

Aligning Discourse and Argumentation Structures using Subtrees and Redescription Mining

Laurine Huber¹, Yannick Toussaint¹,
Charlotte Roze¹, Mathilde Dargnat² and Chloé Braud¹

¹ Université de Lorraine, CNRS, Inria, LORIA (UMR 7503), F-54000 Nancy, France
firstname.lastname@loria.fr,

² ATILF, Université de Lorraine, CNRS (UMR 7118), Nancy, France
et Institut des Sciences Cognitives Marc Jannerod, CNRS (UMR 5304), Bron, France
mathilde.dargnat@univ-lorraine.fr

Abstract

In this paper, we investigate similarities between discourse and argumentation structures by aligning subtrees in a corpus containing both annotations. Contrary to previous works, we focus on comparing sub-structures and not only relation matches. Using data mining techniques, we show that discourse and argumentation most often align well, and the double annotation allows to derive a mapping between structures. Moreover, this approach enables the study of similarities between discourse structures and differences in their expressive power.

1 Introduction

This paper presents preliminary results in aligning different text structure representations. Using graph and redescription mining, we compare argumentative and discourse trees. The former represents the way arguments are organized through support or attack relations, the latter accounts for the coherence of texts by linking segments with semantico-pragmatic relations.

Aligning structures such as argumentation and discourse trees could help to understand the links between these representations, to build some bridges between theories, or to allow a better understanding of the expressive power of the different formalisms.

The arg-microtexts-multilayer corpus¹ (Stede et al., 2016) provides three representations of short texts: RST trees (Mann and Thompson, 1988),² SDRT graphs (Lascares and Asher, 2007), and argumentation (ARG) structures as described in (Peldszus and Stede, 2013), based on Freeman’s theory (Freeman, 1991).

In this preliminary study, we focus on RST and ARG annotations. We propose to describe each

text by two views, one corresponding to the set of subtrees extracted from the RST tree, and the other to the subtrees extracted from the ARG tree. The best alignment between subtrees is computed thanks to a redescription mining approach.

2 Related Work

A manual analysis of the correspondences between RST and argumentation relations (Peldszus and Stede, 2016) has already shown that a 1-to-1 mapping leads to some mismatches. For example, 39% of the supports, 72% of the rebuts, and 33% of the undercuts do not have a corresponding RST edge. These mismatches have been explained by granularity differences in annotations. Thus, we propose here to consider alignment at the level of subtrees allowing more complex combinations of relations.

Cabrio et al. (2013) showed that 5 Argumentation Schemes (AS) from (Walton et al., 2008) can be mapped to Penn Discourse TreeBank (PDTB) relations (Prasad et al., 2008). They built an hypothetical mapping of AS to PDTB relations, and extracted 10 examples from the PDTB. Two annotators had to say whether the AS definition was relevant to the example, and the measured Cohen’s kappa showed a significant agreement ($\kappa = 0.71$). Though their goal was to study the link between argumentation and discourse, their approach was based on both human hypothesis and human annotation. Unlike them, we propose an automatic approach based on data mining. To our knowledge, it is the first generic and systematic approach for mapping discourse and argumentation.

3 Methodology

The three-step process aims at finding an exhaustive and systematic alignment over the corpus between “parts” of the RST and “parts” of the ARG representations. First, for each text, its RST and ARG representations are respectively transformed

¹<https://github.com/peldszus/arg-microtexts>

²<https://www.sfu.ca/rst/>

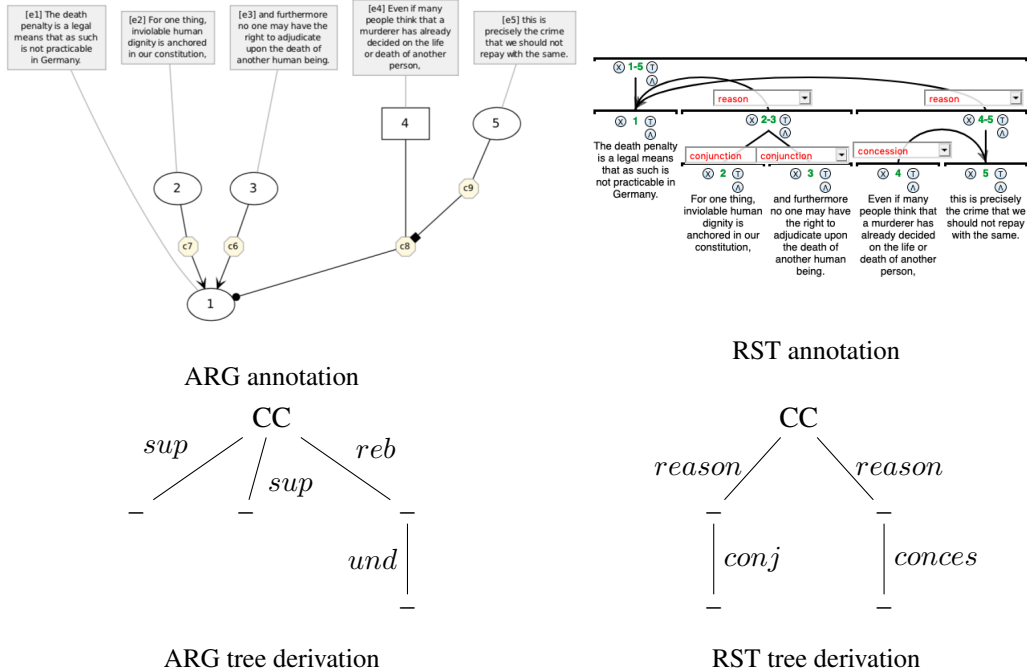


Figure 1: Text *micro_b006* annotated in ARG and RST and trees derived from annotations

into two labeled trees. Then, RST and ARG trees being considered separately, subgraph mining extracts all subtrees common to at least 2 texts. Each subtree becomes a feature used for describing one of the two views (i.e. RST and ARG) of each text. In the last step, redescription mining searches for alignments between features of the RST view and features of the ARG view.

3.1 Encoding RST and ARG representations into trees

RST and ARG representations are encoded into two distinct trees, referred as initial RST or ARG trees in the following. In both ARG and RST trees, we label CC the root node, to represent the central claim and the main nucleus respectively.³

As we do not consider the text but only the structure, and because we do not consider the sequentiality of the units in the texts proposed by (Wachsmuth et al., 2017), other nodes are left unlabeled (-). These unlabeled nodes represent the Argumentative Units (AUs) in ARG trees and the Discourse Units (DUs) in RST trees.⁴ Labels on the edges correspond to the argumentative or discursive relations between AUs and DUs, respectively. Trees are built in a straightforward

³The “most nuclear” unit or central unit (Stede, 2008).

⁴An AU can comprise multiple DUs, to express that multiple DUs form an argument only when combined.

way, except in ARG representations when some relations are not directed to an AU but to a relation (undercut for example). Inspired from Wachsmuth et al. (2017), we make them target the premise of the undercut relation (see ARG tree example in Figure 1).

3.2 Creating two views

In this step, we extract subtrees from the whole set of RST initial trees on the one hand, and subtrees from ARG initial trees on the other hand. Each subtree is given a unique identifier and becomes a boolean feature to be associated with texts: a text has a feature if the subtree is in its initial tree.

Extracting subtrees gSpan (*Graph-Based Substructure Pattern Mining*) (Yan and Han, 2002) is an algorithm to extract frequent subtrees from a graphset \mathbb{G} . Informally, a graph h is a *subtree* of a graph g if h is contained in g , and h is *frequent* if, given a support threshold σ_s , at least s graphs of \mathbb{G} contain h . We applied gSpan,⁵ on the ARG treeset and the RST treeset with $\sigma_s = 2$.

Building the data-tables From the gSpan runs, we represent the boolean features in two binary

⁵We use the following python implementation: <https://github.com/betterenvi/gSpan> as we are interested in subtrees that include the central claim, we used the algorithm on undirected graphs.

data-tables, called views, where the rows correspond to the texts and the columns to the features.

3.3 Redescription Mining

In data analysis, redescription mining (Galbrun and Miettinen, 2017) is the task of finding two distinct characterizations of the same set of objects (i.e. texts in this experiment). Inputs of redescription mining are the views of the texts. The goal is to find two boolean expressions, called queries, $q1$ and $q2$, where $q1$ and $q2$ are formulae over the features of the ARG view and the RST view respectively, and where the support of $q1$ and $q2$ are sufficiently similar, so that they explain (approximately) the same set of texts. This similarity is measured by the Jaccard index:

$$J(q1, q2) = \frac{\text{supp}(q1 \wedge q2)}{\text{supp}(q1 \vee q2)} \quad (1)$$

where $\text{supp}(q)$ is the number of texts where q occurs. In other words, Jaccard quantifies how big the overlapping between the objects that evaluate true in $q1$ and those that evaluate true in $q2$ is.

The exploration strategy of ReRemi is based on atomic updates. First, the algorithm computes the Jaccard for all possible pairs of atomic queries, in other words all redescrptions that can be built from one feature for each view. These pairs are ranked following their Jaccard in a decreasing order and the n best pairs are kept. Starting with the best pair, the algorithm applies operations (addition, deletion, edition) on either query to improve the candidate redescription until no further improvement of the Jaccard can be done. The first redescription has been built. The algorithm then iterates on the remaining best pairs. We use the ReRemi Algorithm (Galbrun and Miettinen, 2012) implemented in Siren with the predefined parameters of the tool. Conjunctions and disjunctions are allowed in the queries but the length of a query is restricted to 4. This length restriction has an impact on the redescrptions that are found. The algorithm must maximize the Jaccard with maximum 4 features in each query. Thus some patterns that we may want to observe do not appear in the queries, because if they were used instead of another, they will make the Jaccard lower.

4 Data

The corpus contains 112 micro-texts, each of them answering a controversial issue (e.g., “Should Germany introduce the death penalty?”). We

use 2 of the 3 types of annotations provided: the RST and argumentation representations (Peldszus and Stede, 2015). RST trees are annotated with 28 relations, with 2 to 12 relations per tree, the most frequent relations are: *reason* (180), *concession* (65), *list* (63), *conjunction* (44), *antithesis* (32), *elaboration* (37), and *cause* (20). ARG trees have 2 to 9 relations, 5 distinct relations are annotated, the most frequent being *support* (263), *rebut* (108) and *undercut* (63).

gSpan produces 311 RST and 98 ARG features, both sharing at least 2 objects.⁶ The most frequent RST feature occurs in 105 texts while the most frequent ARG feature occurs in 94 texts. Only 22 RST features are shared by more than 10 texts, and 18 ARG features are shared by more than 13 texts.

5 Results

	q1	q2	J(q1,q2)	# texts
<i>Rd1</i>	a57	r40 ∨ r65 ∨ r123	0.691	54
<i>Rd2</i>	a58	r61 ∨ r119 ∨ r125	0.351	13
<i>Rd3</i>	a23 ∨ a59	r125	0.3	8

Table 1: Examples 3 redescrptions. aX and rX correspond resp. to ARG and RST subtrees.

Table 1 gives three redescription examples over the 31 obtained. For reason of space, we mainly comment and discuss these three redescrptions. *Rd1* has the highest Jaccard value (0.691). The support value is 54, meaning that 54 texts share both the left and the right part of the redescription. *Rd2* is a specialisation of *Rd1*, and, finally, *Rd3* is a redescription where the conjunction is on the argumentation side. Subtrees corresponding to the query features are drawn in Figure 2, 3 and 4.

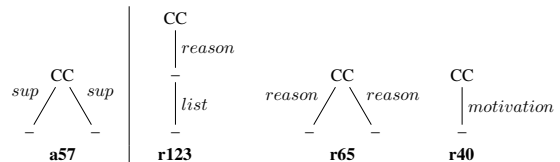


Figure 2: Subtrees corresponding to features of *Rd1*.

The 54 texts described by *Rd1* all contain feature a57 in their ARG tree, but the disjunction on the RST query emphasizes a difference in the granularity of ARG and RST formalisms. More precisely from the 54 texts, 30 contain r123, 22

⁶Because the parameter $\sigma_s = 2$ is given to gSpan.

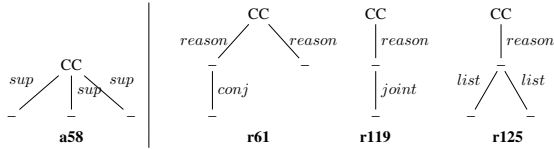


Figure 3: Subtrees corresponding to features of *Rd2*.

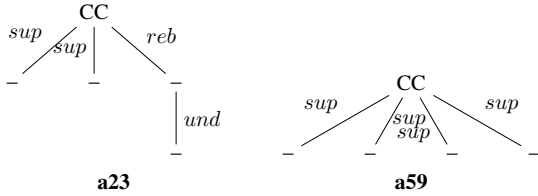


Figure 4: Subtrees corresponding to ARG features of *Rd3*, for *r125* feature see Figure 4

contain *r65*, 2 contain *r40*. In other words, in half of the data, when the ARG structure contains two support pointing to the CC, the RST tree includes either a *reason* relation followed by two *list*, or two *reason* relations, or one *motivation* directed to the CC. The objects described by *Rd2* and *Rd3* are also described by *Rd1* so *Rd2* and *Rd3* can be seen as specialisations of *Rd1*. *Rd2* can be read in the same way as *Rd1*, namely from 23 texts containing *a58*, 13 are aligned with either *r61* (3), or *r119* (3) or *r125* (7). Contrary to *Rd1* and *Rd2*, the disjunction on the ARG side in *Rd3* suggests that the feature *r125* (appearing 8 texts) can be mapped into two different ARG structures: *a59* which concerns 2 texts, and *a23* which concerns 5 texts.

6 Discussion

A 1(ARG)-to-many(RST) mapping: One interesting question that arises, when looking at *Rd1*, concerns the disjunction on the discourse side, i.e. different discourse structures that represent the same argumentative structure. Thus, this redescription could be an illustration of granularity difference between the two representations, RST being more fine-grained than ARG. However, by looking more precisely at the texts, we distinguish two different issues related to granularity as described below.

Granularity in labeling relations: We observe that an edge labeled by a given ARG relation can be aligned with RST edges with different labels. The explanation could be that a limited set of 5 relations is used to annotate argumentation while the annotators were given a larger set of 28 relations

to annotate discourse.

As shown in (Peldszus and Stede, 2016), a support relation can correspond in fact to several distinct RST relations, most often *reason* and *justification* (found in another redescription), but also *motivation*, as also found via our extraction procedure. The lower frequency of *motivation* compared to *reason* could come from the fact that the latter is more generic, the former being used only to motivate actions (Carlson and Marcu, 2001).

Granularity in structure: *Rd1* seems to imply a similarity between RST structures described by *r123* and *r65*, that is a *list* embedded under a *reason*, and two *reason* directed to the CC. The *list* relation links comparable items, which is not mandatory for two *reasons* annotated independently. This fine level of granularity is not expressed in ARG trees.

Moreover, for the two cases with a *motivation* relation, we notice that the embedded node is in fact annotated either with a *list* or with a *conjunction*, the latter being very similar to the former and thus corresponding to another compatible structure. Note here that we do not extract a subtree parallel to *r123* but involving a *motivation* by applying our method: while this structure exists, it only appears once in the data, and thus does not meet our minimum support criterion. Lowering the support threshold is an option, it could be compared to the use of a relation grouping to allow an automatic recognition a priori of similar RST labels.

If we assume that discourse structures are more fine grained than argumentation structures, we could parametrised ReReMi to extract only atomic ARG queries to obtain a redescription of each single ARG structure. However, the following study of *Rd3* comes to contradict this hypothesis.

Depth and width of the subtrees: Some redescriptions with lower support and Jaccard concern deeper or larger subtrees than in *Rd1*. For example in *Rd2*, the *a58* subtree includes *a57*, *r125* includes *r123*, and *r61* includes *r65*.

It thus seems meaningful to consider that a deeper/larger structure in one view is aligned with a deeper/larger structure in the other view. Thus, we would have liked to consider *Rd2* as a specialisation of *Rd1*, emphasizing the following: when embedded within a relation matching a support,

the multinuclear RST relations `conjunction` and `list` express an additional support in ARG.

However, we observe that a new subtree, namely `r119`, occurs in the RST part of *Rd2*. This non parallel new subtree being mapped to `a58` still needs further investigation.

A many(ARG)-to-1(RST) mapping: As RST structures are more fine grained than argumentation structures, we could expect ARG structures to be aligned with a disjunction of RST structures. However, the other way around is also true. *Rd3* emphasizes that some RST (`r125`) structures are aligned with a disjunction of ARG structures.

The `r125` RST tree contains 3 `lists` in a `reason` related to `CC`. On the ARG side, `a23` contains two `supports` and a sequence of `rebut` and `undercut` directed to the `CC`. The third expected support for `a23` to partially map `r125` comes from the following assertion: if an `X` undercuts a `Y`, which in turn rebuts the `CC`, then `X` is in a support relation to `CC`.

The pattern *a59* is founded in 5 texts, but only 2 of them contain `r125` in their RST representation. Here, 4 support nodes in ARG are mapped to 3 `lists` in RST. For these 2 texts, the fourth expected support relation comes from deeper elements in the trees. In one of the texts, a segment that is in a `e-elaboration` in one of the `list` element is used as a support in the ARG tree. In the second text, a `restatement` plays the same role. Thus, nodes involved in these substructures are split into two `supports` in the ARG annotation. Despite a small Jaccard, this many-to-1 mapping is very informative.

Weakness due to tree representation: One weakness of our tree representation is that we omit the position of the segments in the text. Doing this for ARG and RST subtrees extraction allows to consider the subtrees regardless of their place in the text. However, features aligned by a redescription do not necessarily refer to the same part of the text.

In Figure 1, while text *b006* illustrates *Rd1*, the segments contained in `a57` and `r123` do not correspond: a correct mapping would align both support in ARG with the `conjunction` embedded in the `reason` in RST (segments 2 and 3). In the same way, the ARG `undercut` and `rebut` in sequence would have been mapped to

the RST `concession` and `reason` in sequence (segments 4 and 5).

7 Conclusion

The alignment of text structures can be done with redescription mining applied on subtrees. The automatic process, compared to manual methods, enables a systematic comparison of different formalisms. Applied to a small corpus of argumentative texts, this preliminary experiment demonstrates the effectiveness of our approach to compare structures in different frameworks, but also to get insights on the encoding used within a specific formalism.

Several improvements are currently under study. First, as we only used the predefined parameters of ReRemi, we can reparametrize it. We can restrict the ARG side of the redescription to an atomic query in order to associate a conjunction of RST subtrees to each ARG subtree. We may also reparametrize ReRemi to get a higher number of redescrptions and possibly longer queries. Indeed, the 4-features limitation in a query blocks the emergence of more interesting features (deeper subtrees) in the redescrptions.

Second, tree representations should include links to text segments in order to enable a fairer alignment between ARG and RST structures.

Finally, the methodology could also be extended to other formalisms (e.g. SDRT), or used to provide a grouping of substructures from one theory to another.

8 Acknowledgement

This work was supported partly by the french PIA project "Lorraine Université d'Excellence", reference ANR-15-IDEX-04-LUE, and the PEPS blanc from CNRS (INS2I).

References

- Elena Cabrio, Sara Tonelli, and Serena Villata. 2013. [From Discourse Analysis to Argumentation Schemes and Back: Relations and Differences](#). In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, João Leite, Tran Cao Son, Paolo Torroni, Leon van der Torre, and Stefan Woltran, editors, *Computational Logic in Multi-Agent Systems*,

- volume 8143, pages 1–17. Springer, Berlin, Heidelberg.
- Lynn Carlson and Daniel Marcu. 2001. Discourse tagging reference manual. Technical report, University of Southern California Information Sciences Institute.
- James B Freeman. 1991. *Dialectics and the macrostructure of arguments: a theory of argument structure*. Foris Publications, Berlin.
- Esther Galbrun and Pauli Miettinen. 2012. From black and white to full color: extending redescription mining outside the Boolean world. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(4):284–303.
- Esther Galbrun and Pauli Miettinen. 2017. *Redescription Mining*. SpringerBriefs in Computer Science. Springer International Publishing.
- Alex Lascarides and Nicholas Asher. 2007. *Segmented Discourse Representation Theory: Dynamic Semantics With Discourse Structure*. In Harry Bunt and Reinhard Muskens, editors, *Computing Meaning*, volume 3. Springer Netherlands, Dordrecht.
- William Mann and Sandra Thompson. 1988. Rhetorical structure theory: Towards a functional theory of text organization. *TEXT*, 8:243–281.
- Andreas Peldszus and Manfred Stede. 2013. From Argument Diagrams to Argumentation Mining in Texts: A Survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 7(1):1–31.
- Andreas Peldszus and Manfred Stede. 2015. An annotated corpus of argumentative microtexts. In *Proceedings of the First European Conference on Argumentation: Argumentation and Reasoned Action*, volume 2, pages 801–816.
- Andreas Peldszus and Manfred Stede. 2016. Rhetorical structure and argumentation structure in monologue text. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 103–112, Berlin, Germany. Association for Computational Linguistics.
- Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltasakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn discourse treebank 2.0. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco. European Language Resources Association (ELRA).
- Manfred Stede. 2008. Rst revisited: Disentangling nuclearity. *Subordination ‘versus’ Coordination’ in Sentence and Text*, pages 33–59.
- Manfred Stede, Stergos Afantenos, Andreas Peldszus, Nicholas Asher, and J  r  my Perret. 2016. Parallel discourse annotations on a corpus of short texts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).
- Henning Wachsmuth, Giovanni Da San Martino, Dora Kiesel, and Benno Stein. 2017. The impact of modeling overall argumentation with tree kernels. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2379–2389, Copenhagen, Denmark. Association for Computational Linguistics.
- Douglas Walton, Christopher Reed, and Fabrizio Macagno. 2008. *Argumentation Schemes*. Cambridge University Press, Cambridge.
- Xifeng Yan and Jiawei Han. 2002. gSpan: graph-based substructure pattern mining. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 721–724, Maebashi City, Japan. IEEE.