# COMMIT at SemEval-2016 Task 5: Sentiment Analysis with Rhetorical Structure Theory

**Kim Schouten** and **Flavius Frasincar**
Erasmus University Rotterdam
P.O. Box 1738, NL-3000 DR Rotterdam, the Netherlands

## Abstract

This paper reports our submission to the Aspect-Based Sentiment Analysis task of SemEval 2016. It covers the prediction of sentiment for a given set of aspects (e.g., subtask 1, slot 2) for the English language using discourse analysis. To that end, a discourse parser implementing the Rhetorical Structure Theory is employed and the resulting information is used to determine the context of each aspect, as well as to compute the expressed sentiment in that context by weighing the discourse relations between words. While discourse analysis yields high level linguistic information that can be used to better predict sentiment, the proposed algorithm does not yet stack up to the high-performing machine learning approaches that are commonly exploited for this task.

## 1 Introduction

With sentiment analysis being at the forefront of research, many avenues are explored to find that one new algorithm that will outperform all others. This drive towards excellence is of no surprise given the high practical value this type of algorithms have and the added value they can yield for businesses and consumers alike. This is especially true for aspect-level sentiment analysis, where sentiment scores are assigned, not to a document or sentence, but to the various characteristics, or aspects, of the entity under consideration. Such a fine-grained analysis of, for instance, products or services, can provide many useful insights into consumer thinking.

The majority of the algorithms for aspect-level sentiment analysis is centered around the use of a machine learning classifier and involves tasks such as feature construction and parameter estimation. While the past has shown that this kind of algorithms have strong performance, other directions are also explored (Schouten and Frasincar, 2016). One of these is the concept-driven approach (Cambria et al., 2015), which functions at the semantic level, built upon a layer of natural language processing components. Another direction, which is the research we describe here, is using discourse analysis to improve sentiment classification.

Discourse analysis looks at how the various text segments interact with each other. For example, in *"So even though this laptop is ugly, bulky and way overpriced, I still like it."*, the first half of the sentence gives an explanation for the second part of the sentence. As illustrated in the above example, knowing which parts of the text are ancillary and which parts of the text form the core of the sentence can play a vital role in classifying sentiment. In this case, the sentiment for the laptop in general is positive, even though all of the discussed aspects are negative. Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) describes the various discourse units and their relations, and multiple RST parsers exist that can extract these discourse elements (Marcu, 1997; Surdeanu et al., 2015).

This research shows how one can use RST to classify aspect sentiment. The basic principles of RST are explained in Section 2. The following section showcases some related work on RST, while Section 4 describes how our pipeline is set up. The algorithm



*Proceedings of SemEval-2016*, pages 356–360,
San Diego, California, June 16-17, 2016. 

is evaluated in Section 5, followed by conclusions and future work in Section 6.

## 2 Rhetorical Structure Theory

When performing an RST analysis, the text is first split into clauses, which are called elementary discourse units (EDUs). These clauses form the basic building blocks of the discourse tree. The RST parser then postulates relations between the EDUs, selecting them from a predefined list of discourse relations. There are two basic types of relations: mononuclear and multinuclear relations. The former connects two discourse elements where one element is supporting the other, whereas the latter connects two or more discourse elements that do not have such a clear division of roles. For example, in *'I've only had mine a day but I'm already used to it...'*, the part before 'but' is ancillary and is called the satellite. It supports the second clause, which is called the nucleus, by setting up a contrast. An example of a multinuclear relation is *'It is in the best condition and has a really high quality.'* In this sentence, both elements are nuclei since none of them is specifically supporting the other and they are on the same level instead. EDUs that are linked by a discourse relation together form a new clause that can be linked to another clause again. In this way, a hierarchical structure, which is called the discourse tree, can be formed that spans a whole document. In Figure 1, the discourse tree is shown for the following example.

> 'Being a PC user my whole life, it's taking a bit of time to adapt to the OS of a Mac but I 'm finding my way around.'

## 3 Related Work

One of the first approaches that applies RST to sentiment analysis is (Taboada et al., 2008). This work suggests to rank words within a satellite EDU differently from words that are within a nucleus EDU. Even such a simple use of discourse information already leads to a higher performance of their framework. Note that because only EDUs are used, this method only exploits information from the leafs of the discourse tree.

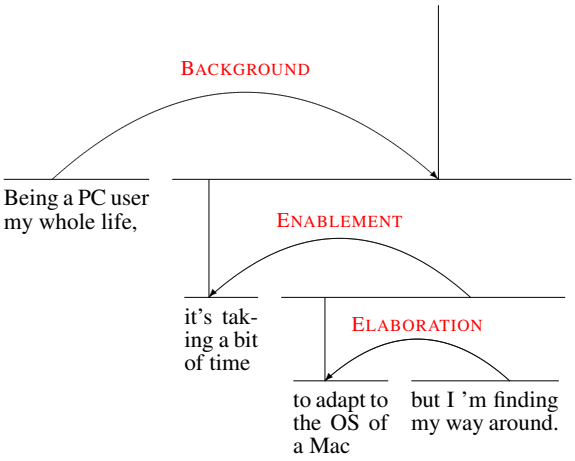


**Figure 1:** Full discourse tree of a simple review sentence (the curved lines denote a mononuclear relation between a nucleus and a satellite).

This is also true for the approach taken in (Heerschop et al., 2011), which uses the same split between words in a nucleus or words in a satellite. Additionally, they take the rhetorical relations these EDUs are in, into account. One of the outcomes of this research is that certain relations are more important for sentiment analysis than others. Hence, words that are in a EDU with an important relation should have a higher influence on the sentiment analysis than others. Another interesting idea was that some EDUs are in a contrasting relation and sentiment scores for words in these EDUs should be negated or flipped. In the described experiments, $F_1$ increased with 15% compared to a baseline because of the exploitation of discourse information. Note that this approach performs sentiment analysis at the sentence level, while we are applying discourse analysis for aspect-level sentiment analysis.

Being closer to aspect-level, the method described in (Zirn et al., 2011) applies RST at the sub-sentence or clause level embedding it inside a Markov Logic Chain. The research mainly focuses on finding relations that negate the sentiment of a certain clause since this has a high impact on the total sentiment score. Compared to a Support Vector Machine baseline, the employed model shows a significant improvement.

## 4 Framework

In order to apply RST for aspect-level sentiment analysis, the discourse context of an aspect has to

be defined first, since this varies from aspect to aspect. Fortunately, we can exploit the discourse tree to select the relevant parts of the text with respect to a given aspect. Knowing the relevant parts of the sentence, the sentiment can be computed by having different weights on the various discourse relations and multiplying the sentiment scores for individual words through the relevant parts of the discourse tree, aggregating the partial scores to one final score that determines the sentiment class.

The first step consists of finding the leaf node(s) in the discourse tree that cover a given aspect. For explicit aspects, the exact position within the sentence is known (i.e., with the `from` and `to` values) , so determining the relevant EDU is straightforward. Sometimes, an aspect spans multiple EDUs. This is the case for some explicit aspects and for all implicit aspects, since the latter do not have a specific target (i.e., `target` is `NULL` and `from` and `to` are zero). When multiple EDUs are returned by this step, the following steps are performed for each EDU, and the final results are aggregated at the end.

Since satellites are complementing the information presented in the nuclei, it stands to reason that, when determining the sentiment of an aspect that is described in a nucleus, we also utilize the information in the satellite that supports that nucleus. On the other hand, when an aspect is described in a satellite, we do not need to include the information of its nucleus since nuclei do not add information to satellites. This information asymmetry helps us to define the context of an aspect. We can look at the discourse tree and, starting at the leaf containing the aspect, move up the tree. As long as we are encountering nuclei we need to go higher because the related satellites need to be included in the context. Hence, as soon as we arrive at a satellite node, that node will be the root of the context tree, since its related nucleus does not need to be included because of the reasons specified before. For the example in Figure 1, since the aspect 'OS' is in a nucleus, we go up to include its elaborating satellite, thereby arriving in a satellite node. Since the first encountered satellite node will be the root node of the context tree, we will not move up the discourse tree any further.

With the context tree available, each word is assigned a sentiment score, using the Stanford Sentiment Tool (Socher et al., 2013). The sentiment values are then combined using Formula 1.

$$sent(s_i) = \sum_{t_j \in s_i} sent(t_j) \times \prod_{r_n \in P_{s_i}} w_{r_n}, \forall s_i \in S_a. \quad (1)$$

where $S_a$ is the set of leaf nodes of the context tree for aspect $a$, $sent(s_i)$ is the sentiment score corresponding to leaf node $s_i \in S$, and $P_{s_i}$ denotes all edges on the path from the root node of the context tree to leaf node $s_i$. Furthermore, $sent(t_j)$ is the sentiment score for word $t_j \in s_i$ and $w_{r_n}$ denotes the weight associated with the rhetorical role of edge $r_n$. These weights are obtained by running a Genetic Algorithm, optimizing for accuracy.

In the final step, all of the $sent(s_i)$ values for a given aspect are added to arrive at a final sentiment score for each aspect, as shown in Formula 2.

$$sent(s_a) = \sum_{s_i \in S_a} sent(s_i) \quad (2)$$

To map the sentiment score of an aspect to a class label, we use a threshold $\epsilon$ to make the distinction between positive and negative classes. As suggested in (Heerschop et al., 2011), we compute the average sentiment score for all aspects that are positive as well as for all aspects that are negative. The $\epsilon$ threshold is then set as the mean of those two values. This helps to avoid the sentiment bias in reviews. Note than in its current form, the algorithm is only capable of assigning a positive or a negative sentiment class to an aspect. Hence all neutral aspects will be misclassified.

**Implementation Notes**

The above framework is implemented using the Stanford CoreNLP (Manning et al., 2014) pipeline, including the already mentioned sentiment component (Socher et al., 2013), together with the CLU-LAB (Surdeanu et al., 2015) discourse parser.

## 5 Evaluation

The evaluation is performed on a previously unseen test set by the organizers of the task. For a description of the used data sets, we refer to the task description paper (Pontiki et al., 2016). The results of this evaluation are shown in Table 1. The proposed algorithm is limited to predicting the senti-

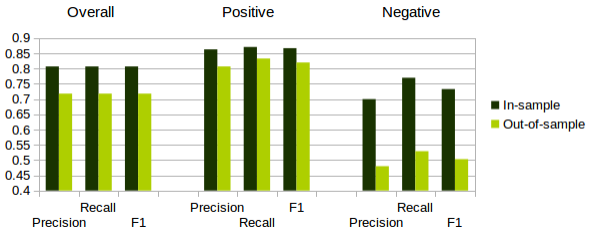


**Figure 2:** Results on the restaurants data, specified per sentiment class.

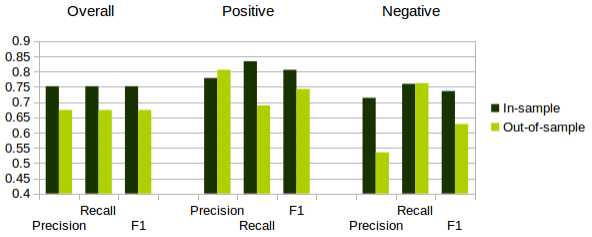


**Figure 3:** Results on the laptops data, specified per sentiment class.

ment value only (i.e., slot 3 for subtask 1) and only operates on English.

| | Restaurants | Laptops |
|---|---|---|
| In-sample | 79.1% | 77.3% |
| 10-fold cross-validation | 76.1% | 71.8% |
| Official test data | 71.8% | 67.5% |

**Table 1:** Results of the proposed algorithm on the two considered datasets. Note that the official results on the restaurant data set are slightly higher than in the official ranking due to a few parsing errors with HTML entities in the submitted file.

As can be seen in Table 1, the algorithm suffers from overfitting, which is caused by optimizing the weights of the various RST relation types. Interestingly, the positive class of the restaurants dataset appears to be unaffected by this overfitting, as its performance on the cross-validated training data is only 2% lower than the in-sample accuracy. This is in stark contrast to the negative class, which suffers a sharp decrease in performance when going from in-sample to cross-validation. However, even comparing the cross-validation results with the test data shows a gap in performance. This can be due to a number of reasons, where it is most likely that the test data is slightly different in terms of language usage than the training data.

Looking at the precision and recall scores for the positive and negative class in Figures 2 and 3, we can see that on the restaurants data, the negative class is poorly predicted, with both a low precision and a low recall. The positive class, however, is performing only slightly worse than on the training data, with the biggest hit taken on the precision. For the laptops data, the situation is different. Here, both the positive and the negative class have a slightly higher recall than precision on the training data. Looking the test data, however, shows that the algorithm predicts too many negative aspects, as the precision for the negative class plummets, but the recall stays roughly the same. For the positive class, the reverse is true: since it only predicts positive in a limited number of cases, the precision is high, but recall is much lower than on the training data.

Negative comments are more difficult to find, as evidenced by the shown performance scores, because of the greater range of expressions used for negative expressions. For example, besides using words with a known negative sentiment, people can express themselves using sarcasm or by describing some aspect and letting the reader conclude that that is sub-par.

## 6 Conclusions

This research shows how Rhetorical Structure Theory (RST) can be used for aspect-level sentiment analysis. Using discourse information when determining sentiment allows certain parts of the text to be stressed while others can be diminished in influence with respect to the sentiment computation. Optimizing the weights of the various relations with a Genetic Algorithm unfortunately leads to overfitting, reducing its effectiveness. Furthermore, we hypothesize that the rather straightforward way of multiplying sentiment through the context tree might be too simplistic. Hence, our suggestions for future work are to limit overfitting of the relation weights and to look for a more sophisticated approach to combining the sentiment scores within the context tree. A good avenue for future research is to investigate the dependence of RST methods on language usage, since the laptop data and restaurant data show such a different result when comparing in-sample and out-of-sample performance. Determining the context tree is also a subject for future research, as

we have not yet tried a full range of possibilities. Last, embedding the discourse information in a classifier like Support Vector Machines is worthy of investigation to see the effect of combining high level linguistic information with the power of statistical inference.

## Acknowledgments

The authors are supported by the Dutch national program COMMIT

## References

Erik Cambria, Soujanya Poria, Federica Bisio, Rajiv Bajpai, and Iti Chaturvedi. 2015. The CLSA model: A novel framework for concept-level sentiment analysis. In *16th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2015) (Part II)*, pages 3–22.

Bas Heerschop, Frank Goossen, Alexander Hogenboom, Flavius Frasincar, Uzay Kaymak, and Franciska de Jong. 2011. Polarity analysis of texts using discourse structure. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM, 2011)*, pages 1061–1070. ACM.

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.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60. Association for Computational Linguistics.

Daniel Marcu. 1997. The rhetorical parsing of natural language texts. In *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics (EACL 1997)*, pages 96–103. Association for Computational Linguistics.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeny Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June. Association for Computational Linguistics.

Kim Schouten and Flavius Frasincar. 2016. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1631–1642. 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 - Human Language Technologies (NAACL HLT 2015): Software Demonstrations*.

Maite Taboada, Kimberly Voll, and Julian Brooke. 2008. Extracting sentiment as a function of discourse structure and topicality. Technical Report TR 2008-20, Simon Fraser University School of Computing Science.

Cäcilia Zirn, Mathias Niepert, Heiner Stuckenschmidt, and Michael Strube. 2011. Fine-grained sentiment analysis with structural features. In *5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*, pages 336–344. Association for Computational Linguistics.