Aspect-Based Sentiment Analysis on the Web using Rhetorical Structure Theory

Rowan Hoogervorst[1], Erik Essink[1], Wouter Jansen[1], Max van den Helder[1], Kim Schouten[1], Flavius Frasincar[1], and Maite Taboada[2]

[1] Erasmus University Rotterdam
P.O. Box 1738, NL-3000 DR Rotterdam, the Netherlands
[2] Simon Fraser University
8888 University Dr., Burnaby, B.C. V5A 1S6, Canada
{369567rh,369723ee,376121mh,357893wj}@student.eur.nl
{schouten, frasincar}@ese.eur.nl, mtaboada@sfu.ca

Abstract. Fine-grained sentiment analysis on the Web has received much attention in recent years. In this paper we suggest an approach to Aspect-Based Sentiment Analysis that incorporates structural information of reviews by employing Rhetorical Structure Theory. First, a novel way of determining the context of an aspect is presented, after which a full path analysis is performed on the found context tree to determine the aspect sentiment. Comparing the proposed method to a baseline model, which does not use the discourse structure of the text and solely relies on a sentiment lexicon to assign sentiments, we find that the proposed method consistently outperforms the baseline on three different datasets.

1 Introduction

Being an integral part of most people's lives, the Web is one of the primary outlets for consumers to express their opinions on products and services they feel engaged with. This engagement can stem from the fact that they purchased a certain product or service resulting in a glowing review of that product or service, but it can also come in the form of an outcry on social media against a product or service because of some shortcoming that prevents the consumer from actually buying it. The willingness to freely share these thoughts and emotions is a driving force behind the success of review sites and review sections on e-commerce sites.

The ubiquity of reviews in e-commerce has proven to significantly affect customer decisions [3], as well as to provide a valuable marketing tool for companies [4]. However, to get a robust overview of a certain product or service, a large number of reviews needs to be covered. This calls for an automated method that performs sentiment analysis on consumer reviews.

For more than a decade [19, 25], many different methods have been developed that aim to automatically extract consumer sentiment from reviews. Over the years, not only has the accuracy of these methods been improved, its level of detail has also increased. Whereas the first methods computed sentiment with respect to the whole review, later methods analyzed the text at a finer level of granularity, such as at the sentence level (e.g., [11]) or even sub-sentence level (e.g., [26]).

In contrast to methods that compute a sentiment value for a given piece of text, Aspect-Based Sentiment Analysis (ABSA) aims to extract sentiment values for a set of aspects (i.e., characteristics or traits) of the product or service being reviewed [22]. Hence, ABSA considers the joint problem of finding aspects (i.e., what characteristics of the entity under review are discussed?) and computing sentiment for each found aspect (i.e., what sentiment is expressed on that particular trait?). The second task also requires that one must identify the exact part of the text that covers a certain aspect. In this research we focus on the second task only, and thus the aspects that are discussed in the text are considered given. This assumption is valid for many review sites in which the aspects of interest are predefined and the user sentiment is gauged for these particular aspects so that products or services can be easily compared.

An interesting new method that has been introduced at the review level [7] and sub-sentence level [29] is that of using discourse relations in the text to compute sentiment values. More specifically, these approaches use a parser implementing the Rhetorical Structure Theory (RST) [15] to find discourse relations in the text, exploiting those to assign weights to the different discourse elements in the text. Hence, parts that are important for the discourse can be emphasized with a high weight, while parts that are less relevant can be diminished with a lower weight. The application of discourse relations can lead to significant performance improvements, as evidenced by the 15% increase in F_1-score reported in [7]. Similar results are reported in [29], where using RST at the sub-sentence level is shown to lead to considerable improvements over a Support Vector Machine baseline.

While the application of discourse analysis for sentiment analysis has been successful at both the review and sub-sentence level, its application has, to the best knowledge of the authors, not been considered for ABSA. Unfortunately, a direct extension of the existing methods to the aspect level is non-trivial due to the fact that aggregating text units is not as natural for aspects as it is for text elements. Instead, a crucial step when applying discourse analysis to ABSA is to define a context for the investigated aspect, mapping the aspect to a certain set of text elements. The work presented in this paper aims to extend the application of discourse analysis to the aspect level, where we focus on finding discourse relations through the application of RST. A main contribution in this respect is a novel way of defining the aspect context based on the discourse relations found through the RST analysis. Furthermore, we suggest how to incorporate this new way of defining the aspect context into a larger framework for ABSA.

The organization of this paper is as follows. In Sec. 2, we consider RST and its current application to sentiment analysis. The main processing framework is introduced in Sec. 3, where we present our method of finding aspect context, as well as the weighting scheme and natural language processing steps involved in computing the sentiments associated to the aspects. We then discuss the actual implementation of our framework in Sec. 4. The performance of the proposed method is evaluated in Sec. 5, and in Sec. 6 conclusions and suggestions for future work are presented.

2 Related Work

While many of the suggested methods for sentiment analysis have proven to be successful [22], a possible deficiency of these traditional methods is that they do not make use of structural elements in text. As has been shown by [1], considering such semantic relations in text may have a positive impact on the sentiment mining task. As a result, we have recently seen the development of different sentiment mining methods that take the discourse structure of text into account. A common characteristic of these methods is that they tend to rely on the use of RST [15] for finding the discourse relations in the text.

To find discourse relations in text, an RST analysis first splits the text into clauses, also called elementary discourse units (EDUs). It then postulates relations between these EDUs, selecting from a list of predefined rhetorical relations. One can distinguish between two types of relations: mononuclear relations and multinuclear relations. In mononuclear relations, the two discourse elements have an unequal status, with one element being the more prominent 'nucleus' and the other being the supporting 'satellite'. An example of a mononuclear relation is:

'I bought this laptop, because it has a good processor.'

In this sentence, the clause after the comma is ancillary (and therefore the satellite) to the first clause (the nucleus), as it gives an explanation to the first part. In multinuclear relations, all elements have the same status and are considered nuclei. An example of such a multinuclear relation is:

'This laptop is neither fast nor does it have large storage capacity.'

Here, none of the elements is more prominent than the other and they are thus both considered nuclei.

An important property of an RST analysis is that EDUs can be aggregated to form new clauses, for which we can then again determine the discourse relation they are in. By proceeding in this way, a complete hierarchical structure of the text is obtained, which can be represented as a tree. In RST, the hierarchical structure of a text is referred to as the discourse tree.

Due to the complexity of RST methods, many researchers thought in the 1990s that discourse analysis could only be performed in combination with fully specified clauses and sentence structures [10, 12]. However, in [17], the first discourse parser that works with unrestricted text is presented. The proposed parser automatically determines the EDUs in unrestricted text as well as the discourse relations between these EDUs, using several lexicographical rules.

Based on the successful application of RST parsers for unstructured text, RST is a natural candidate for use with online consumer reviews and RST has been applied to sentiment analysis in several different ways. One of the first works to apply RST to sentiment analysis is [24], which suggests to rank words that are contained in a satellite differently than those that are contained in a nucleus. When determining whether a word is contained in a satellite or a nucleus, only the leaf level of the discourse tree is considered. Interestingly, this relatively simple split into nuclei and satellites already leads to an improved performance of their suggested sentiment orientation calculator.

This idea is extended by [7], where, although still functioning at the leaf-level of the discourse tree, not just a distinction between nucleus and satellite is used, but also the rhetorical relations between the discourse elements. For this analysis, the eight relations that are most frequently found in the used dataset are considered, out of the 23 rhetorical relations from [15]. One of the findings is that words that are contained in some relation may be of more importance than others, or it may be the case that a relation indicates the presence of a negation. In the given evaluation setup where sentiment is classified per sentence, applying RST leads to a 15% improvement in F_1 score compared to a baseline which does not incorporate the discourse structure of the text. Sentence-level sentiment analysis is also the focus of [27], where discourse information is used to formulate constraints for a Conditional Random Field model. It is shown that these discourse-based constraints are especially important in improving sentiment prediction.

In [29], RST is applied at the sub-sentence level through an application of Markov Logic. The main focus is on dividing the relations into contrasting and non-contrasting relations, as a contrasting relation may potentially negate the sentiment found in an EDU. In an experimental evaluation of the proposed method, a considerable improvement is found, compared to a baseline model without discourse information.

Another method that operates at the sub-sentence level is presented in [28], where sentiment is predicted for each EDU. An interesting part of this research is the comparison of RST with the Penn Discourse Treebank (PDTB). The main conclusions are that RST outperforms PDTB and that methods that include discourse information when predicting sentiment for an EDU outperform the baselines that do not have access to this information.

In [13], sentiment is also predicted for each EDU, however, the authors present a Bayesian model that jointly models sentiment, aspects, and discourse markers. Unfortunately, the method assumes that the overall document sentiment is given, which is not the case in ABSA.

The application of discourse analysis can bring significant improvements to the analysis of sentiment. However, it has not yet been applied to sentiment analysis at the aspect level. In order for that to be possible, it is crucial to find the context of a certain aspect within the text, since the sentiment should be computed from that particular context. Next to the actual sentiment analysis method itself, the proposed method for finding this aspect context is one of the main contributions of this paper.

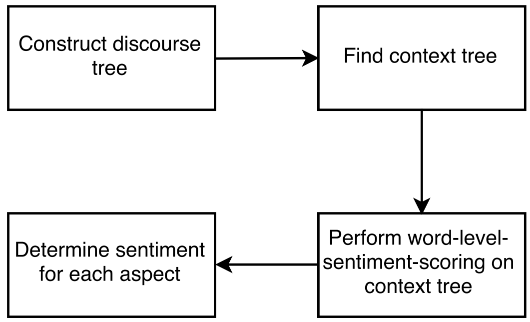

Fig. 1: Outline of framework

3 Framework

In this section we discuss the framework used to find the sentiment with respect to some predefined aspects in online customer reviews. Fig. 1 shows the main steps as proposed in the framework and these steps will be elaborated on one-by-one in the coming subsections.

3.1 Constructing the Discourse Tree

To incorporate the structural relations of a review into the analysis, our method relies on the application of RST to analyze the discourse structure in the review text. Two important elements of such an analysis are to determine the type of discourse parser used and the exact set of rhetorical relations considered in the analysis.

In this work, we propose the use of a document discourse parser to construct discourse trees, which has the advantage over sentence level parsers that one can also take inter-sentence relations into account. Since the context of an aspect is expected to be rather small, the use of inter-sentence relations may be advantageous considering that reviews tend to contain rather informal language in which sentences are short. An example would be:

'I like the speed of this laptop. But starting up is terribly slow.'

In such a sentence the actual aspect relating to the speed of the laptop is in the first sentence, but the very positive sentiment in this first sentence is actually reconsidered to some extent in the second sentence. Hence, to properly find the sentiment relating to the aspect, the inter-sentence relationship is of importance, which confirms the need for a document-level discourse parser.

Furthermore, a subset of the 23 discourse relations, as first introduced by [15], is utilized to analyze the discourse structure of the text. For this work, we choose to use the eighteen most frequently found relations, as identified by [8], instead of the small subset of eight relations used by [7], as we hypothesize that in the supervised setting the framework might still be able to find the correct impact of a relation, even when a relation is not often encountered. Moreover, we expect errors in estimating the impact values for these infrequent relations to have only a minor effect on performance.

3.2 Finding the Context Tree

In order to determine the sentiment associated with a certain aspect, it is important to define the context of that aspect within the text. To that end, a method is proposed that uses the found RST relations to define which parts of the text are covering which aspect. The method starts by finding the leaves in the RST tree that contain aspects. The aspects themselves are given for the data we use. An example of an annotated sentence from the one of the used datasets is given below (including the obvious spelling mistakes, as is common in user generated content on the Web).

```
<sentence id="1028246:1">
  <text>Service was devine, oysters where a sensual as
      they come, and the price can't be beat!!!</text>
  <opinions>
    <opinion target="Service" category="SERVICE#GENERAL"
        polarity="" from="0" to="7" />
    <opinion target="oysters" category="FOOD#QUALITY"
        polarity="" from="20" to="27" />
    <opinion target="NULL" category="RESTAURANT#PRICES"
        polarity="" from="0" to="0" />
  </opinions>
</sentence>
```

For each sentence, the aspects are given together with the position inside the sentence (i.e., with the `from` and `to` values). Some aspects are implicit and do not have a specific target inside the sentence (i.e., `target` is `NULL` and `from` and `to` are zero). In certain cases, including all instances of implicit aspects, the aspect is not contained in a single EDU, but its description is spread over multiple leaf nodes. In such a situation, all leaf nodes related to the aspect are considered separately and the final results are aggregated. The polarity values are empty, since that is the very thing our method predicts for each aspect.

After finding the leaf nodes that contain the aspect, we determine the relevant part of the review based on the fact that satellites tend to be complementary to nuclei. To illustrate this, consider the example:

> 'I think the hard-drive is good, since it has large storage capacity, so I bought it.'

Fig. 2 shows the rhetorical structure of this example. In this example the aspect sentiment related to the aspect of the overall quality of the hard-drive is contained in the EDU 'I think the hard-drive is good', but to fully determine the

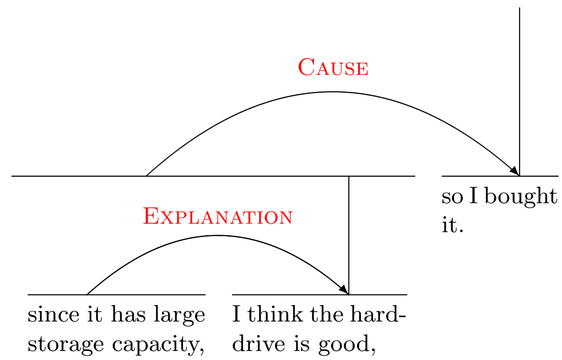

Fig. 2: Full discourse tree of a simple review sentence (the curved lines denote a mononuclear relation between a nucleus and a satellite)

sentiment expressed on the hard-drive we need its corresponding satellite: 'since it has large storage capacity'.

Based on this complementary relation of satellites to nuclei, we argue that a satellite should be taken into account when determining aspect sentiment for the case where the satellite relates back to a nucleus containing the aspect. On the other hand, we argue that nuclei do not add to the understanding of the satellite. Hence, we have a natural definition of context through the asymmetry found in the relation between nucleus and satellite. Relating this back to the discourse tree, it follows that the context defined by this asymmetry is a sub-tree of the original discourse tree. This subtree contains as root the lowest level satellite in which the aspect is contained. More specifically, for the example discussed in Fig. 2, this means that both the satellite and nucleus of the lower level of the tree should be considered, where this nucleus and satellite jointly correspond to the part of the sentence until the junction made by 'so'.

To find the context tree for an aspect, the algorithm looks for the lowest level satellite that contains an aspect. This is done by checking whether the leaf node containing the aspect is a satellite. If this is the case, we stop searching. Otherwise we check whether the parent node (i.e., the linked node one level higher than the leaf node in the discourse tree) is a satellite. If that is not the case, we move up to its parent node and repeat this procedure until we reach a satellite. One can easily verify that this procedure indeed returns the indicated subtree for the example considered in Fig. 2.

In some cases, one aspect can have multiple context trees. Consider the following example for which the discourse tree is given in Fig. 3.

> 'I like this laptop, because it starts up quickly and reacts immediately to my commands.'

In this example, the aspect relating to the speed of the laptop is described in the second part of the sentence 'because it starts up quickly and reacts immediately to my commands'. However, this sub-sentence consists of two leaves, and thus for both leaves a context-tree is found. In this small example, both leaf nodes

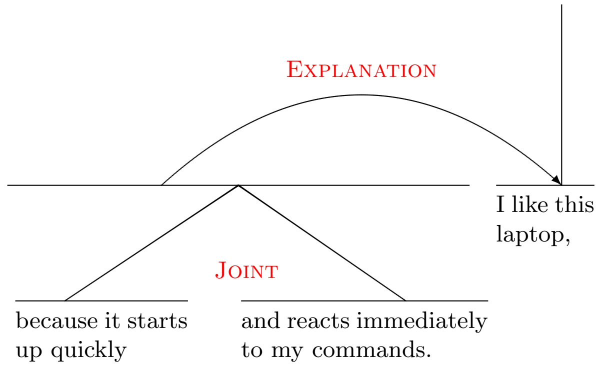

Fig. 3: Full discourse tree for a sentence with multiple context trees (note that the straight lines of the 'Joint' relation denote the fact that this is a multinuclear relation)

will have the same context tree, since they share an immediate parent which is also a satellite. Hence the context tree for both leaves is the tree corresponding to this sub-sentence.

In other cases, the same aspect can be referred to in separate sentences. Since the RST analysis takes inter-sentence relations into account, these cases are naturally dealt with. An example of the RST analysis taking inter-sentence relations into account can be found in the following example, for which the discourse tree is given in Fig. 4.

> 'I was quite hopeful about the speed. However, it is truly atrociously slow, especially when starting up.'

The aspect 'speed' (e.g., of a laptop), is literally mentioned in the first sentence but without expressing any sentiment. In contrast to this, the second sentence expresses a strong negative sentiment on this aspect, but without literally mentioning the aspect again. The contrasting inter-sentence relation is exploited by our method to combine the sentiment from the second sentence with the aspect in the first sentence.

3.3 Performing Word-Level Scoring

After finding the context tree the next step is to determine the sentiment of the different EDUs contained in the context tree. Note that since EDUs get combined further up the tree, we should compute the sentiment for all leaf nodes in the context tree. To determine the sentiment for the leaves we use a sentiment lexicon-based approach that is constructed around the notion of synsets, which correspond to a set of cognitive synonyms.

In this approach all words in the considered reviews are first disambiguated since words can have different meanings in different context. To find a word's

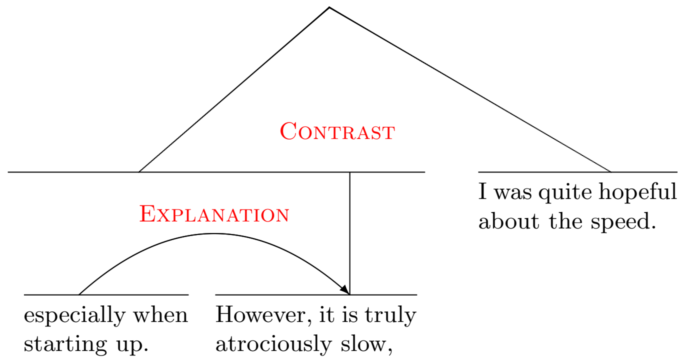

Fig. 4: Full discourse tree showing inter-sentence relation

meaning, its part-of-speech (POS) and lemma are initially collected. As a lexicon can still have multiple entries for a POS of a word, the information regarding the POS of the word is then complemented by its corresponding word sense as determined by the Lesk algorithm [14]. Using the POS, lemma, and word sense of a word, its sentiment score can be obtained from a sentiment lexicon.

3.4 Determining Aspect Sentiment

Combining the output from the previous steps leads to the calculation of the sentiment for each aspect based on its context tree and sentiment score per EDU. This can be done using either the full-path rhetorical structure model or the partial rhetorical structure model. When using the partial rhetorical structure model, only the relations at the leaf level of the discourse tree are utilized, as these are the most fine-grained. In contrast, using the full-path rhetorical structure will allow the use of the path from the root of the context tree to the leaf level. For this work, the full-path rhetorical structure model is employed, as it has been shown to better capture the expressed sentiment [9].

To introduce this method more formally, we use some relevant notation. Let S be the set of leaf nodes of the context tree, $sent(s_i)$ be the sentiment score corresponding to leaf node $s_i \in S$, and P_{s_i} denote all nodes on the path from the root node of the context tree to leaf node s_i. Furthermore, let $sent(t_j)$ be the sentiment score for word $t_j \in s_i$. Last, let w_{r_n} denote the weight associated with the rhetorical role of node r_n. Then, the sentiment score of a leaf node $s_i \in S$ can be computed as

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

In addition to the context tree and the model described above, we also need a weighting scheme which determines for each rhetorical role r_n, the weight w_{r_n}. In our framework we apply the weighting scheme as proposed in [7]. It hypothesizes that relations only differentiate between the importance of different satellites and not of different nuclei. This is intuitively illustrated by the example as introduced in Fig. 2, where it seems plausible that the explanation relation as found in the lower level of the tree does not contribute to the nucleus in the leaf level of the tree. Hence, we obtain a single weight for all EDUs attached to a nucleus, whereas we consider a separate weight for each satellite relation. More formally, with R denoting the set of RST relations, the set of weights is defined as $W = \{(r, \text{Satellite}) | r \in R\} \cup \{\text{Nucleus}\}$. Here, a relation $r_n = (Nucleus, r)\ \forall r \in R$ is assigned the weight that corresponds to the relation $Nucleus$ as considered in W.

The weights are then optimized on training data using a genetic algorithm. Starting with an initial population of random weights, the fittest of these are combined in every step to produce a new set of weights. Moreover, each step mutations occur with a certain probability, to ensure variety in the population, lowering the risk of getting stuck in local optima. The chosen fitness measure is the F_1 value that is obtained when running the algorithm with the given weights.

The last step in our framework is to determine the sentiment for each aspect. To that end, the sum is taken of all the sentiment scores belonging to the context trees that we found to relate to the current aspect. A threshold ϵ is then used to categorize the sentiment score into the positive or negative class. As suggested by [7], the threshold ϵ is set as the mean of (1) the average computed sentiment score for the aspects with positive sentiments and (2) the average computed sentiment score of the aspects with a negative sentiment, to avoid the sentiment bias in reviews. In its current form, the proposed algorithm does not predict neutral sentiment values and is limited to positive and negative only.

4 Implementation

Our implementation is done in the Java programming language, using the SentiWordNet [2, 5] sentiment lexicon, the CLULAB [23] processors for RST parsing, and Stanford CoreNLP [16] for various basic natural language processing tasks.

SentiWordNet is used to assign a sentiment score to each disambiguated word, represented as a WordNet synset. For each synset, three scores are available: objectivity, negativity, and positivity. These three scores always sum up to one, hence when knowing two scores, the third can be inferred. To arrive at a single sentiment score for each synsets, we ignore the objectivity score and subtract the negativity score from the positivity score, resulting in a real number in the $[-1, 1]$ interval [9].

CLULAB is a text-level discourse parser which we use to construct a discourse tree for each review. It is primarily based on the HILDA discourse parser [8] and it uses the same discourse relation set as the discourse parser developed by [6]. This means that CLULAB can be used to parse entire texts, whereas sentence level discourse parsers can only be used to construct discourse trees for separate sentences. To break down the input text into EDUs, also called discourse

Algorithm 1: *findLeaves(tree, aspect)*

```
input  : The discourse tree and aspect under consideration
output: List with all leaves that contain the aspect
// Find leaves that contain aspects
if tree is leaf then
    if tree contains aspect then
        return tree;
    else
        return ∅;
    end
end
aspectLeaves ⟵ ∅;
for all children of tree do
    aspectLeaves ⟵ aspectLeaves ∪ findLeaves(child, aspect) ;
end
return aspectLeaves;
```

segmentation, and find the relations between them, support vector machines (SVMs) are used. To specify the different relations between EDUs, CLULAB uses a discourse relation set consisting of 18 different relations. Examples of the considered relations are attribution, condition, and cause. In addition, it specifies whether the specific relation is multinuclear or mononuclear.

4.1 Finding the Context Tree

After constructing the discourse trees, the algorithm presented in Algorithm 1 finds all leaves that contain the aspect under investigation, using as input the aspect to consider, together with the previously constructed discourse tree. The algorithm starts at the root of the discourse tree and recursively checks all the leaf nodes whether they contain the aspect or part of it. If so, that leaf node is added to the list which is the output of the algorithm.

In the previous step we found all leaf nodes that contain the aspect, irrespective of them being a nucleus or a satellite. In the next step, the context tree of each of the selected leaf nodes is determined based on the asymmetry between nucleus and satellite. The algorithm for this task is given by Algorithm 2. The algorithm starts from the position of a leaf node, after which it iteratively evaluates the role of the parent node of the current node. If the parent node is a nucleus, the parent node becomes the current node and the algorithm moves up one step to evaluate its parent node. This procedure is repeated until the evaluated parent node is either a satellite or the root of the discourse tree. Then the algorithm stops and returns the tree that has that node as its root. Hence, every leaf node will get a context tree that is defined as its closest ancestor, that is a satellite, or the top ancestor (i.e., root of the discourse tree), together with all nodes that share this ancestor.

Algorithm 2: *defineContext(aspectLeaves)*

input : $aspectLeaves$, the set of leaves that contain the considered aspect
output: $contextTrees$, the set of all context trees for the considered aspect

```
// Find closest ancestor that is a satellite
contextTrees ⟵ ∅;
foreach leafNode ∈ aspectLeaves do
    node ⟵ leafNode;
    while hasParentNode(node) and typeOf(parentNode(node)) = Nucleus
    do
        node ⟵ parentNode(node);
    end
    // Context tree, defined by its root node, is added to set
    contextTrees ⟵ contextTrees ∪ {node}
end
```

4.2 Performing Word-Level Scoring

The next step involves assigning scores to all the nodes that have been previously classified as leaf nodes in the context tree. First, punctuation is removed from the text in the EDUs that correspond to the leaf nodes in the context tree. Then the text is split into separate words. After disambiguating to get the meaning of a word, its sentiment score is retrieved from SentiWordNet. Summing up the sentiment scores yields the sentiment score for each EDU that is linked to a particular aspect.

4.3 Determining Aspect Sentiment

The last step of our algorithm is determining the sentiment score for each context tree. For this purpose, we sum the weighted sentiment of the leaf nodes of a context tree (the RST-based weights are here aggregated by multiplication from leaf to root). The pseudocode for this step is presented in Algorithm 3, where this algorithm should be called with the root node of the context tree as *node* and with a $weightNode$ equal to 0. In this algorithm we apply recursion to assign to all leaf nodes a score, which is then weighted as it returns back to the top of the tree. Here we make use of the function $getRelationWeight(node)$, which gives the weight based on the RST relation it is involved in and whether or not this node is a nucleus or satellite.

The last step is now to compare the obtained sentiment score for the aspect to the value of ϵ. If the sentiment score is smaller than ϵ, the returned sentiment for the aspect is negative, otherwise it is positive.

5 Evaluation

This section first introduces the datasets that have been used to evaluate the performance of our proposed methods, after which a comparison of the perfor-

Algorithm 3: *assignScores(nodeTree, weights, weightNode)*

input : the context *tree* under consideration represented by a node, the *weights* for the RST relations, the weights so far built up in the tree *weightNode*
output: The sentiment score of the context tree
$currentWeight \longleftarrow weightNode$;
if *node is not the root node of the context tree* **then**
  $currentWeight \longleftarrow currentWeight * getRelationWeight(node)$;
end
if *node is a leaf node* **then**
  score $\longleftarrow$ the sentiment for this EDU;
  return $score * currentWeight$;
end
$newScore \longleftarrow 0$;
foreach *child* **of** *node* **do**
  $newScore \longleftarrow newScore + assignScores(child, weights, currentWeight)$;
end
return *newScore*

mance of these methods is made. Additionally, this section introduces a baseline method which we benchmark our method against.

5.1 Data Description

In the evaluation we used three sets of reviews. All of these datasets are from the SemEval ABSA Task [21, 20]. We consider here the dataset on restaurant reviews of the SemEval 2014 and the datasets on restaurant and laptop reviews of the SemEval 2015. For the aspects considered for the dataset about laptops, one can think of among others the battery, the screen, and the hard-drive. For the restaurant datasets, aspects relate to for example the atmosphere, taste of the food, and level of service.

An important difference between these datasets is that the SemEval 2014 dataset consists of only single sentence reviews, while the SemEval 2015 datasets consider reviews with multiple sentences. Furthermore, the size of the datasets differs as the restaurants dataset of 2014 contains a total of 3041 reviews and a total of 3693 aspects. The dataset of laptops contains instead 277 reviews, 1739 sentences and 1974 aspects, while the dataset of restaurants of the SemEval 2015 contains a total of 254 reviews, 1315 sentences, and 1654 aspects. The datasets contain mostly positive and negative aspects, but a small number of aspects are annotated as neutral or conflicted. Since our method only predicts positive or negative, the performance with respect to these classes is not reported. Note that for the overall performance measures, these instances are considered incorrect classifications and hence are included in the computation of the performance measures.

Table 1: Performance for the laptops 2015 dataset

(a) Performance of baseline method

Precision	Recall	F_1

(b) Performance of proposed method

	Precision	Recall	F_1
Overall	0.67	0.67	0.67
Positive	0.67	0.88	0.76
Negative	0.69	0.47	0.56

5.2 Baseline Method

The acquired results are compared against a baseline model in order to evaluate the performance of the suggested method. In this case the baseline method considers a simple natural language processing approach that employs SentiWordNet and does not account for the discourse structure of the text. For each aspect we consider a fixed size context window of one to three words around the aspect. For aspects that do not have a specific target in the sentence but are implicit, we consider the whole sentence as the context window. Let the set of the words in this context for aspect i be given by c_i, then the baseline method computes the sentiment for aspect i as

$$\text{sent}(i) = \sum_{t_j \in c_i} \text{sent}(t_j). \tag{2}$$

where t_j are the words in the current review j in which i is an aspect. Similar to the proposed method, the sentiment score for each word is retrieved from SentiWordNet.

5.3 Comparison of Methods

The proposed method is evaluated on three datasets using the common 10-fold cross-validation technique. The performance of the baseline and the proposed method are presented in Table 1a and Table 1b for SemEval 2015 laptop data. Since this data set does not provide any location within the sentence for aspects, the context window for the baseline is irrelevant.

These tables clearly show that the proposed method outperforms the baseline. An important observation is that the performance is somewhat lower for negative aspects than for positive ones. A possible explanation for this lower performance on negative polarities might be found in the fact that negative reviews are often written with many positive words [18]. However, using discourse information

Table 2: Performance for the restaurants 2015 dataset

(a) Performance of baseline method

	Context window = 1			Context window = 2			Context window = 3		
Category	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
Overall	0.57	0.57	0.57	0.54	0.54	0.54	0.50	0.50	0.50
Positive	0.70	0.74	0.72	0.69	0.67	0.68	0.68	0.61	0.64
Negative	0.15	0.15	0.15	0.16	0.20	0.18	0.16	0.23	0.19

(b) Performance of proposed method

	Precision	Recall	F_1
Overall	0.74	0.74	0.74
Positive	0.80	0.86	0.83
Negative	0.52	0.47	0.49

mitigates this issue to some extent, as the proposed method is considerably less sensitive to this phenomenon than the baseline.

Table 2a and Table 2b show the results for the baseline method and the proposed method for the SemEval restaurant 2015 reviews, respectively. The results found here confirm to a large extent the observations made for the previous dataset. Again, the proposed method outperforms the baseline model.

Last, Table 3a and Table 3b show the performance of both methods on the SemEval restaurants 2014 dataset. These tables show that the performance of our method is lower than for the other two datasets. A possible reason for the decrease in performance is that this dataset only considers single sentence reviews. As we apply RST analysis at the review level, this implies that we can not use inter-sentence relations in this dataset, as done for the restaurant 2015 dataset. However even in this scenario where the RST analysis cannot be used to its full potential, it is still the better option, yielding higher performance than the baseline method, both for negative and positive aspects.

6 Conclusion

While the application of Rhetorical Structure Theory (RST) in sentiment analysis has already been proven to obtain good performance at higher levels of text granularity, the method has not yet been explored for Aspect-Based Sentiment Analysis (ABSA). For this reason we propose a framework that uses RST for ABSA. In this framework, discourse trees are first created by using a document level discourse parser. The main contribution of this paper is the definition of the aspect context through constructing a context tree based on the asymmetrical relation between satellites and nuclei in RST. This context tree is used for the

Table 3: Performance for the restaurants 2014 dataset

(a) Performance of baseline method

	Context window = 1			Context window = 2			Context window = 3		
Category	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
Overall	0.50	0.50	0.50	0.47	0.47	0.47	0.43	0.43	0.43
Positive	0.56	0.83	0.67	0.55	0.74	0.63	0.54	0.67	0.60
Negative	0.12	0.07	0.09	0.15	0.15	0.15	0.15	0.20	0.17

(b) Performance of proposed method

	Precision	Recall	F_1
Overall	0.60	0.60	0.60
Positive	0.64	0.91	0.75
Negative	0.42	0.32	0.36

sentiment scoring, for which we use sentiment lexicon-based word-scoring and a full-path based rhetorical structure processing scheme.

To evaluate the performance of the proposed framework a comparison between the suggested methods and a baseline model that does not incorporate the discourse tree of the review is made. The comparison of the performances of the proposed method to the benchmark is made on the basis of three different datasets, comprised of laptop and restaurant reviews. We find that for all three datasets, the baseline model is clearly outperformed, but the proposed method seems susceptible to negative reviews that use many positive words.

Based on the success of using RST for ABSA as reported in this paper, a next step would be to extend the proposed methodology with additional components. An interesting avenue of research would be to incorporate the found discourse structure and context tree into a classification algorithm such as a Support Vector Machine. This combines the raw power of machine learning with the finesse and detail of discourse analysis.

Acknowledgments

The authors of this paper are supported by the Dutch national program COMMIT.

References

[1] Asher, N., Benamara, F., Mathieu, Y.Y.: Appraisal of opinion expressions in discourse. Lingvisticæ Investigationes 32(2), 279–292 (2009)

[2] Baccianella, S., Esuli, A., Sebastiani, F.: SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010). vol. 10, pp. 2200–2204 (2010)

[3] Bickart, B., Schindler, R.M.: Internet forums as influential sources of consumer information. Journal of Interactive Marketing 15(3), 31–40 (2001)

[4] Chen, Y., Xie, J.: Online consumer review: Word-of-mouth as a new element of marketing communication mix. Management Science 54(3), 477–491 (2008)

[5] Esuli, A., Sebastiani, F.: SentiWordNet: A publicly available lexical resource for opinion mining. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006). vol. 6, pp. 417–422. European Language Resources Association (ELRA) (2006)

[6] Feng, V.W., Hirst, G.: Text-level discourse parsing with rich linguistic features. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012). pp. 60–68. Association for Computational Linguistics (2012)

[7] Heerschop, B., Goossen, F., Hogenboom, A., Frasincar, F., Kaymak, U., de Jong, F.: Polarity analysis of texts using discourse structure. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM, 2011). pp. 1061–1070. ACM (2011)

[8] Hernault, H., Prendinger, H., duVerle, D.A., Ishizuka, M., et al.: HILDA: a discourse parser using support vector machine classification. Dialogue and Discourse 1(3), 1–33 (2010)

[9] Hogenboom, A., Frasincar, F., de Jong, F., Kaymak, U.: Using rhetorical structure in sentiment analysis. Communications of the ACM 58(7), 69–77 (2015)

[10] Kamp, H., Reyle, U.: From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory, vol. 42. Springer Science & Business Media (1993)

[11] Kim, S.M., Hovy, E.: Determining the sentiment of opinions. In: Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004). Association for Computational Linguistics (2004)

[12] Lascarides, A., Asher, N., Oberlander, J.: Inferring discourse relations in context. In: Proceedings of the 30th Annual Meeting on Association for Computational Linguistics (ACL 1992). pp. 1–8. Association for Computational Linguistics (1992)

[13] Lazaridou, A., Titov, I., Sporleder, C.: A bayesian model for joint unsupervised induction of sentiment, aspect and discourse representations. In: Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (ACL 2013). pp. 1630–1639. Association for Computational Linguistics (2013)

[14] Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In: Proceedings of the Fifth Annual International Conference on Systems Documentation (SIGDOC 1986). pp. 24–26. ACM (1986)

[15] Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. Text-Interdisciplinary Journal for the Study of Discourse 8(3), 243–281 (1988)

[16] Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 55–60. Association for Computational Linguistics (2014)

[17] Marcu, D.: The rhetorical parsing of natural language texts. In: Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics (EACL 1997). pp. 96–103. Association for Computational Linguistics (1997)
[18] Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2008)
[19] Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: Sentiment classification using machine learning techniques. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002). pp. 79–86. Association for Computational Linguistics (2002)
[20] Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., Androutsopoulos, I.: Semeval-2015 task 12: Aspect based sentiment analysis. In: Proceedings of the Ninth International Workshop on Semantic Evaluation (SemEval 2015). pp. 486–495. Association for Computational Linguistics (2015)
[21] Pontiki, M., Papageorgiou, H., Galanis, D., Androutsopoulos, I., Pavlopoulos, J., Manandhar, S.: Semeval-2014 task 4: Aspect based sentiment analysis. In: Proceedings of the Eighth International Workshop on Semantic Evaluation (SemEval 2014). pp. 27–35. Association for Computational Linguistics and Dublin City University (2014)
[22] Schouten, K., Frasincar, F.: Survey on Aspect-Level Sentiment Analysis. IEEE Transactions on Knowledge and Data Engineering 28(3), 813–830 (2016)
[23] Surdeanu, M., Hicks, T., Valenzuela-Escárcega, M.A.: Two practical rhetorical structure theory parsers. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT 2015): Software Demonstrations (2015)
[24] Taboada, M., Voll, K., Brooke, J.: Extracting sentiment as a function of discourse structure and topicality. Tech. Rep. TR 2008-20, Simon Fraser University School of Computing Science (2008), `ftp://fas.sfu.ca/pub/cs/TR/2008/CMPT2008-20.pdf`
[25] Turney, P.D.: Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2002). pp. 417–424. Association for Computational Linguistics (2002)
[26] Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT 2005). pp. 347–354. Association for Computational Linguistics (2005)
[27] Yang, B., Cardie, C.: Context-aware learning for sentence-level sentiment analysis with posterior regularization. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014). pp. 325–335. Association for Computational Linguistics (2014)
[28] Zhou, Y.: Fine-grained Sentiment Analysis with Discourse Structure. Master's thesis, Saarland University, Germany (2013), `http://lct-master.org/getfile.php?id=783&n=1&dt=TH&ft=pdf&type=TH`
[29] Zirn, C., Niepert, M., Stuckenschmidt, H., Strube, M.: Fine-grained sentiment analysis with structural features. In: 5th International Joint Conference on Natural Language Processing (IJCNLP 2011). pp. 336–344. Association for Computational Linguistics (2011)