Use of Discourse Knowledge to Improve Lexicon-based Sentiment Analysis

Pedro Paulo BALAGE FILHO
Research Group in Computational Linguistics, University of Wolverhampton, United Kingdom
Spoken Language Laboratory, INESC ID Lisboa, Universidade do Algarve, Portugal

Abstract

Sentiment Analysis deals with the computational treatment of sentiment in texts. Discourse is a linguistic level of analysis where the author represents ideas and links concepts in a rational chain of thoughts. One important representation of discourse is the Rhetorical Structure Theory (RST). The objective of this work consists in to use discourse knowledge to improve a lexicon-based sentiment classifier. To achieve this goal it presents a lexicon-based algorithm adapted to weight portions of text under particular RST relations distinctly. Two experiments are reported. The first experiment verifies if the RST improves sentiment classification. It also shows the discourse relations which are most important in the process. The second experiment incorporates discourse markers in the algorithm in order to eliminate the necessity of a RST annotated corpus. It uses the weights learned in the first experiment to perform the sentiment classification.

Key-words

Sentiment Analysis; Lexicon-based Sentiment Analysis; Discourse; Rhetorical Structure Theory.

1. Introduction

In the age of information, the ability to access, retrieve and process data is of vital importance. According to Lyman and Varian (2003), the world produced in 2003 between one and two exabytes of unique information. Eric Schmidt, executive chairman from Google, affirmed that, in 2010, every two days we create as much information as we did from the dawn of civilization

up until 2003. According to him, it is something like five exabytes of data each day, and most of its content is user-generated [1]

In face to the unquestionable grow of information produced by internet users, it remains a challenge to organize and extract useful information from this content. All this produced information has become of great interest to companies interested in following the reputation of their services or products. They are increasingly following product mentions through blogs, social networks and product reviews.

On the other hand, users are also frequently demanding more information about products and companies in order to buy a new product or service. Websites for product reviews have become an important resource to find opinions and influence users (Bailey, 2005).

Due to the importance of processing all this content, there is a natural necessity to study and understand how to deal with opinions or sentiments in text. The goal of sentiment analysis is to provide analysis of the sentiments present in documents. Sentiment analysis, also known as opinion mining, is a relatively new research topic in computational linguistics that addresses the problem of understanding opinionated texts.

In a document, sentiment can be expressed in different ways. It can be classified in function of the existence of sentiment, i.e., it is either polar or neutral. It can be categorized as positive or negative. Some authors also consider the six “universal” emotions (Ekman et al., 1982): anger, disgust, fear, happiness, sadness, and surprise. This paper approaches sentiment in the positive and negative categories.

This work focuses in a particular aspect of sentiment analysis. In text with sentiment, it is usual for the author to include expectations and coherent ideas in the discourse level. This work aims to identify and aggregate such information to be provided to a sentiment classifier.

The use of discourse structure to represent ideas is evident in text with sentiment. Sentiment classifiers can use such structure to better understand the text and emphasizes what is more important. The aim of this work is to improve lexicon-based sentiment analysis using the discourse structure.

Lexicon-based sentiment analysis is an approach to sentiment classification where a dictionary of sentiment words is applied to determine if a text is positive or negative.

In this study, discourse structure is analysed by the Rhetorical Structure Theory (RST) (Mann, 1987) discourse framework. In this theory, the author intentions are organized into discourse relations which can be determined in the text. The goal of this work is to show how discourse can be detected, shaped and adjusted in order to improve a lexicon-based sentiment classifier.

This document is structured as follows: Sections 2 and 3 show the main concepts in sentiment analysis and RST theory, Section 4 shows related works, Section 5 presents the SO-RST algorithm defined in this study, Section 6 presents the experiments and, finally, Section 7 concludes.

2. Sentiment Analysis

Sentiment analysis or opinion mining deals with the computational treatment of opinion, sentiment and subjectivity in text (Pang et al., 2002). In a broad way, sentiment analysis can be seen as a document classification task where an algorithm needs to classify a text based on the sentiment it contains.

Sentiment classification can be decomposed in three different levels of analysis: feature level, sentence level or document level. Feature-level sentiment analysis determines the polarity of the sentiment expressed over a particular feature or product. Sentence-level sentiment analysis deals with the sentiment classification at the sentence-level. Document-level sentiment analysis aims to classify documents based on the sentiment expressed in the whole document. In this level, the task corresponds to analysing the text in a coherent way.

Sentiment classifiers have two basic approaches: lexicon-based method and the machine learning method. The lexicon-based method uses a dictionary of terms and their respective polarities, also known as semantic orientations. This method computes the polarity of a document, sentence or feature based on the number of positive or negative terms in the text. The machine learning approach can be supervised or unsupervised. Supervised machine

learning uses a training corpus with labelled examples to learn the domain lexicon for each sentiment class in order to build a classification model. The unsupervised machine learning uses an unlabelled corpus to compute by similarity a set of features for the sentiment classes.

The Lexicon-based method is known for being domain-independent, while the machine learning method tends to adapt to the domain that the classifier learns. Also, the lexicon-based method does not require a corpus of training, only a dictionary of semantic orientations, which is useful for new domains or topics when we do not have a corpus available. On the other hand, machine learning classification is known as better for discovering hidden sentiment vocabulary specific of the training domain. In this sense, machine learning methods can achieve higher accuracy when compared with lexicon-based methods in specific domains (Aue and Gamon, 2005) (Pang and Lee, 2008, section 4.4).

Although both methods exhibit particular advantages and disadvantages, it is recognized a better ability of lexicon-based methods to incorporate and analyse new linguistic features (Taboada et al., 2011). It is simpler for a lexicon-based method to change the semantic orientation of the words in a sentence when linguistic phenomena are found. As a result, this work uses a lexicon-based method in our sentiment classification.

The lexicon method is based on the same linguistic concept used by the reader when it assesses a text (Taboada et al., 2011). In this method, a classifier can simply averages the semantic orientations found in the text, or it can use a full linguistic analysis (one that involves analysis of word senses or argument structure).

The most important lexicon-based method is reported by Taboada et al. (2011). The authors describe experiments with the Semantic Orientation CALculator (SO-CAL) (Taboada et al., 2006; Taboada and Grieve, 2004), a system to measure the semantic orientation of a text. Their work takes two assumptions: (a) individual words have a prior polarity, which is independent from context; (b) the semantic orientation can be expressed as a numerical value.

Taboada et al. (2006) report a method to build a semantic orientation dictionary using adverbs, adjectives, nouns and verbs. The dictionary consists in semantic orientation values assigned to words in a scale of -5 to 5, where -5 stands for totally negative and 5 for totally positive.

For the process of building the dictionary and the SO-CAL system they used the SFU Review Corpus (Taboada et al., 2006; Taboada and Grieve, 2004). This corpus is a collection of 400 reviews from the website Epinions.com extracted from eight different categories: books, cars, computers, cookware, hotels, movies, music, and phones. Within each collection, the reviews were split into 25 positive and 25 negative reviews, for a total of 50 in each category.

The SO-CAL algorithm can be summarized as follows:

i. Load the dictionary with the semantic orientation for the words (adjectives, verbs, nouns and adverbs)
ii. If an intensifier is found in the text, increase or decrease in a determined scale the semantic orientation for the next polar word.
iii. If a negation marker is found in the text, shift by 4 the semantic orientation of the next polar word.
iv. If a modal verb is found in the text, change the semantic orientation of the next polar word to 0 (neutral).
v. All polar words are computed and their sum is divided by the number of sentences. This value is the semantic orientation for the text.
vi. If the text semantic orientation is above a threshold, the text is positive, otherwise it is negative.

3. RST

Discourse is a linguistic level of analysis where the author represents his intentions in a rational logic chain of thoughts. In a general way, different aspects of the discourse are shaped by different discourse theories. Discourse theories are ways to explain and structure the discourse.

Rhetorical Structure Theory (RST) is a descriptive theory proposed by Mann (1987) that explains the use of rhetorical relations in the text in order to keep the coherence. RST defines relations between text spans, which are the minimum unities of discourse, also known as Elementary Discourse Unities (EDUs) (Mann and Thompson, 1988; Taboada and Mann, 2006).

The theory is organized under twenty six relations that link text spans in a tree structure. Each relation links two spans of text in terms of the intentions desired by the author in the discourse level.

For some relations, the linked segments can assume the functions of nucleus or satellite. The nucleus is the most relevant segment of text, the one in which the relation is based. The satellite is the weak element in the relation, the one who derives the relation. A nucleus can be sustained in the text without the satellite, but the opposite is not true. Some relations do not present a satellite and then they have both nucleus. These relations are called multi-nuclear.

In the literature, one can find some automatic RST parsers for several languages (Marcu, 2000; Pardo and Volpe Nunes, 2008; Subba and Di Eugenio, 2009). In the process of construction, a RST parser is built with a specific domain in mind. For reviews domain, in the best of your knowledge, there is no RST parser available.

4. Previous Works

A first work to argue the importance of the discourse structure for sentiment analysis is described by Polanyi and Zaenen (2006). This theoretical work shows how some contextual valence shifter can change the natural semantic orientation of the words.

Pang et al. (2002) included the information where each word is located in the feature set for a machine learning method. Specifically, the position where the tokens appear demonstrated to improve the classification, also verified by Taboada et al. (2011).

Pang and Lee (2004) observed that the position has influence in the context of summarizing sentiment in a document. In contrast with topic-based text summarization, where the beginnings of articles usually keep the main information about the topic, the last sentences of a review have been shown to express the relevant opinion in the text. Theories of lexical cohesion motivate the representation used by Devitt and Ahmad (2007) for sentiment polarity classification of financial news.

Taboada et al. (2008) proposes a combination of local and global information in the determination of semantic orientation. They use the discourse structure and the topicality to improve the sentiment classification accuracy for the SO-CAL algorithm. Their approach consists in assigning extra-weight to the semantic orientations for the most relevant sentences in the text. They use two different approaches. The first approach uses the discourse structure via Rhetorical Structure Theory and extracts the nuclei as the relevant part. The second approach uses a support vector machines classifier to extracts the most relevant topic sentences from text. The best results were achieved when the relevant sentences were multiplied by a factor of 1.5 while the irrelevant by a factor of 0.5. They showed that the use of weights on relevant sentences leads to an improvement over word based methods that consider the entire text equally. The methods showed an increase in the overall performance from 72% (SO-CAL) to 80.00% (RST) and 80.67% (Topicality) for the SFU Review Corpus (Taboada et al., 2006; Taboada and Grieve, 2004).

Somasundaran (2010) presents a complete study about the use of discursive knowledge in sentiment analysis. She uses discursive knowledge and machine learning classifiers for recognizing stances in dual-sided debates from the product and political domains. For product debates, she uses web mining and rules to learn and employ elements of discourse-level relations in an unsupervised fashion. For political debates, she uses a supervised approach to encode the building blocks of discourse-level relations as features for stance classification. Her results show that the discourse-level relations can enhance and improve upon word-based methods.

5. SO-RST

As described in section 2, lexicon based methods are useful to incorporate new linguistic features in the classifier algorithm. We have showed the algorithm SO-CAL (Taboada et al., 2011), which simply computes the semantic orientation of the words present in the text based in a sentiment dictionary.

The SO-RST algorithm presented in this work is an adaptation of the SO-CAL algorithm, which was modified to take in account the RST structure of the text. Each relation in RST is defined in terms of discourse unities,

denominated Elementary Discourse Unities (EDUs) or spans. The majority of relations presents a nucleus span, responsible for the main discourse content, and a satellite span, responsible to the relation developed from the nucleus. The approach taken by this work is to assign a distinct weight or importance for each RST relation. Using RST structure, our algorithm aims to give a higher or lower importance to RST spans and consequently improve the classification. To illustrate our algorithm, please consider the following example.

(1) I like the product appearance. One day, it broke down. Hence, I believe it is a bad product.

In the Example 1, the first sentence does not belong to any RST scope, so we say it presents the virtual relation “None”. The second and third sentences have a Result relation. The sentence 2 is defined as nucleus of such relation while sentence 3 is the satellite.

In our algorithm we consider a factor which multiplies the semantic orientation of each polar word under the scope of a RST relation. We named this factor as a weight w_i which is covered by the relation i. The Example 2 shows how the weights will be assigned.

(2) I like(+4) the product appearance.

$SO1 = 4 \times w_{none}$

One day it broken(-2) down.

$SO2 = -2 \times w_{ResultNucleus}$

Hence, I believe it is a bad(-2) product.

$SO3 = -2 \times w_{ResultSatellite}$

$TotalSO = SO1 + SO2 + SO3$

Like the original SO-CAL, the algorithm classifies the text based on the average of the semantic orientation computed. We based our experiment in the work reported by Taboada and Grieve (2004), where the SO-CAL was

used with a threshold of 0.62. We also use the same dictionary of sentiment provided by Taboada and Grieve (2004).

The evaluation of our classifier algorithm is based on the amount of instances correctly classified. In this work we adopt accuracy as the evaluation measure. The Figure 1 shows a diagram for the SO-RST algorithm proposed in this work.

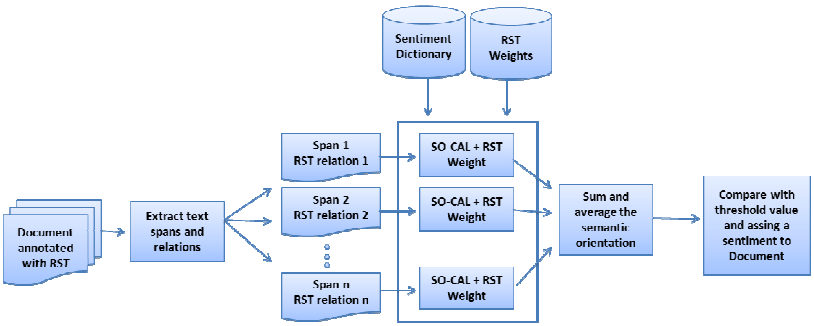

Figure 1. Diagram showing the steps followed by the SO-RST algorithm

The algorithm input is a document annotated with RST. In this document, text spans are linked through RST relations. The Algorithm extracts these spans and the RST relation they encompass. In this extraction, only the RST relations which linked leaves in the RST tree are considered. Each span extracted is sent to calculate the semantic orientation for the words present. The semantic orientation calculator is adapted from the SO-CAL with an extra weight if a word is under the influence of a RST relation.

After to calculate the semantic orientation for all sentences, the algorithm computes an averaged sum and compares this value against a threshold to classify the text as positive or negative.

In order to test our hypotheses and learn how to weight each particular relation we conducted two major experiments described in the next section.

6. Experiments and Results

The first experiment conducted aims to find the best configuration of weights which maximizes the accuracy of the SO-RST algorithm described. In sum, we want to learn which RST relations are important in a lexicon-based sentiment analysis algorithm and which ones are not.

For this, we used the SFU Review Corpus annotated with RST (Taboada et al., 2006; Taboada and Grieve, 2004). The corpus provides the annotation at the sentence level, i.e., only the relations found within sentences were annotated. In average, each text contains 24 sentences and 698 words. The corpus version with RST annotation presents, per text, in average, 55 spans and 15.19 RST relations.

The second experiment designs and incorporates a shallow RST parser in the algorithm. The experiment objective is far from designing and implementing a full RST parser for the reviews domain. Our method focuses on identifying shallow RST relations in the text, evidenced by discourse markers and word clues. The experiment focuses on the relations that helped achieving a good average accuracy in the first experiment and explore how to incorporate those relations in the algorithm.

The next subsections describe in detail both experiments.

6.1 Identifying the Best Weights

This first experiment uses machine learning techniques to learn from a RST annotated corpus. The experiment splits the corpus into four folds, equally distributed among the categories and sentiment classes. Each of these four folds is going to be used to perform a cross-validation and, in the end, the average accuracy is computed. This process is required in order to train and test the algorithm with different portion of data, which assures that the average result is not biased for any particular set of texts present in the corpus. In this experiment, the four-folds cross validation performs the learning process 4 times. Each time, a third part of the corpus is used for training and the remaining part of the corpus is used for testing. In these four times, distinct parts of the corpus are used for testing ensuring the

uniformity of the results. To test the weights learned in the learning step we simply apply the SO-RST algorithm described in the previous section.

In the learning process, it is infeasible to compute the best weights by simply testing every possible combination. For example, if we wish to learn how to weight the 26 relations present in the RST theory with the values 0 or 1, we would have 226 different possibilities, which is approximately 68 million of combinations. Due the impractical possibility of this experiment by a brute force method, this work appealed to a heuristic method. The method adopted is a genetic algorithm technique, which is able to achieve a solution closer to the optimal solution without the necessity to test all combinations.

Our experiment was initialized with random values and configured with a population of size 40, i.e., in each generation 40 different configurations of weights are tested and the programs which achieve the higher accuracies are more susceptible to have their weights propagated to the next generation. The experiment computed 100 generations and returned the set of weights, identified by relation, for the program with the highest accuracy verified among all generations.

In this experiment we have two main goals. The first is to verify, by the best weight assigned, how useful a particular relation is for sentiment analysis classification. The second goal is to verify if the weights optimized for the training set, when applied in the testing set instances, lead to a better accuracy.

In order to best cover the adequacy of RST theory to lexicon-based sentiment classification, we configured our experiment in two scenarios. In the first scenario we used the same weight for the nucleus and satellite span under the relation (no distinction between nucleus and satellite). In the second scenario, for each relation, we use different weights for the nucleus and the satellite spans.

Inside each scenario we have also two ways to apply the weights. The first method receives binary weights (0 or 1), i.e., the words under those relations are included or not in the compute of the text semantic orientation. In the

second method, each relation is multiplied by a real number ranging from 0 to 5.

Our two scenarios (weight the whole span, or weight satellite and nucleus distinctly) combined with the two methods (binary or real weights) for each, resulted in 4 different experiments and results. In order to compare the improvement achieved by each experiment, we used a baseline algorithm. In the baseline, the algorithm provides a classification without taking in account the RST structure (we assign weight 1 for each relation).

To ensure the representativeness of the experiments, we only apply weights for those relations which show enough evidence in the corpus. In this study we only use the relations which have a minimum frequency of 30 instances. It is in our judgement that relations with the frequency less than 30 instances will not provide representative results. All the relations chosen are mono-nuclear (present nucleus and satellite spans).

The results obtained by the two tested scenarios are shown in the Table 1 and Table 2.

In the training set for the scenario 1, the values show that the learning algorithm improved the average accuracy in the heuristic process to determine the best weights. Using binary weights the average accuracy for the training set was 73.50% against 72.00% from the baseline (Table 1a). Using real weights the average accuracy for the training set was 78.50% against 72.00% from the baseline (Table 1b). These results demonstrate that the learning algorithm achieved its goal and determined which weights maximize the accuracy measure.

In the test set for the scenario 1, the average accuracy using binary weights was 71.25% (Table 1a) and the average accuracy using real weights was 75.75% (Table 1b). The baseline accuracy for both was 72.25%. The conclusion is that the learned weights improved the average accuracy when real values were assigned. The same was not verified when binary weights were used. The values reported were submitted to a two-sample student t-test and their proved to be statistically significant ($P < 0.5$).

Table 1. Accuracy measure for cross-folding validation with the weights learned by the genetic algorithm for the Scenario 1

a) Binary Weights

		1st Fold	2nd Fold	3rd Fold	4th Fold	**Average**
Train Set	Baseline	72.67%	72.67%	71.00%	71.67%	**72.00%**
	Experiment	74.33%	73.67%	71.67%	74.33%	**73.50%**
Test Set	Baseline	71.00%	71.00%	76.00%	71.00%	**72.25%**
	Experiment	70.00%	71.00%	75.00%	69.00%	**71.25%**

b) Real Weights

		1st Fold	2nd Fold	3rd Fold	4th Fold	**Average**
Train Set	Baseline	72.67%	72.67%	71.00%	71.67%	**72.00%**
	Experiment	78.33%	80.00%	77.00%	78.67%	**78.50%**
Test Set	Baseline	71.00%	71.00%	76.00%	71.00%	**72.25%**
	Experiment	75.00%	72.00%	82.00%	74.00%	**75.75%**

In the second scenario (nucleus and satellite spans weighted separately) the learning algorithm was also able to improve the average accuracy in the training set. Using binary weights, the average accuracy for the training set was 74.25% (Table 2a). Using real weights, the average accuracy for the training set was 78.92% (Table 2b). The baseline accuracy was 72.00%. These results demonstrate again that the learning algorithm achieved his goal and determined which weights maximize the accuracy measure.

In the test set for the second scenario, the average accuracy using binary weights was 70.75% (Table 2a) and the average accuracy using real weights was 73.75% (Table 2b). The baseline accuracy was 72.25%. The values show that the weights learned improved the average accuracy only with real weights. The values reported were also submitted to a two-sample student t-test and they proved to be statistically significant ($P < 0.5$).

An analysis of the weights learned in both scenarios shows that some relations presented importance in some folds (weights bigger or equal than 1) and in others not (weights smaller than 1). For the relations which showed a consistent pattern (all folds with values bigger or smaller than 1), we can

Table 2. Accuracy measure for cross-folding validation with the weights learned by the genetic algorithm for the Scenario 2

a) Binary Weights

	Method	1st Fold	2nd Fold	3rd Fold	4th Fold	**Average**
Train Set	Baseline	72.67%	72.67%	71.00%	71.67%	**72.00%**
	Experiment	75.33%	74.00%	73.00%	74.67%	**74.25%**
Test Set	Baseline	71.00%	71.00%	76.00%	71.00%	**72.25%**
	Experiment	70.00%	68.00%	76.00%	69.00%	**70.75%**

b) Real Weights

		1st Fold	2nd Fold	3rd Fold	4th Fold	**Average**
Train Set	Baseline	72.67%	72.67%	71.00%	71.67%	**72.00%**
	Experiment	80.00%	80.67%	76.67%	78.33%	**78.92%**
Test Set	Baseline	71.00%	71.00%	76.00%	71.00%	**72.25%**
	Experiment	69.00%	72.00%	79.00%	75.00%	**73.75%**

assess, based on the values, the importance they show in the sentiment classification. For the relations which didn't show a consistent pattern (some folds with values bigger and others smaller than 1), nothing can be said about their importance.

Our attention focus is on the experiment with real values. This experiment shows a better accuracy measure in the test set when compared to the baseline. For scenario 1, the relations circumstance, condition, elaboration, evaluation, evidence, interpretation, means and result showed a consistent pattern of high weights providing evidence that the spans under those relations are important in our sentiment classifier. The relation concession showed a consistent pattern of low weights, providing evidence that the spans under this relation are not important. In the second scenario, using real weights, we see a consistent pattern of high importance for the relations: circumstance (nucleus), condition (satellite), evaluation (satellite), evidence (satellite), interpretation (nucleus), interpretation (satellite), means (nucleus), means (satellite), purpose (nucleus), purpose (satellite). The relation evidence (nucleus) shows a consistent pattern of low importance.

6.2 RST Module

The previous experiment showed how RST theory can help sentiment analysis classification and presented the particular relations involved in this process. Although good results were achieved, the applied methodology depends on text annotated with RST. The experiment detailed in this chapter aims to remove the dependency of text annotated with RST in the SO-RST algorithm.

The first experiment showed how RST relations are used in a lexicon-based sentiment classifier. The results showed that the both scenario 1 and scenario 2 in the previous experiment achieved a good performance when used weights ranging from 0 to 5. Due this result, this experiment focuses in defining discourse structures which allow the classifier to identify those relations and apply the learned weights.

Our decision was to use regular expressions to match the discourse patterns and define the relation boundaries. We decided to use the same linguistic information that lexicon-based algorithm had, the word form and the part-of-speech. We decided to not perform a syntax analysis since the objective of the experiments was to rely only in discourse markers present in the lexicon-level of the text.

We investigated two sources in order to elucidate the patterns: the Discourse Tagging Reference Manual provided by Carlson and Marcu (2001) and the SFU Review Corpus annotated with RST previously used in the first experiment. The patterns were manually crafted by the author. Each pattern was defined by looking for discourse markers present intra-sentence, i.e, discourse markers which relate two spans inside the same sentence. The segmentation into EDUs is also provided by the pattern.

Each rule created was checked against the SFU Review Corpus in order to maximize the detection of true positives and minimize the detection of false positives.

Table 3 shows the total number of rules crafted for each relation and the number of sentences those rules matched in the SFU Review Corpus.

Table 3. Number of rules crafted for each relation and respective number of sentences matched by those rules in the SFU Review Corpus

Relation	Number of Rules	Number of Sentences Matched
Anthitesis	6	227
Background	2	1776
Cause	3	388
Circumstance	3	256
Concession	4	206
Condition	3	480
Elaboration	2	76
Means	1	134
Purpose	1	52
Unless	1	35
Total	**26**	**3630**

In this experiment we incorporated the RST rules in a new module called RST module which was incorporated in the SO-RST algorithm. We used the same weights learned in the previous experiment. We organize this experiment in two different scenarios in a similar way with the previous experiment. In scenario 1 - we used the weights from the scenario 1 in the previous experiment - the algorithm shows no distinction between nucleus and satellite. In the scenario 2 - we used the weights from the scenario 2 in the previous experiment - nucleus and satellite spans receive distinct weights.

To assign those weights, we selected in both scenarios the relations which had a consistent patter of importance and the average weight bigger than 3 or lower than 0. This decision was taken to guarantee that only the relations which show a distinction importance in the last experiment were used in this experiment.

To test our method with the assigned weights we applied the classification algorithm into two corpora: SFU Review Corpus and Movie Review Corpus V2 (Pang and Lee, 2004). The results for the accuracy were also compared with a baseline algorithm. This baseline uses the same corpora, but does not assign a weight to the RST relation (weight = 1). The results for both scenarios are summarized in the Table 4 and Table 5.

Table 4. Comparison of a lexicon-based classifier in the SFU Review Corpus with the RST module

Corpus	Accuracy
Baseline	74.81%
SO-RST - Scenario 1	74.06%
SO-RST - Scenario 2	75.57%

Table 5. Comparison of a lexicon-based classifier in the Movie Reviews Corpus V2 with the RST module

Corpus	Accuracy
Baseline	71.90%
SO-RST - Scenario 1	71.55%
SO-RST - Scenario 2	71.40%

Our experiment shows inconsistent results for both corpus. In SFU Review Corpus, the SO-RST achieved 74.06% of accuracy with the weights from scenario 1 and 75.57% with the weights from scenario 2. The baseline achieved 74.81% of accuracy. In the Movie Reviews Corpus V2, the SO-RST achieved 71.55% of accuracy with the weights from scenario 1 and 71.40% with the weights from scenario 2. The baseline achieved 71.90% of accuracy.

The factors we believe which lead us to these results are:

i. the patterns crafted cover only a small set of the discourse phenomena which occurs in the text;
ii. the patterns crafted do not cover all the important RST relations;
iii. some relations which received a high weight in the first experiment were not covered by the patterns or had few instances recognized;
iv. the use of simple lexicon discourse markers may not be enough to improve sentiment classification.

7. Conclusions

This work demonstrates how to incorporate the discourse knowledge into an algorithm in order to provide a better performance for a lexicon-based sentiment classifier.

In comparison with the previous works in sentiment analysis which directly approaches the discourse structure (Somasundaran, 2010; Taboada et al., 2008) we gave more support to the claim that the discourse structure is relevant to sentiment classification. The novelty of this work lies in demonstrating which relations in the RST theory have more impact when used with a lexicon-based sentiment classifier.

The shallow RST parser module is another outcome for this work. The parser exempts the necessity of a RST annotated corpus for the algorithm. The results of this module and the discussion presented are important to further studies in the field.

The work of this dissertation raises many questions about the use of RST in the sentiment analysis classification. Future directions of this work can focus on the improvement of the RST parser; the use of an available automatic RST parser; or the application of this study in other languages.

Acknowledgements

This project was supported by the European Commission, Education & Training, Erasmus Mundus: EMMC 2008-0083, Erasmus Mundus Masters in NLP & HLT programme.

I would like to thank my supervisors, Dr. Constantin Orasan (University of Wolverhampton) and Prof. Dr. Mário Silva (Instituto Superior Técnico) for their support and advises during this project.

References

Aue, A. and Gamon, M. (2005), Customizing Sentiment Classifiers to New Domains: A Case Study, in: *Proceedings of RANLP*, Vol. 49.

Bailey, A. (2005), Consumer awareness and use of product review websites, *Journal of Interactive Advertising*.

Carlson, L. and Marcu, D. (2001), *Discourse tagging reference manual*, ISI Technical Report ISI-TR-545.

Devitt, A. and Ahmad, K. (2007), Sentiment polarity identification in financial news: A cohesion-based approach, in: *Annual Meeting - Association for Computational Linguistics*, Vol. 45, p. 984.

Ekman, P., Friesen, W. V. and Ellsworth, P. (1982), *Emotion in the human face*, Vol. 2, Cambridge University Press.

Lyman, P. and Varian, H. R. (2003), *How much information – 2003*, Technical report, School of Information Management and Systems, University of California at Berkeley.

Mann, W. (1987), *Rhetorical structure theory: A framework for the analysis of texts*, Technical report, University of Southern California - Marina Del Rey Infromation Sciences Institute.

Mann, W. and Thompson, S. (1988), Rhetorical structure theory: Toward a functional theory of text organization, in: *Text-Interdisciplinary Journal for the Study of Discourse*, Vol. 8, Walter de Gruyter, Berlin/New York Berlin, New York, pp. 243–281.

Marcu, D. (2000), The theory and practice of discourse parsing and summarization, in: *Computational Linguistics*, Vol. 28, MIT Press, pp. 81–83.

Pang, B. and Lee, L. (2004), A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, p. 271.

Pang, B. and Lee, L. (2008), *Opinion Mining and Sentiment Analysis*, Now Publishers Inc.

Pang, B., Lee, L. and Vaithyanathan, S. (2002), Thumbs up?: sentiment clas- sification using machine learning techniques, in: *Proceedings of the EMNLP '02*, Association for Computational Linguistics, Morristown, NJ, USA, pp. 79–86.

Pardo, T. and Volpe Nunes, M. (2008), *On the Development and Evaluation of a Brazilian Portuguese Discourse Parser*, Revista de Informática Teórica e Aplicada, Vol. 15, pp. 43–64.

Polanyi, L. and Zaenen, A. (2006), Contextual valence shifters, in: *Computing Attitude and Affect in Text: Theory and Applications*, Springer, pp. 1–10.

Somasundaran, S. (2010), *Discourse-level Relations for Opinion Analysis*, PhD thesis, University of Pittsburgh.

Subba, R. and Di Eugenio, B. (2009), *An effective discourse parser that uses rich linguistic information*, Computational Linguistics , Association for Computa- tional Linguistics, pp. 566–574.

Taboada, M., Anthony, C. and Voll, K. (2006), Methods for creating semantic orientation dictionaries, in: *Conference on Language Resources and Evaluation (LREC)*, pp. 427–432.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K. and Stede, M. (2011), *Lexicon- based methods for sentiment analysis*, Computational Linguistics Vol. 35, MIT Press, pp. 1–41.

Taboada, M. and Grieve, J. (2004), Analyzing appraisal automatically, in: *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, pp. 158–161.

Taboada, M. and Mann, W. (2006), *Applications of rhetorical structure theory*, Discourse studies , Vol. 8, SAGE Publications, p. 567.

Taboada, M., Voll, K. and Brooke, J. (2008), *Extracting sentiment as a function of discourse structure and topicality*, Simon Fraser University, Tech. Rep , Vol. 20.

[1] **http://techcrunch.com/2010/08/04/schmidt-data/** (Accessed 1 March 2012). Eric Schmidt: Every 2 Days We Create As Much Information As We Did Up To 2003'.