A Discourse Signal Annotation System for RST Trees*

Luke Gessler, Yang Liu, and Amir Zeldes
Department of Linguistics
Georgetown University
{lg876,yl879,amir.zeldes}@georgetown.edu

Abstract

This paper presents a new system for open-ended discourse relation signal annotation in the framework of Rhetorical Structure Theory (RST), implemented on top of an online tool for RST annotation. We discuss existing projects annotating textual signals of discourse relations, which have so far not allowed simultaneously structuring and annotating words signaling hierarchical discourse trees, and demonstrate the design and applications of our interface by extending existing RST annotations in the freely available GUM corpus.

1 Introduction

Discourse signals help language users recognize semantic and pragmatic relationships that hold between clauses and sentences in discourse, also known as coherence or rhetorical relations. Discourse markers such as coordinations (e.g. 'but'), subordinating conjunctions (e.g. 'although'), and adverbials (e.g. 'instead') are usually considered the most explicit signals and are relatively well studied, but work on other types of discourse signals has been more limited. These include semantic, syntactic, and morphological features; for example, repeated mention, parallel syntactic constructions, and inflection for tense and aspect can also signal discourse relations.

Building corpora annotated for discourse signals is important for empirical studies of how writers and speakers signal relations in naturally occurring text, and how readers or hearers are able to recognize them. However, for one of the most popular frameworks for analyzing discourse relations, Rhetorical Structure Theory (RST, Mann and Thompson 1988), there are currently no tools which allow full-featured annotation of both RST trees and signals. RST is a functional theory of text organization that interprets discourse as a hierarchical tree of clauses or similar discourse units, meaning that annotation interfaces must accommodate the complexity of tree structures. The main contribution of this paper is in enabling a completely new type of annotation within the framework of RST, simultaneously targeting the ways in which humans construct discourse trees and identify relations in a single system. Although we base our work on an existing RST interface, the expansions presented here bridge a substantial gap in RST annotation, which has to date been unable to link complete trees to concrete discourse signal positions in a single annotation tool and format, linking specific tokens and other signaling devices to positions in the tree.

Our system, shown in Figure 1, features state of the art support for viewing and editing signals, and benefits from an underlying interface offering full RST editing capabilities, called rstWeb (Zeldes, 2016). No installation is needed for end users in a project since the tool is web-based, and annotators can easily collaborate. Docker images and a local version are available for easy deployment and we make all code available open source via GitHub.[1]

2 Previous Work

Numerous studies have examined discourse signals (e.g. Knott and Sanders 1998), but the largest corpora with signal annotations have been produced in the framework of the Penn Discourse Treebank (PDTB, Prasad et al. 2008, and similarly for Chinese, Zhou and Xue 2012, and other languages) and the RST Signalling Corpus (RST-SC, Taboada and Das 2013), both built on top of

*We would like to thank Richard Eckhart de Castilho, Debopam Das, Nathan Schneider and Maite Taboada, as well as three anonymous reviewers for valuable comments on earlier versions of this paper and the system it describes.

[1] https://github.com/amir-zeldes/rstweb

Proceedings of Discourse Relation Parsing and Treebanking (DISRPT2019), pages 56–61
Minneapolis, MN, June 6, 2019.

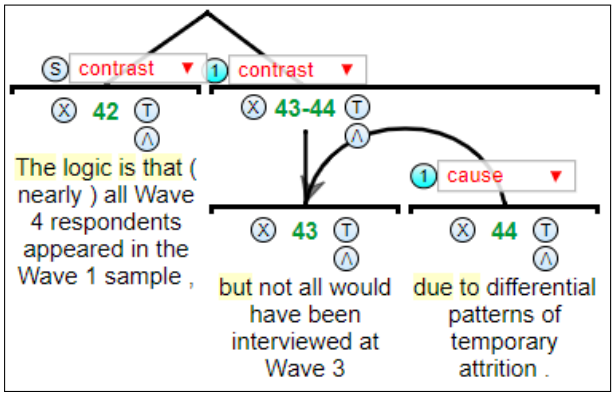

Figure 1: Signaling in a discourse tree fragment.

The $2.5 billion Byron 1 plant near Rockford, Ill., was completed in 1985.
In a disputed 1985 ruling, the Commerce Commission said Commonwealth Edison could raise its electricity rates by $49 million to pay for the plant.
But state courts upheld a challenge by consumer groups to the commission's rate increase and found the rates illegal.
The Illinois Supreme Court ordered the commission to audit Commonwealth Edison's construction expenses and refund any unreasonable expenses.

Figure 2: PDTB connective and argument spans.

the text of the Wall Street Journal corpus (Marcus et al., 1993). We examine the tools used to produce these, as well as other approaches, below.

2.1 Discourse Signals in PDTB

PDTB employs a lexically grounded approach to discourse relations and their signals by annotating 1) *Explicit* and *Implicit* connectives and their associated argument spans, which are not constrained to be single clauses or sentences; 2) *supplementary information* that is considered relevant but not required for the interpretation; 3) textual expressions that establish coherence other than connectives called *Alternative Lexicalizations* (*AltLex*); 4) relation senses for *Explicit* and *Implicit* connectives and *AltLex* relations; 5) attribution within discourse relations including categories such as *source, type, scopal polarity,* and *determinacy* (Prasad et al., 2008). Unlike RST, which identifies hierarchic structures in text, PDTB-style annotations do not form a hierarchy and need not cover the entire text.

According to Prasad et al. (2014), annotation workflow in PDTB-style resources has varied in the development of comparable corpora in other languages (e.g. Zhou and Xue 2012) and genres (e.g. Prasad et al. 2011), which could potentially affect annotator effort and inter-annotator agreement (e.g. Sharma et al. 2013). For instance, when annotating the example in Figure 2 where an Explicit connective *But* is present, one could easily identify the argument spans by highlighting them in different colors (i.e. Arg1 in yellow and Arg2 in blue in Figure 2) as well as the sense tag associated with it, in this case, *Comparison.Concession*. Thus, depending on what elements can be found in text to reliably identify relations (i.e. either an argument or a connective), the annotation workflow may differ. Moreover, potential span overlaps with other relations are not problematic, since each relation is annotated independently, and signals for multiple relations are not visualized simultaneously. Annotators can thus annotate relations and signals concurrently. However, this tool is not suitable for the type of annotation addressed by our tool, since no hierarchy of discourse units can be represented in the way required for RST trees.

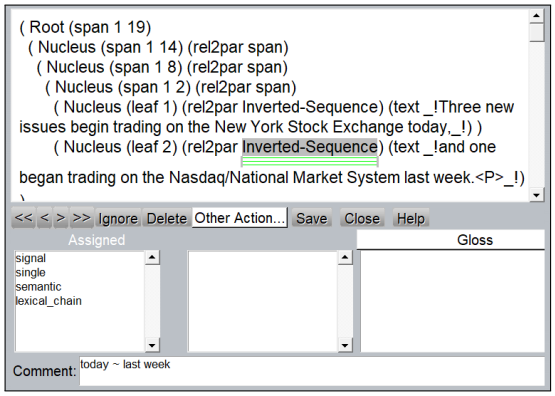

Figure 3: Signal annotation from RST-SC in the UAM tool.

2.2 Signals in RST-SC

Since RST originally did not foresee annotating relation signals, RST-SC takes existing trees in the RST Discourse Treebank (RST-DT, Carlson et al. 2003) as a ground truth, and adds explicit annotations for how each relation can be identified. Because trees are hierarchical, annotations apply not to spans of text, but to relations attached to nodes in the tree. Multiple signals corresponding to different words are possible for the same relation, and some signals do not correspond to words in the text (e.g. genre conventions, graphical layout and more). Since we also annotate signals in RST trees, this corpus is the most comparable to what we aim to produce with the tool described here.

Because of the lack of an annotation tool capable of simultaneously representing RST trees and signal spans, Taboada and Das (2013) used the UAM CorpusTool (O'Donnell, 2008), illustrated in Figure 3, to annotate the underlying LISP

format files of RST-DT directly. The UAM tool is not aware of the LISP bracket structure of the RST tree: annotators simply add underlines to any span in the plain text file and categorize them, taking care to add annotations only to the position of the label of the relation being signaled, a potentially error prone process. In Figure 3, an *Inverted-Sequence* relation ("Three new issues begin trading ... and one began...", with the temporal sequence inverted) has three signals, each corresponding to a green underline. To switch between annotations, users may click on an underline – the selected one in this case has the signal type 'semantic', subtype 'lexical_chain'.

Since UAM cannot connect the signal annotation to specific tokens, RST-SC provides no information about the location of the signaling tokens. In other words, unlike in PDTB, annotations are not anchored to words in the text. For instance, the 'lexical_chain' signal shown in Figure 3 corresponds to the words 'today' and 'last week', which signal the temporal relation in the text (the comment box in the figure confirms this, though such comments are not consistently available in RST-SC, and the location of the word in the comment is not notated unambiguously, if the word occurs multiple times).

To explore the actual words corresponding to RST-SC signals, Liu and Zeldes (2019) anchored annotations to word positions using a tabular grid based interface called GitDox (Zhang and Zeldes, 2017), in addition to UAM. Annotators were asked to locate relevant information in UAM and transfer the results, including signal types/subtypes, source/target of relations and associated tokens, to GitDox. Because GitDox only provides a tabular spreadsheet-like input, relation names and positions were indicated as plain text annotations of the relevant signal tokens, a process which is slow and error prone. Liu and Zeldes (2019) reached moderate agreement on anchoring the existing signal annotations (see Table 1 below), and concluded that a better tool was critical for the task.

2.3 Other Tools

In addition to the tools mentioned above, RhetDB (Pardo, 2005) has also been used to annotate signals. RhetDB does allow for the annotation of discourse signals, but its limitations include its inability to graphically represent a full RST tree and the fact that it only runs on the Windows operating system locally.

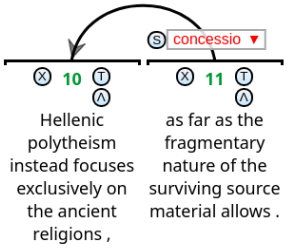

Figure 4: Two discourse units joined with a relation in the interface. The "X" clears the node's parent relation, the "T" adds a span above the node, and the "∧" creates a multinuclear relation. Relations are edited via drag-and-drop.

The Basque RST TreeBank (Iruskieta et al., 2013) includes visualizations of discourse signals, but these signals cannot be viewed in the context of a fully graphically represented RST tree, and are instead represented as a separate annotation layer in a dedicated interface built for the corpus.

3 Implementation

3.1 rstWeb

The signal annotation system was developed on top of an existing interface, rstWeb. rstWeb (Zeldes, 2016) is a web application that allows collaborative, online annotation of RST trees. It was intended to replace an older desktop application, RSTTool (O'Donnell, 2000), which is no longer being maintained. Developed in Python and JavaScript and running in the browser, it allows administrators to set up projects for annotators, assign them multiple versions of documents for annotation experiments, and control files and schemas centrally. Annotators need only a browser and login, and all annotations and versions of files, including optional annotation step logs, are collected on a server.

rstWeb provides a solid foundation for our signal annotation system. Its feature-set for RST annotation tasks is mature and flexible, and unlike older RST interfaces, minimizes the clicks required for common tasks by avoiding multiple modes for linking/unlinking relations and creating nodes. To maintain this advantage, we chose to integrate signal annotation into the same environment used for building RST trees, rather than a separate mode (see Figure 4).

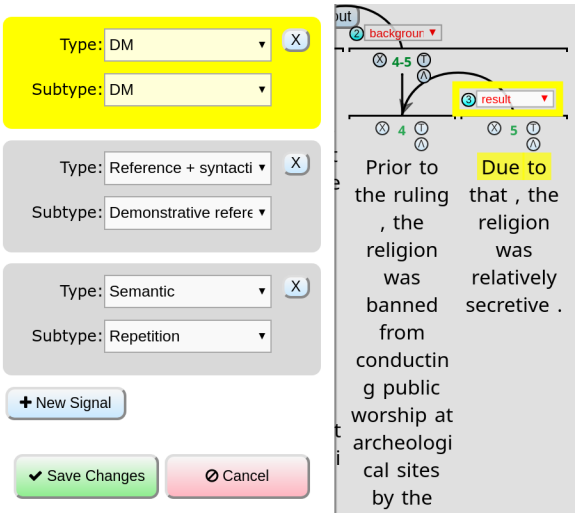

Figure 5: The signal sidebar, toggled by a button labeled "S" next to a relation. Signals have a type, subtype, and associated tokens, highlighted with a click. Here three signals indicate a RESULT relation: One, the discourse marker (DM) "due to", has been selected in the sidebar, highlighting its associated tokens. Clicking on other signals, such as Semantic, subtype Repetition below it, highlights their associated tokens.

3.2 The Signal Annotation System

Prior to this work, rstWeb had no support for signal annotation. The contribution of the present work was to build a signal annotation system on top of rstWeb to allow annotators to view and edit signal annotations and make these available for export and use in downstream tasks.

In the signal annotation system, annotators may associate signals with relations after they are created by hitting a button next to the relation which opens a sidebar (see Figure 5). Different annotation workflows are conceivable, including only annotating signals once RST trees are complete or annotating signals and building RST trees in tandem, as well as either annotating all kinds of signals by going over the entire text once, or focusing on one relation type at a time (e.g. annotating signals for all CAUSE relations first, then moving on to the next type, etc.).[2]

Once associated with a relation, a signal can be linked with any subset of tokens in the text. The significance of the signal annotation system is in enabling RST analysts to annotate discourse signals with a feature-set that is more comprehensive and ergonomic than any other existing RST interface. Signal types are fully configurable, with no restriction on the placement and number of tokens that may be associated with a signal, and relations can be associated with multiple signals.

3.3 Data Model

A signal in our system consists of four elements:

1. A relation whose type (RESULT, CONCESSION, etc.) the signal is helping to indicate
2. A possibly empty list of tokens which comprise the signal
3. A type that categorizes the signal according to its linguistic nature
4. A more fine-grained subtype

Each relation from 1. can have multiple signals having elements 2.-4., and 2. can be an empty set, as some signals may have *no* associated tokens. For example, RST-SC assumes that factors such as genre conventions or graphical layout (e.g. a sequence of indented paragraphs or bullet points, even when no token encodes a bullet point glyph) can be used by writers to signal a meaningful structure, which readers can identify. Our interface supports such explicit, typed annotations, without reference to specific token indices.

The introduction of signals anchored to tokens creates a new complication for the representation format of RST data, the commonly used `.rs3` XML format: since RST trees only connect discourse units, word level tokenization has been ignored in RST annotation tools to date. However, because our annotations associate tokens with the relations they are cues for, and users are meant to click or drag across cue words to mark them as signals, tokenization is essential to signal annotation. To address this, we have added automatic tokenization facilities for imported documents written in alphabetic languages using a Python port[3] of the TreeTagger tokenizer[4]; built-in tokenization for Asian languages and morphologically rich languages remains outstanding, but for these languages pre-tokenized data that has been processed with appropriate tools can be imported.

The signal *type* and *subtype* attributes categorize annotations based on a pre-determined annotation scheme. By default, rstWeb uses the types from RST-SC (Das and Taboada, 2018), but any annotation scheme can be defined, and multiple

[2] We thank an anonymous reviewer for noting these potentially different workflows.

[3] https://github.com/amir-zeldes/rstWeb/blob/develop/modules/whitespace_tokenize.py

[4] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

schemes can be maintained to accommodate multiple projects on the same installation. The `.rs3` format used by rstWeb and RSTTool was extended to include signal information. The new format, `.rs4`, is backward compatible with both tools, meaning that files containing signal information may be opened with RSTTool (though the signals cannot be displayed).

4 Evaluation

To assess the benefits of our interface for signal annotation, we re-annotated a pilot data set of three documents from RST-SC, containing 506 tokens with just over 90 signals. In Table 1 we compare our results to scores achieved for the same annotation task of anchoring RST-SC data to specific signal tokens in Liu and Zeldes (2019).

	L&Z19	**this paper**
% identical	86.0	90.9
Cohen's kappa	0.52	0.77

Table 1: Comparison with Liu and Zeldes (2019).

Next to the numerical results showing an improvement in kappa, annotators reported the new interface was much easier and faster to work with. Feedback from the original annotators of RST-SC also suggests the interface is much more suited to the signal annotation task.

5 Applications and Outlook

We are currently using the interface presented here to annotate RST signals in GUM (Zeldes, 2017), a freely available, richly annotated corpus with 126 documents and some 109,000 tokens across eight genres: academic, biography, fiction, interviews, news, travel guides, how-to guides and reddit forum discussions. Since the scheme by Das and Taboada (2017) is based solely on Wall Street Journal articles, signal types and subtypes need to be extended to cover more genres. For instance, RST-SC genre features include subtypes such as *Newspaper Layout, Newspaper Style Attribution* and *Newspaper Style Definition*; however, these are not enough to represent other genre-specific layouts – e.g. in academic articles (headings, formulas etc.). We are also working on search and visualization facilities to explore data annotated with discourse trees and signals. We plan to use the ANNIS platform (Krause and Zeldes, 2016), which already visualizes RST trees, and add interactive ways to explore signaling tokens in documents as well as signals for individual relations, which we view as an important extension to RST.

One of the goals of the current project is to learn which new types of signals are needed to describe signaling in different text types, and to discover differences in signals across genres. These in turn will help us to develop new models of the features used in discourse relation identification, which may be more or less general, or language and text-type specific. We are also exploring how human annotated signal spans compare with the words most attended to by neural models for automatic relation classification (see Zeldes 2018:178-188 for some first results). With the release of an easy-to-use interface for signal annotation within the RST framework, we hope that more corpora with signal-enhanced RST trees will be developed in more languages, and advance our understanding of how readers identify relations in practice.

References

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.

Debopam Das and Maite Taboada. 2017. Signalling of coherence relations in discourse, beyond discourse markers. *Discourse Processes*, pages 1–29.

Debopam Das and Maite Taboada. 2018. RST signalling corpus: A corpus of signals of coherence relations. *Language Resources and Evaluation*, 52(1):149–184.

Mikel Iruskieta, María Jesús Aranzabe, Arantza Diaz de Ilarraza, Itziar Gonzalez, Mikel Lersundi, and Oier Lopez de la Calle. 2013. The RST Basque TreeBank: an online search interface to check rhetorical relations. In *Proceedings of the 4th Workshop on RST and Discourse Studies*.

Alistair Knott and Ted Sanders. 1998. The classification of coherence relations and their linguistic markers: An exploration of two languages. *Journal of Pragmatics*, 30(2):135–175.

Thomas Krause and Amir Zeldes. 2016. ANNIS3: A new architecture for generic corpus query and visualization. *Digital Scholarship in the Humanities*, 31(1):118–139.

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.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Michael O'Donnell. 2000. RSTTool 2.4 – a markup tool for Rhetorical Structure Theory. In *Proceedings of the International Natural Language Generation Conference (INLG 2000)*, pages 253–256, Mitzpe Ramon, Israel.

Michael O'Donnell. 2008. The UAM CorpusTool: Software for corpus annotation and exploration. In *Proceedings of the XXVI congreso de AESLA*, pages 3–5, Almeria, Spain.

Thiago Alexandre Salgueiro Pardo. 2005. *Métodos para análise discursiva automática.* Ph.D. thesis, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

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 6th International Conference on Language Resources and Evaluation (LREC 2008)*, pages 2961–2968, Marrakesh, Morocco.

Rashmi Prasad, Susan McRoy, Nadya Frid, Aravind Joshi, and Hong Yu. 2011. The Biomedical Discourse Relation Bank. *BMC bioinformatics*, 12(1):188.

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.

Himanshu Sharma, Praveen Dakwale, Dipti M Sharma, Rashmi Prasad, and Aravind Joshi. 2013. Assessment of different workflow strategies for annotating discourse relations: A case study with HDRB. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 523–532. Springer.

Maite Taboada and Debopam Das. 2013. Annotation upon annotation: Adding signalling information to a corpus of discourse relations. *Dialogue and Discourse*, 4(2):249–281.

Amir Zeldes. 2016. rstWeb - a browser-based annotation interface for rhetorical structure theory and discourse relations. In *Proceedings of NAACL-HLT 2016 System Demonstrations*, pages 1–5, San Diego, CA.

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

Amir Zeldes. 2018. *Multilayer Corpus Studies*. Routledge Advances in Corpus Linguistics 22. Routledge, London.

Shuo Zhang and Amir Zeldes. 2017. GitDOX: A linked version controlled online XML editor for manuscript transcription. In *Proceedings of FLAIRS 2017, Special Track on Natural Language Processing of Ancient and other Low-resource Languages*, pages 619–623.

Yuping Zhou and Nianwen Xue. 2012. PDTB-style discourse annotation of Chinese text. In *Proceedings of ACL 2012*, pages 69–77, Jeju Island, Republic of Korea.