How much progress have we made on RST discourse parsing? A replication study of recent results on the RST-DT

Mathieu Morey and **Philippe Muller** and **Nicholas Asher**
IRIT, Université Toulouse & CNRS, Univ. Paul Sabatier
{mathieu.morey, philippe.muller, nicholas.asher}@irit.fr

Abstract

This article evaluates purported progress over the past years in RST discourse parsing. Several studies report a relative error reduction of 24 to 51% on all metrics that authors attribute to the introduction of distributed representations of discourse units. We replicate the standard evaluation of 9 parsers, 5 of which use distributed representations, from 8 studies published between 2013 and 2017, using their predictions on the test set of the RST-DT. Our main finding is that most recently reported increases in RST discourse parser performance are an artefact of differences in implementations of the evaluation procedure. We evaluate all these parsers with the standard Parseval procedure to provide a more accurate picture of the actual RST discourse parsers performance in standard evaluation settings. Under this more stringent procedure, the gains attributable to distributed representations represent at most a 16% relative error reduction on fully-labelled structures.

1 Introduction

While several theories of discourse structure for text exist, discourse parsing work has largely concentrated on Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) and the RST Discourse Treebank (RST-DT) (Carlson et al., 2003), which is the largest corpus of texts annotated with full discourse structures. The RST-DT, annotated in the style of RST, consists of 385 news articles from the Penn Treebank, split into a training and test sets of 347 and 38 documents.The standard evaluation procedure for RST discourse parsing, RST-Parseval, proposed by Marcu (2000), adapts the Parseval procedure for syntactic parsing (Black et al., 1991). RST-Parseval computes scores on discourse structures with no label (S for Span) or labelled with nuclearity (N), relation (R) or both (F for Full). The semantic nature of discourse relations makes discourse parsing a difficult task. However, the recent introduction of distributed representations of discourse units has seemingly led to significant improvements, with a claimed relative error reduction of 51% on fully labelled structures. As part of a broader study of methods and evaluation metrics for discourse parsing, we collected predictions from nine RST discourse parsers and reimplemented RST-Parseval. In section 2, we present these RST parsers and report their published scores on RST-Parseval. In section 3, we replicate their evaluation and show that most of the heterogeneity in performance across RST parsers arises from differences in their evaluation procedures. In section 4, we replace RST-Parseval with the standard Parseval procedure and obtain a more accurate picture of the actual performance of RST parsers.

2 A sample of RST discourse parsers

Almost all RST discourse parsers are evaluated on the test section of the RST-DT using manually segmented Elementary Discourse Units (EDUs). We contacted by email the main or corresponding author of each recently (2013–2017) published, text-level RST discourse parser evaluated in this setting and asked the authors to provide us with the predictions they used in their study or a procedure that would enable us to reproduce identical or at least similar predictions. When our attempts were unsuccessful we tried to reproduce similar predictions from published materiel (source code, binaries, model). We managed to obtain or reproduce predictions for 9 parsers from 8 studies. The first parser, denoted **HHN16 HILDA**, is a reimplementation (Hayashi et al., 2016) of the classic, bottom-up, greedy HILDA parser with a linear SVM model (Hernault et al., 2010) ; this parser serves as a reference point to evaluate the progress made by more recent parsers. **SHV15**

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1330–1335
Copenhagen, Denmark, September 7–11, 2017.

D is a variant of the HILDA parser with different models (perceptron for attachment of discourse units, logistic regression for relation labelling) and a slightly different feature set adapted to use predicted syntactic dependency trees (Surdeanu et al., 2015). **JCN15 1S-1S** is a two stage (sentence- then document-level) CKY chart parser with Dynamic Conditional Random Field (DCRF) models, in its 1 sentence - 1 subtree (1S-1S) variant that builds a document-level RST tree on top of sentence-level subtrees built for each sentence independently (Joty et al., 2013, 2015). **FH14 *g*CRF** is a two stage (sentence- then document-level) bottom-up, greedy parser with linear-chain CRF models (Feng and Hirst, 2014). We use the version of the parser available on the author's webpage, that lacks post-editing and contextual features. **BPS16** is a sequence-to-sequence parser, heuristically constrained to build trees, with a hierarchical neural network model (hierarchical bi-LSTM) (Braud et al., 2016). **LLC16** is a CKY chart parser with a hierarchical neural network model (attention-based hierarchical bi-LSTM) (Li et al., 2016). **BCS17 mono, BCS17 cross+dev** are two variants of a transition-based parser that uses a feed-forward neural network model (Braud et al., 2017). **JE14 DPLP** is a shift-reduce parser that uses an SVM model (Ji and Eisenstein, 2014). We use predictions provided by the author, from an improved, unpublished version of the parser.

The first four parsers (HHN16 HILDA, SHV15 D, JCN15 1S-1S, FH14 *g*CRF) use, as features, only localist representations of the input and parsing state, i.e. surface-form and syntactic information: length of discourse units (DUs), distance between DUs, n-grams of words and POS tags, relations of syntactic dominance between DUs... The last five parsers (BPS16, LLC16, BCS17 mono and cross+dev, JE14 DPLP concat) build distributed representations of DUs, complemented with a subset of localist representations.

The authors used various implementations of RST-Parseval, but all applied a right-heavy binarization procedure to the reference RST trees: Each node of arity greater than 2 is replaced with a right-branching cascade of binary nodes. In the publications, the tables of results provide a unique score for labeled structures, corresponding to either the R or F metric, with no explicit distinction. The F_1 scores published in the literature for the parsers in our sample are reported in Table 1, where an en-dash (–) indicates missing scores. We also report the scores of human agreement, computed and reported by Joty (2015), over the doubly annotated subset of the RST-DT consisting of 53 documents (48 from train, 5 from test).

parser	**S**	**N**	**R** or **F**
HHN16 HILDA	82.6	66.6	54.2
SHV15 D	–	–	55.2
JCN15 1S-1S	82.6	68.3	55.8
FH14 *g*CRF	84.9	69.9	57.2
BPS16	83.6	69.8	55.1
LLC16	**85.8**	71.1	58.9
BCS17 mono	85.0	72.3	60.1
BCS17 cross+dev	85.1	**73.1**	61.4
JE14 DPLP concat	82.1	71.1	**61.6**
human	88.7	77.7	65.8

Table 1: Published F_1 scores.

The parsers in the second group seem to perform markedly better than the parsers in the first group, especially on the hardest subtasks of predicting (partly or fully) labelled structures (N and R or F). Collectively, the parsers in the second group claim absolute improvements over the parsers in the first group by 0.9, 3.2 and 4.2 points, corresponding to a relative error reduction of 24% on S, 41% on N and 51% on R or F, compared to human agreement. While discourse parsing is a difficult, semantic task with relatively little annotated training data, authors attribute these significant gains to the capacity of distributed representations to capture latent semantic information and generalize over a long tail of alternative surface forms. As a preliminary step towards probing these claims, we replicated the evaluation of these parsers' predictions.

3 Evaluation

We collected or reproduced predictions from each parser and replicated the evaluation procedure [1]. The predictions came in various formats: bracketed strings as in the RST-DT, lists of span descriptions, trees or lists of attachment decisions. We wrote custom functions to load and normalize the predictions from each parser into RST trees. While we favor evaluating against the original,

[1]The source code and material are available at https://github.com/irit-melodi/rst-eval

non binarized reference RST trees, we conformed in this replicative study to the de facto standard in the RST parsing literature: We transformed the reference RST trees into right-branching binary trees and used these binary trees as reference in all our evaluation procedures. We also examined the source code from the evaluation procedures provided by the authors to determine whether the published scores corresponded to the R or F metric. In so doing we noticed a potentially important discrepancy in the various implementations of the RST-Parseval procedure: the implementations used to evaluate the parsers in the first group compute **micro-averaged** F_1 scores, as is standard practice in the syntactic parsing community, whereas the implementations used to evaluate the parsers in the second group compute **macro-averaged** F_1 scores across documents. The micro-averaged F_1 score is computed globally over the predicted and reference spans from all documents ; the macro-averaged F_1 score across documents is the average of F_1 scores computed independently for each document.

We implemented both strategies and report the corresponding scores in two separate tables. Parsers originally evaluated with micro-averaging scores are in the top half of each table, parsers originally evaluated with macro-averaged scores in the bottom half. An asterisk (*) marks parsers for which we reproduced predictions using code and material made available by the authors, although the experimental settings are not guaranteed to match exactly those from the original study. A double asterisk (**) marks a parser for which we used predictions generated by the author using an improved, unpublished version of the parser posterior to the original study. Lines with no asterisk in Tables 2 to 4 correspond to parsers whose authors sent us their original predictions. Replicated scores expected to match scores in Table 1 are underlined.

Table 2 contains the micro-averaged F_1 scores on each metric (S, N, R, F). As expected, parsers in the first group obtain micro-averaged scores equal or close to their published scores reported in Table 1. More strikingly, the micro-averaged scores for the parsers in the second group are much lower than their published scores [2] and most of their claimed advantages over the parsers in the first group has vanished. On S and N, parsers in the second group do not improve over parsers in the first group ; on R and F the best parser in the second group provides an absolute improvement of 0.9 and 1.4 points. This improvement corresponds to a relative error reduction of 11% on R and 16% on F, much lower than the 51% claimed in the literature. [3]

parser	S	N	R	F
HHN16 HILDA	82.6	66.6	54.6	54.3
SHV15 D *	82.6	67.1	55.4	54.9
JCN15 1S-1S	82.6	68.3	55.8	55.4
FH14 gCRF *	**84.3**	**69.4**	56.9	56.2
BPS16	79.7	63.6	47.7	47.5
LLC16	82.2	66.5	51.4	50.6
BCS17 mono	81.0	67.7	55.7	55.3
BCS17 cross+dev	81.3	68.1	56.3	56.0
JE14 DPLP **	82.0	68.2	**57.8**	**57.6**
human	88.3	77.3	65.4	64.7

Table 2: Micro-averaged F_1 scores.

parser	S	N	R	F
HHN16 HILDA	85.9	72.1	60.0	59.4
SHV15 D *	85.1	71.1	59.8	59.1
JCN15 1S-1S	85.7	73.0	60.9	60.2
FH14 gCRF *	**87.0**	**74.1**	61.1	60.5
BPS16	83.6	69.8	55.4	55.1
LLC16	85.4	70.8	58.4	57.6
BCS17 mono	85.0	72.3	60.8	60.1
BCS17 cross+dev	85.1	73.1	61.6	61.4
JE14 DPLP **	85.0	71.6	**62.0**	**61.9**
human	89.6	78.3	66.7	65.3

Table 3: Macro-averaged F_1 scores.

Table 3 contains the macro-averaged F_1 scores. Parsers in the first group obtain macro-averaged scores markedly higher than the micro-averaged scores from Table 2. Parsers in the second group obtain macro-averaged scores that are equal or close to the published scores reported in Table 1, which confirms our analysis of the source code of their evaluation procedures. The global picture on

[2] The milder decrease of the DPLP scores, especially on S, is directly attributable to improvements in the latest, unpublished version of the parser.

[3] Our replicated scores for human agreement are 0.4 points lower than those published on S, N, R, possibly due to different approaches in handling divergences in EDU segmentation on the doubly annotated subset of documents.

macro-averaged scores is consistent with that on micro-averaged scores: On S and N, parsers in the second group do not improve over parsers the first group and the best parser brings an absolute improvement of 0.9 and 1.4 points on R and F. On each metric, the two lowest scores are obtained by parsers from the second group.

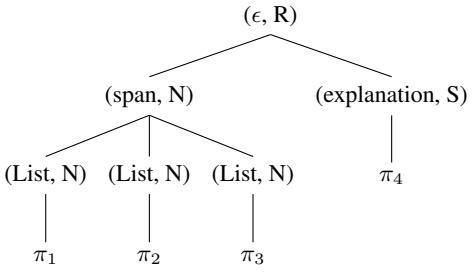
To sum up, parsers in the first group have identical scores in Tables 1 and 2, except for slight differences between our evaluation procedure and the authors', or between the predictions used in our evaluation compared to the original study. The second group of parsers have identical scores in Tables 1 and 3, modulo the same factors. The (exactly or nearly) matching entries between Tables 1, 2 and 3, underlined in Tables 2 and 3, are evidence of the two averaging strategies (micro in Table 2, macro in Table 3) used by the authors in their publications (Table 1). A comparison between Tables 2 and 3 reveals that the averaging strategy similarly affects both groups of parsers. As a result, the performance level among recent RST discourse parsers is much more homogeneous than the situation depicted in the literature. The distributed representations of DUs computed and used in JE14 DPLP (Ji and Eisenstein, 2014) and possibly BCS17 cross+dev (Braud et al., 2017) plausibly capture semantic information that helps with predicting discourse relations and structure, but the current experimental results do not provide a similarly strong support for BPS16 (Braud et al., 2016), LLC16 (Li et al., 2016) and BCS17 mono (Braud et al., 2017).

More generally, it is important that authors compute and report scores that accord with standard practice, unless duly motivated. The standard practice in syntactic parsing is to report micro-averaged scores for overall performance, often complemented with macro-averaged scores over classes to gain valuable insight into the average performance of parsers across labels, especially infrequent ones. Early work in RST discourse parsing follows this practice, reporting micro-averaged scores for global performance, plus distinct scores for each relation class or macro-averaged scores over all relation classes (Hernault et al., 2010; Feng and Hirst, 2014). The latter should not be confused with the scores published for BPS16, LLC16, BCS17 (mono, cross+dev) and JE14 DPLP, which are macro-averaged over documents.

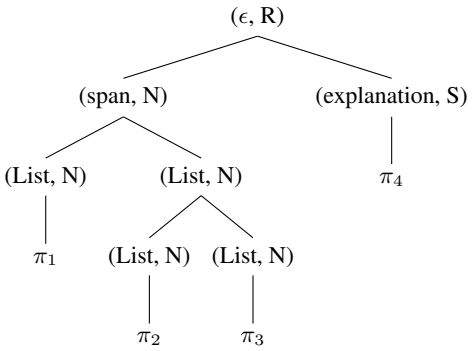
4 Elements for a fairer evaluation

RST-Parseval crucially relies on an encoding of RST trees into constituency trees such that the rhetorical relation names are placed on the children nodes, and the nuclei of mononuclear relations are conventionally labelled SPAN. RST-Parseval resembles the original Parseval, except it considers a larger set of nodes to collect all nuclearity and relation labels in this encoding: the root node (whose label and nuclearity are fixed by convention) is excluded and the leaves, the EDUs, are included. On the one hand, RST-Parseval can handle discourse units of arity greater than 2, in particular those consisting of a nucleus independently modified by two satellites through distinct mononuclear relations. This avoids introducing discourse units that were not part of the original annotation, which a preliminary binarization of trees would have induced. On the other hand, RST-Parseval considers approximately twice as many nodes as the original Parseval would on binarized trees (at most $2n - 2$ nodes for n EDUs, compared to $n - 1$ attachments in a binary tree), and the relation labels of most nuclei are redundant with the nuclearity of a node and its sister (SPAN for a nucleus whose sisters are satellites, and the same label as its sisters for a nucleus whose sisters are nuclei). Both aspects artificially raise the level of agreement between RST trees, especially when using manual EDU segmentation.

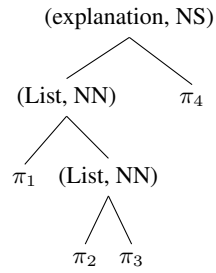
However, all the parsers in our sample except (Sagae, 2009; Heilman and Sagae, 2015) predict binary trees over manually segmented EDUs and evaluate them against right-heavy binarized reference trees. In this setting, Marcu's encoding of RST trees RST-Parseval are no longer motivated. We can thus revert to using the standard Parseval procedure on a representation of binary RST trees where each internal node is a labelled attachment decision to obtain a more accurate evaluation of RST parser performance. Figure 1 represents (a) an original RST tree using Marcu's encoding, (b) its right-heavy binarized version, (c) the tree of labelled attachment decisions for the right-heavy binarized tree. To the best of our knowledge, we are the first to explicitly use an evaluation procedure for RST parsing closer to the original Parseval for syntax, although the trees of labelled attachment decisions we use directly correspond to the trees built by many RST parsers, eg. shift-reduce parsers. Table 4 provides the micro-averaged F_1



(a) Original RST tree



(b) Right-heavy binarized tree



(c) Labelled attachment decisions for the right-heavy binarized tree

Figure 1: Original RST tree, right-heavy binarization and labelled attachment decisions

scores for the parsers in our sample, using Parseval.

Parseval is more stringent than RST-Parseval, with the best system obtaining 46.3 on fully labelled structures (F). Parsers in the first group are competitive with parsers in the second group, outperforming them on S and to a lesser extent on N. Parsers in the second group reduce relative error by 8% on R and 16% on F, much lower than the published figures in the literature.

5 Conclusion

We replicated standard evaluation procedures in RST discourse parsing for 9 parsers and showed that most gains reported in recent publications are an artefact of implicit differences in evaluation procedures. We also showed how to use the standard Parseval procedure instead of Marcu's adaptation RST-Parseval, which artificially raises scores. Overall, the recent gains attributable to distributed representations represent at most a relative error reduction of 16%. Our study reveals an urgent need for the RST discourse parsing community to re-examine and standardize their evaluation procedures.

parser	S	N	R	F
HHN16 HILDA	65.1	54.6	44.7	44.1
SHV15 D *	65.3	54.2	45.1	44.2
JCN15 1S-1S	65.1	55.5	45.1	44.3
FH14 gCRF *	**68.6**	**55.9**	45.8	44.6
BPS16	59.5	47.2	34.7	34.3
LLC16	64.5	54.0	38.1	36.6
BCS17 mono	61.9	53.4	44.5	44.0
BCS17 cross+dev	62.7	54.5	45.5	45.1
JE14 DPLP **	64.1	54.2	**46.8**	**46.3**
human	78.7	66.8	57.1	55.0

Table 4: Micro-averaged F_1 scores on labelled attachment decisions (original Parseval).

Acknowledgments

This research was supported by ERC Advanced Grant n. 269427. We thank the authors of the parsers who made their predictions and evaluation scripts available, and the reviewers for helpful comments.

References

E. Black, S. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. Procedure for quantitatively comparing the syntactic coverage of english grammars. In *Proceedings of the Workshop on Speech and Natural Language*, HLT '91, pages 306–311. Association for Computational Linguistics.

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

Chloé Braud, Barbara Plank, and Anders Søgaard.

2016. Multi-view and multi-task training of rst discourse parsers. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1903–1913, Osaka, Japan. The COLING 2016 Organizing Committee.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2003. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In Jan van Kuppevelt and Ronnie Smith, editors, *Current Directions in Discourse and Dialogue*, pages 85–112. Kluwer Academic Publishers.

Vanessa Wei Feng and Graeme Hirst. 2014. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 511–521, Baltimore, Maryland. Association for Computational Linguistics.

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

Michael Heilman and Kenji Sagae. 2015. Fast Rhetorical Structure Theory Discourse Parsing. *arXiv preprint*, arXiv:1505.02425.

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

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

Shafiq Joty, Giuseppe Carenini, and Raymond T Ng. 2015. Codra: A novel discriminative framework for rhetorical analysis. *Computational Linguistics*, 41(3):385–435.

Shafiq R Joty, Giuseppe Carenini, Raymond T Ng, and Yashar Mehdad. 2013. Combining intra-and multi-sentential rhetorical parsing for document-level discourse analysis. In *ACL (1)*, pages 486–496.

Qi Li, Tianshi Li, and Baobao Chang. 2016. Discourse parsing with attention-based hierarchical neural networks. In *EMNLP*, pages 362–371.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical Structure Theory: Towards a Functional Theory of Text Organization. *Text*, 8(3):243–281.

Daniel Marcu. 2000. *The theory and practice of discourse parsing and summarization*. MIT press.

Kenji Sagae. 2009. Analysis of discourse structure with syntactic dependencies and data-driven shift-reduce parsing. In *Proceedings of the 11th International Conference on Parsing Technologies*, IWPT '09, pages 81–84, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mihai Surdeanu, Thomas Hicks, and Marco A Valenzuela-Escárcega. 2015. Two practical rhetorical structure theory parsers. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 1–5.