be predicted—some relations are easier to recognize or are less ambiguous, and some relations have high prior likelihood, making guessing them a safe bet. Although these properties may not be useful for realistic prediction of errors when we do not have a gold parse, they can be of interest for understanding tree properties which are difficult for parsers to get right.

To make matters even more complex, the factors mentioned above interact in subtle ways with each other and with explicit marking status. For example, CONTINGENCY relations are easy to recognize thanks to the reliable DM ‘if’ as in (1), but this is not always the case, as in (2) which uses subject-verb inversion to mark a conditional. Some relations are almost never marked by DMs, but may still be easy, such as ATTRIBUTION, which can be identified via speech verbs, as in (3).

(1) [Um **if** you don’t want to do a tour of Pittock Mansion,] $\xrightarrow{\text{gold:CONTINGENCY}}$ [I’d still recommend like taking the trail up there]$_{\text{GUM\_vlog\_portland}}$

(2) [“**Had it happened** an hour later] $\xrightarrow{\text{gold:CONTINGENCY}}$ [It would have been much worse]$_{\text{GUM\_news\_crane}}$

(3) [Any judge in this country would **agree**] $\xrightarrow{\text{gold:ATTRIBUTION}}$ [that opening and closing statements along are not a trial.]$_{\text{GUM\_speech\_impeachment}}$

This complexity means that a realistic model of difficult parsing environments may need to consider more variables, and the interactions mean that a simple linear model cannot capture the rich patterns in the data. In this section, we therefore use XGBoost (Chen and Guestrin, 2016), a highly accurate ensemble gradient boosting framework which is able to harness arbitrary interactions between features and is highly regularized to prevent overfitting, meaning it can be expected to find a near-optimal mapping of our variables to parser error occurrences. For this experiment, we will attempt to predict 'hard' EDUs, which we define as EDUs which most models predict incorrectly.

However, it is not immediately clear what kinds of features we should allow the model to use: on the one hand, we would like to know what constellations in gold RST trees are difficult, including the gold relation label or the relative importance of being a leaf node vs. a hub with many dependents, as well as the contributions of DMs and distractors. On the other hand, in a realistic scenario we would not be able to know whether a DM is a distractor without knowing the gold relation, and we would not know how many dependents a node really has.

We thus construct two models: the **REALISTIC** model only has access to features that can reasonably be predicted without the gold parse, including EDU length in tokens, presence of DMs (whether helpful or distracting), the incoming syntactic dependency relation (which can be predicted by a syntax parser), the OOV rate, and genre. The **FULL** model, by contrast, has access to all gold features, including the gold relation class, intra-/inter-sentential status, DM vs. distractor presence etc. The first model is more relevant for realistic scenarios in which we want to diagnose where parser errors are more likely (or how many we might incur), while the second is more helpful for understanding what is hard in an RST graph given the gold graph itself. Figure 3 gives an analysis of feature importances using classification gain for both REALISTIC and FULL, which score 67.26% and 76.28% respectively over a majority baseline score of 58.26%, which predicts that RST parsers will never be wrong, for the bottom-up architecture. For top-down, the scores of the two models are 65.26% and 76.61% respectively.

The XGBoost library's plots automatically highlight the most important features for both parser architectures, which for the **REALISTIC** model is only **the syntactic function of the EDU**. This likely indicates the overwhelming importance of knowing whether an EDU has a typical intra-sentential role, such as a relative or adverbial clause, which is likely to be predicted correctly. The next features begin with length (short EDUs are likely to have similar ones attested in training data compared to long ones), then genre (since some genres are harder), and only then the typical NLP difficulty predictor, the OOV rate (which is slightly less useful when EDU length is also known, since the two correlate). The last feature, presence of DMs, is still useful but less so, especially since it folds in occurrences of helpful and distracting DMs. There are no substantial differences between top-down and bottom-up here for GUM v9.

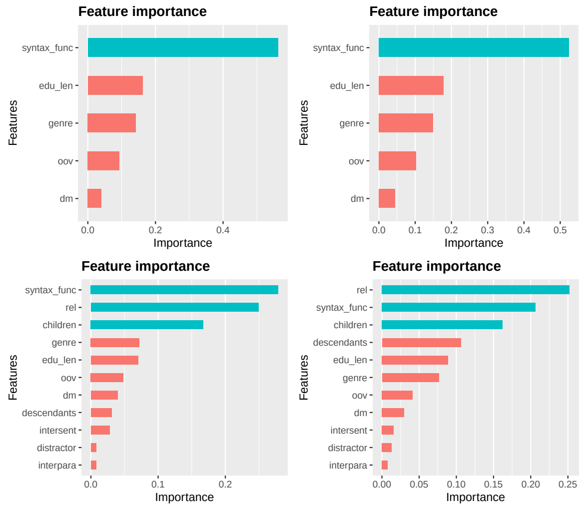

Figure 3: Feature Importances for the **REALISTIC** (top) and **FULL** (bottom) XGBoost Models for GUM from both BOTTOM-UP (left) and TOP-DOWN (right) Architectures. Very important features are highlighted in **teal**.

Turning to the **FULL** model, we see that syntactic function is still very important: it beats gold label for bottom-up models and follows it for top-down. Some relations are easier than others, or different subsequent conditions apply to them, and this matters about as much as the syntactic attachment type. Number of children (a measure of tree centrality vs. leaf status) is third, only then followed by length and genre, which are still quite helpful. Number of descendants (which is correlated with children) follows for top-down, but is far lower for bottom-up parsers. We then see OOV rate outranking DMs, which outrank less important features, such as the no longer crucial inter-sentential/inter-paragraph status, which are also highly correlated with some of the features above (syntax for the former, number of children for the latter, since many children are typical of paragraph head units). Finally distractors are second to last, far below DMs, also because they are rare.

These models indicate that predicting errors without knowing the gold tree is challenging, but a gain of 7–9% over baseline is still possible, mainly by looking at syntactic structure, which indicates intra-sentential status—a predictor much more valuable than DM marking. By contrast, when looking at gold trees, hard parts can most easily be associated with hard relations and syntactic environments, but combining all of the available features leads to an impressive ability to predict where parser models will likely go wrong, with ∼18% gain over baseline.

4.3 The Nature and Meaning of Distractors

Although the previous results suggest distractors play a minor role, their independent correlation with errors and the fact that DMs are generally relevant to discourse relations, raise questions regarding their very existence: why do they appear and how exactly do they affect parsers?

To begin with the second question, we examined the 174 distractors in GUM `test`. For most bottom-up models, which made fewer errors on these cases, 116/174 (66%) were still erroneous, and we decided to manually label whether the majority model-predicted label was consistent with the distractor: if the gold relation is ELABORATION, the distractor is *but*, and the prediction is ADVERSATIVE, then prediction is consistent with the distractor, but if the prediction is CONTINGENCY, then it is not. We use PDTB's mapping of connectives to classes to match DMs to relations.

For 74/116 cases (63.8%), the majority label was consistent with the distractor—in other words, the parser may be predicting based on a DM which would normally signal a competing relation. This brings us to the second question: if the relations signaled by distractors are incorrect, why are the distractors present? As an example, we consider two such cases from GUM, shown in (4)–(5).

(4) [**if** Steven didn't see it as weird] $\xrightarrow[\text{pred:CONTINGENCY}]{\text{gold:EXPLANATION}}$ [why should it bother us?]$_{\text{GUM\_fiction\_teeth}}$

(5) [so the reason seems to be that there are things out there that put even these kaiju to shame] $\xleftarrow[\text{pred:ADVERSATIVE}]{\text{gold:EVALUATION}}$ [**But** even this presents a problem]$_{\text{GUM\_reddit\_monsters}}$

In (4), the gold tree has the 'if'-clause as a justification for why it 'shouldn't bother us', which makes sense pragmatically; but formally, the clause seems like a legitimate conditional marked by *if*, and parsers predict CONTINGENCY. In (5), the annotation focuses on the evaluative meaning of the words 'a problem', while parsers, probably provoked by *But*, predict ADVERSATIVE.

We therefore suspect that multiple, concurrent relations may actually hold in data where distractors appear, which is a standard possibility in frameworks like PDTB, where relations are identified based on the presence of DMs. If this applies in RST as well, then in a sense, such parser errors are not really errors at all. Because RST enforces a strict tree constraint, the only way to find out would be to look at alternative RST trees.

In order to do just this, we utilize RST-DT's official double-annotated subset, which has trees from a second annotator for 53 documents. This subset overlaps only 5 documents in the RST-DT test set, which contain only 12 distractors, meaning that the scope of this last analysis is limited; however, in examining these 12 distractors, we discovered that 75% (9/12) actually corresponded to relations **selected as the primary RST relations** by the second annotator in the double annotated data. In other words, the double annotated data confirms that, at least in the case of the RST-DT test set, a large majority of distractors do in fact correspond to multiple concurrent relations which were identified by an experienced RST annotator.

5 Conclusion

This study has several important implications. Firstly and unsurprisingly, the explicit/implicit distinction from shallow discourse parsing is mirrored in RST parsing difficulty, and the dataset released in this paper can help study it further. However, explicit marking is clearly less consequential than intra-sentential status, with which explicitness it correlated. Secondly, OOV rate plays a less important role than we initially suspected, while genre effects remain robust, suggesting that diverse genres may matter more than subject matter. Our results also indicate that current architectures do not differ substantially in what they get right or wrong, and with scores being so similar, differences reduce to computational efficiency and personal preference.

Finally, the study of distractors suggest that RST's tree constraint may mix some cases of multiple concurrent relations with parsing errors, when parsers are actually identifying viable relations. This suggests we may want to consider ways of allowing and adding concurrent relations to RST parses.

References

Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Studies in Natural Language Processing. Cambridge University Press, Cambridge.

Katherine Atwell, Junyi Jessy Li, and Malihe Alikhani. 2021. Where are we in discourse relation recognition? In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 314–325, Singapore and Online. Association for Computational Linguistics.

Katherine Atwell, Anthony Sicilia, Seong Jae Hwang, and Malihe Alikhani. 2022. The change that matters in discourse parsing: Estimating the impact of domain shift on parser error. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 824–845, Dublin, Ireland. Association for Computational Linguistics.

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*, Text, Speech and Language Technology 22, pages 85–112. Kluwer, Dordrecht.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

Zeyu Dai and Ruihong Huang. 2018. Improving implicit discourse relation classification by modeling inter-dependencies of discourse units in a paragraph. 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 141–151, New Orleans, Louisiana. Association for Computational Linguistics.

Debopam Das and Maite Taboada. 2014. RST Signalling Corpus Annotation Manual. Technical report, Simon Fraser University.

Debopam Das and Maite Taboada. 2017. Signalling of coherence relations in discourse, beyond discourse markers. *Discourse Processes*, 55(8):743–770.

Debopam Das, Maite Taboada, and Paul McFetridge. 2019. RST Signalling Corpus. LDC2015T10.

Vera Demberg, Fatemeh Torabi Asr, and Merel Scholman. 2019. How compatible are our discourse annotation frameworks? insights from mapping RST-DT and PDTB annotations. *Dialogue & Discourse*, 10(1):87—135.

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.

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.

René Knaebel. 2021. discopy: A neural system for shallow discourse parsing. In *Proceedings of the 2nd Workshop on Computational Approaches to Discourse*, pages 128–133, Punta Cana, Dominican Republic and Online. Association for Computational Linguistics.

Naoki Kobayashi, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. 2020. Top-down RST parsing utilizing granularity levels in documents. In *AAAI Conference on Artificial Intelligence*.

Sujian Li, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014. Text-level discourse dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25–35, Baltimore, Maryland. Association for Computational Linguistics.

Jessica Lin and Amir Zeldes. 2021. WikiGUM: Exhaustive entity linking for wikification in 12 genres. In *Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 170–175, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yang Liu. 2019. Beyond the Wall Street Journal: Anchoring and comparing discourse signals across genres. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 72–81, Minneapolis, MN. 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.

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(35):314–317.

Yang Janet Liu and Amir Zeldes. 2023. Why can't discourse parsing generalize? A thorough investigation of the impact of data diversity. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3104–3122, Dubrovnik, Croatia. Association for Computational Linguistics.

Zhengyuan Liu, Ke Shi, and Nancy Chen. 2021. DMRST: A joint framework for document-level multilingual RST discourse segmentation and parsing. In *Proceedings of the 2nd Workshop on Computational Approaches to Discourse*, pages 154–164, Punta Cana, Dominican Republic and Online. Association for Computational Linguistics.

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.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Special Issue on Using Large Corpora, Computational Linguistics*, 19(2):313–330.

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.

Philippe Muller, Stergos Afantenos, Pascal Denis, and Nicholas Asher. 2012. Constrained decoding for text-level discourse parsing. In *Proceedings of COLING 2012*, pages 1883–1900, Mumbai, India. The COLING 2012 Organizing Committee.

Noriki Nishida and Hideki Nakayama. 2020. Unsupervised discourse constituency parsing using Viterbi EM. *Transactions of the Association for Computational Linguistics*, 8:215–230.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Rashmi Prasad, Bonnie Webber, and Aravind Joshi. 2014. Reflections on the Penn Discourse TreeBank, comparable corpora, and complementary annotation. *Computational Linguistics*, 40(4):921–950.

Caroline Sporleder and Alex Lascarides. 2004. Combining hierarchical clustering and machine learning to predict high-level discourse structure. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 43–49, Geneva, Switzerland. COLING.

Manfred Stede and Arne Neumann. 2014. Potsdam Commentary Corpus 2.0: Annotation for discourse research. In *Proceedings of the Language Resources and Evaluation Conference (LREC '14)*, pages 925–929, Reykjavik.

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.

Bonnie Webber, Rashmi Prasad, Alan Lee, and Aravind Joshi. 2019. The Penn Discourse TreeBank 3.0 annotation manual. Technical report, University of Edinburgh, Interactions, LLC, University of Pennsylvania.

Nan Yu, Meishan Zhang, Guohong Fu, and Min Zhang. 2022. RST discourse parsing with second-stage EDU-level pre-training. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4269–4280, Dublin, Ireland. Association for Computational Linguistics.

Amir Zeldes. 2017. The GUM Corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Zheng Zhao and Bonnie Webber. 2021. Revisiting shallow discourse parsing in the PDTB-3: Handling intrasentential implicits. In *Proceedings of the 2nd Workshop on Computational Approaches to Discourse*, pages 107–121, Punta Cana, Dominican Republic and Online. Association for Computational Linguistics.

Yilun Zhu, Sameer Pradhan, and Amir Zeldes. 2021. OntoGUM: Evaluating contextualized SOTA coreference resolution on 12 more genres. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 461–467, Online. Association for Computational Linguistics.