UNIVERSIDADE DO ALGARVE
Faculdade de Ciências Humanas e Sociais

UNIVERSITY OF WOLVERHAMPTON
School of Law, Social Sciences and Communications

Pedro Balage Filho

Use of Discourse Knowledge to Improve Lexicon-based Sentiment Analysis

Mestrado Internacional em Processamento de Linguagem Natural e Indústrias da Língua

International Masters in Natural Language Processing and Human Language Technology

Faro, 2012

UNIVERSIDADE DO ALGARVE
Faculdade de Ciências Humanas e Sociais

UNIVERSITY OF WOLVERHAMPTON
School of Law, Social Sciences and Communications

Pedro Paulo Balage Filho

Use of Discourse Knowledge to Improve Lexicon-based Sentiment Analysis

Orientador/Supervisor: Dr. Constantin Orăsan
Orientador/Supervisor: Prof. Dr. Mário Jorge Gaspar da Silva*

*Instituto Superior Técnico, Departamento de Engenharia Informática

Dissertation submitted as part of the study program for the award of the Master degree in Natural Language Processing and Human Language Technology, and supported by a grant from the European Commission, Education & Culture, under the Erasmus Mundus Master Courses Program (ref. EMMC 2008-0083).

Faro, 2012

UNIVERSITY OF WOLVERHAMPTON
SCHOOL OF LAW, SOCIAL SCIENCES AND COMMUNICATIONS
MA NATURAL LANGUAGE PROCESSING & HUMAN LANGUAGE TECHNOLOGY

Name: Pedro Paulo Balage Filho
Date: 15 May 2012
Title: Use of Discourse Knowledge to Improve Lexicon-based Sentiment Analysis

Presented in partial fulfilment of the assessment requirements for the above award

Supervisor: Dr. Constantin Orăsan
Supervisor: Prof. Dr. Mário Jorge Gaspar da Silva*

*Instituto Superior Técnico, Departamento de Engenharia Informática

Declaration:

(i) This work or any part thereof has not previously been presented in any form to the University or to any other institutional body whether for assessment or for other purposes. Save for any express acknowledgements, references and/or bibliographies cited in the work, I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

(ii) It is acknowledged that the author of any project work shall own the copyright. However, by submitting such copyright work for assessment, the author grants to the University a perpetual royalty-free licence to do all or any of those things referred to in section 16(i) of the Copyright Designs and Patents Act 1988 (viz: to copy work; to issue copies to the public; to perform or show or play the work in public; to broadcast the work or to make adaptation of the work.

(iii) This project did not involve contact with human subjects, and hence did not require approval from the LSSC Ethics Committee.

Signed:
Date: 15 May 2012

ABSTRACT

Sentiment Analysis deals with the computational treatment of sentiment in texts. The recent interest for sentiment analysis has grown due the popularity of internet and the increase of user-generated contents, such as blogs, social networks and reviews websites.

This work understands sentiment analysis as a classification problem. In this problem, a text can be classified as positive or negative. Sentiment classifiers can be distinguished by two main approaches: machine learning and lexicon-based. The machine learning approach uses a corpus to automatically learn the best classification features. The lexicon-based approach uses a previously computed dictionary with the sentiment lexicon.

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). This theory organizes the discourse in 26 relations that hierarchically represent the text discourse.

This objective of this work is to use discourse knowledge to improve a lexicon-based sentiment classifier. To achieve this goal it proposes the SO-RST, a lexicon-based algorithm that weights 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.

The results obtained showed which RST relations most help the lexicon-based classifier to achieve a better accuracy. The discourse markers introduced in the algorithm showed some directions to follow and the necessary steps to better study this technique.

Keywords: Sentiment Analysis, Lexicon-based Sentiment Analysis, Discourse, Rhetorical Structure Theory

RESUMO ESTENDIDO

A análise de sentimentos é um campo de estudo que investiga o tratamento dos sentimentos presentes em um texto. Este campo de estudo teve uma recente popularização devido ao crescimento da internet e do conteúdo que é gerado por seus usuários. Exemplos de tais conteúdos são: blogs, redes sociais e sites de opiniões sobre produtos e serviços.

Neste trabalho adota-se a análise de sentimentos como um problema de classificação de textos. Desde modo, um texto pode ser classificado conforme o sentimento que ele representa. De maneira mais abrangente, utilizamos em análise de sentimentos a classificação do sentimento nas classes positivo e negativo. Esta mesma classificação é adotada por este trabalho.

Existem duas abordagens na classificação de sentimentos: classificadores baseados no aprendizado de máquina e classificadores baseados na utilização de léxico.

Os classificadores baseados no aprendizado de máquina utilizam um córpus para aprendizado. Este aprendizado seleciona os atributos mais importantes que distinguem os textos pertencentes às classes de sentimento. O aprendizado de máquina pode ser supervisionado ou não supervisionado. No método supervisionado, um córpus com exemplos rotulados irá auxiliar o classificador a selecionar os atributos mais importantes. No aprendizado não supervisionado, um método de auto-alimentação (*Bootstapping*) irá gerar novos atributos baseados na

similaridade com atributos sementes. Estes atributos podem ser palavras ou locuções, que são comparadas para se determinar se expressam um sentimento negativo ou positivo.

Os classificadores baseados em léxico utilizam um dicionário de palavras ou locuções para classificar o texto. Este dicionário é aplicado ao texto para se determinar a orientação semântica, ou polaridade, de cada palavra. Esta orientação semântica é determinada por um valor numérico. A soma destes valores numéricos de todas as palavras presentes no texto resulta na orientação semântica do texto em si. Para classificação, compara-se este valor com um limiar. Se o valor for superior a este limiar o texto é positivo, senão negativo.

Discurso é um nível de análise linguística de um texto onde o autor representa suas ideias. Estas ideias são conectadas de forma a ligar conceitos em uma cadeira racional de pensamentos. Uma das representações mais importantes do discurso é a *Rhetorical Structure Theory* (RST). Nesta teoria, os segmentos de texto são organizados hierarquicamente em 26 relações possíveis. Cada relação une dois segmentos de texto e pode ser categorizada como uni-nuclear ou multi-nuclear. As relações uni-nucleares unem um segmento denominado núcleo a outro segmento denominado satélite. O núcleo é o elemento principal da relação, sem o qual o sentido do satélite não é mantido. As relações multi-nucleares são relações que possuem dois núcleos.

O objetivo deste trabalho é utilizar o conhecimento discursivo para melhoria de um classificador de sentimentos baseado em léxico. Neste trabalho utiliza-se a teoria RST para modelar o discurso. O principal foco do trabalho é a apresentação de um algorítimo para classificação de sentimentos baseado em léxico. Este algorítimo, chamado de SO-RST, usa o conhecimento discursivo para aumentar ou diminuir a importância de determinadas sentenças no texto. Estas sentenças são sinalizadas pela relação RST que elas abrangem.

O SO-RST é derivado do algoritmo SO-CAL (Taboada et al., 2011).

O SO-CAL utiliza um dicionário de orientações semânticas com valores entre -5 e 5 para atribuir a polaridade das palavras. Para cada palavra do texto encontrada também no dicionário é verificado se o escopo da palavra não pertence a uma negação, a uma palavra intensificadora, ou a um verbo modal. Cada um destes fenômenos linguísticos altera a orientação semântica da palavra. O SO-CAL calcula a soma de todas as orientações semânticas para determinar a classe de sentimento que o texto pertence.

Em nosso algoritmo, foi adicionado um passo extra ao algoritmo reportado pelo SO-CAL. Neste passo, a orientação semântica das palavras presentes no texto é multiplicada por um peso w_i. Este peso w é relacionado com a relação RST i a que a palavra abrange. Este peso pode assumir valores entre 0 e 5.

Para o teste deste algorítimo proposto foi necessário determinar os pesos de cada relação RST. Deste modo, foi realizado um experimento utilizando um córpus de textos de um site de opiniões para calcular os melhores valores para estes pesos. O córpus SFU Reviews (Taboada et al., 2006; Taboada and Grieve, 2004) utilizado neste experimento possui uma anotação manual das relações RST presentes. Neste córpus foi aplicada uma validação cruzada em 4 etapas. Em cada uma das 4 etapas, 3/4 do córpus foram destinadas a treino e a 1/4 para teste. No treino foi utilizado um algoritmo genético para determinar heuristicamente os melhores pesos para cada relação. No teste, validou-se os pesos aprendidos em nosso algoritmo obtendo-se então a medida de exatidão, isto é, o número de textos corretamente classificados.

Neste primeiro experimento obtivemos dois resultados importantes. O primeiro foram os pesos ideais de cada relação RST e assim o quão importante cada relação é para o nosso método. O segundo resultado foi a verificação de que o uso dos pesos aprendidos na fase de treino melhorou o desempenho quando submetidos na fase de teste. Este resultado suporta a evidência que o conhecimento discursivo melhora a classificação de sentimentos baseadas em léxico.

Em um segundo experimento, objetivamos remover a necessidade da utilização de um córpus anotado com RST. Para isto, desenvolvemos um novo módulo que faz a anotação discursiva de um texto baseando-se em padrões léxicos. Estes padrões foram elaborados a partir do manual de anotação RST proposto por Carlson and Marcu (2001) e pelo córpus SFU Reviews anotado com RST. Cada padrão foi anotado manualmente pelo autor e verificado no córpus SFU Reviews para maximizar os verdadeiros positivos e minimizar os falsos positivos.

Deste segundo experimento obtivemos um algoritmo que não depende mais de um córpus RST e que é capaz de aceitar qualquer tipo de texto. Testes realizados com o córpus SFU Reviews e o córpus Movie Reviews 2 (Pang and Lee, 2004) mostraram que mais testes são necessários na direção do módulo de anotação discursiva.

O algoritmo SO-RST desenvolvido e os experimentos reportados mostraram como é possível utilizar o conhecimento discursivo na classificação de sentimentos por léxico. Este estudo abre perguntas para futuros estudos e mostra a importância de relações RST específicas na classificação de sentimentos.

Palavras Chave: Análise de Sentimentos, Análise de Sentimentos Baseada em Léxico, Discurso, Rhetorical Structure Theory

To Juliana, for her unconditional love

Acknowledgements

I want to express my sincere acknowledgements to my family, in special to my wife, Juliana, who always encouraged me to do my best and supported me in every decision.

I would like to express my gratitude to my both supervisors, Dr. Constantin Orăsan and Professor Dr. Mário Silva, for their attention and effort to provide me the best advise during the program and their many corrections of my work. I would also like to thank Professor Dr. Jorge Baptista for the support and advise in this last year of the master program.

An special thanks to Dra. Caroline Brun and Gilbert Rondeau, who supervised my internship in sentiment analysis and with whom I could learn so much.

Finally, but not least, another thank to my colleagues and the professors in master program who allowed me to have one of the great experiences of my life.

Contents

List of Figures

List of Tables

Chapter 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 (Siegler, 2010). A study by Snow (2011) shows that every day, in average, are created:

- 140 million tweets
- 1,5 billion pieces of facebook content
- 10 million posts on tumblr
- 1,6 million blog posts
- 2 million videos on youtube
- 5 million images on flickr
- 60,000 new websites

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). According to two surveys with more than 2000 American adults each, presented in Pang and Lee (2008):

- 81% have done online research on a product at least once;
- 20% do so on a typical day;
- among readers of online reviews, between 73% and 87% report such reviews had a significant influence on their purchase;
- consumers reported being willing to pay from 20% to 99% more for a 5-star-rated item than 4-star-rated item;
- 32% have provided a rating on a product, service, or person via an online ratings system;
- 30% have posted an online comment or review regarding a product or service.

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.

1.1 Motivation

Understanding and processing sentiment in text is not a trivial task. The following text exemplifies some of the factors that challenge sentiment analysis. This text is a review extracted from the website Epinions.com in the category Movies. The sentences are numbered for the purpose of identification.

> *(1) So much could have been done with this classic story. (2) Instead, the result is major disappointment, a poor attempt to mimic The Grinch (which worked) and other movies of this genre. (3) Myers is a fish out of water, trying hard to liven-up an otherwise dreadful script and story line, and pandering to adults with "oh-so-current" references and innuendo.*
>
> *(4) The mean-spirited character over-played by Alec Baldwin is enough to give nightmares to children familiar with his role as the benevolent Mr. Conductor in the Thomas the Tank Engine movie. (5) Kelly Preston is gorgeous.*
>
> *(6) I was insulted every time the two disturbed children were forced by the Director to "look at the camera, show astonishment, look at each other, then look back at the camera". (7) What a tremendous lost opportunity*

This review shows a negative opinion about the movie "Dr. Seuss' The Cat in the Hat." [1] Sentence 1 demonstrates the reviewer desire for a better movie. In

[1] Available at http://www.epinions.com/review/mvie_mu-1127311/content_119377727108

Sentence 2, the reviewer concludes the desired initiated in Sentence 1 with the information that the movie was disappointing. Sentences 3 and 4 give background of his point of view with comments about the plot. Sentence 5 praises an actress performance. Sentence 6 shows a personal affirmation about the reviewer experience. Sentence 7 closes the argumentation based on all the previous sentences.

In this example we can understand why sentiment analysis is not a trivial task. Assuming that a system has to determine if the review is positive or negative, these are some of the challenges it will face:

- understands the object of the review. Link concepts like movie, actors, director, and characters to the reviewer's opinion (e.g., *"poor attempt to mimic The Grinch"*);
- understands temporal information and the presence of *irrealis* (e.g.,*"So much could have been done with this classic story"*);
- understand the discourse structure of the text and the argument evolution sequence (e.g., *"Instead, the result . . . "*);
- understands idiomatic expressions (e.g., *"fish out of water"*);
- understands contextual information about the public expectation (e.g., *"give nightmares"* is a negative aspect for a children movie, but a positive aspect for a horror movie).

For more information, Pang and Lee (2008, chap. 3) present a comprehensive survey about the challenges faced by sentiment analysis.

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 need for this approach becomes clear when the following review from the Epinions.com website is considered.

> *It could have been a great movie. It could have been excellent, and to all the people who have forgotten about the older, greater movies before it, will think that as well. It does have beautiful scenery, some of the best since Lord of the Rings. The acting is well done, and I really liked the son of the leader of the Samurai. He was a likeable chap, and I hated to see him die... But, other than all that, this movie is nothing more than hidden rip-offs.*

In the text above, the user presents a negative opinion about the movie "The Last Samurai." Despite the negative opinion of the text, it presents much more positive ideas than negative ones: *great*, *excellent*, *greater*, *beautiful*, *some of the best*. A sentiment classifier which does not consider the discourse structure will not understand the development of ideas which culminated in negative opinion evidenced by the last sentence (*But, other than all that, this movie is nothing more than hidden rip-offs*). The majority of the sentiment analysis architectures rely in the lexicon and uses simple counts of positive and negative terms to categorize the text.

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 next section presents the objective of this study and the goals it wants to achieve.

1.2 Objectives

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. In summary, this work aims to answer the following questions:

1. Can discourse knowledge help lexicon-based sentiment classifiers?
2. Which RST relations are more important for lexicon-based sentiment classification?
3. How to incorporate those important relations into the classifier algorithm?

By the question 1, this study wants to determinate if the discourse structure present in text gives additional information to the classifier which helps it in the classification process. By the question 2, this study wants to determinate which relations in the RST theory better contribute to the classification process. The importance of a relation is measured by how likely the sentences under this relation indicate the polarity of the text. Question 3 asks how those important relations can be incorporated in a sentiment classifier.

The next section presents the methodology and the experiments conducted to answer those questions.

1.3 Methodology and Results

In order to answer the questions raised in the previous section, this work focuses on a lexicon-based sentiment analysis algorithm which was modified to incorporate the discourse knowledge. This algorithm, called SO-RST, receives as input the sentences present in the text and uses the discourse knowledge to increase or decrease their importance. The algorithm classifies the text as positive or negative depending the sum of the sentiment values found in each sentence.

In this algorithm, the discourse knowledge is modelled by the Rhetorical Structure Theory (RST). The RST theory defines 26 relations responsible to link hierarchically the discourse structure of a text. Each relation connects a pair of text spans.

Lexicon-based sentiment analysis uses a dictionary of polarities in the classification. The polarity of a word, as known as the semantic orientation, is a numeric value which expresses the sentiment presented by the word. In a dictionary where the semantic orientations range from -5 to 5, strong positive words have semantic orientation close to 5 and strong negative words have semantic orientation close to -5.

The SO-RST algorithm uses a dictionary to compute the semantic orientation of individual words. This orientation is then modified by the presence of negation, intensifiers and *irrealis*. The final semantic orientation of a sentence is computed by the sum of the semantic orientation presented in the individual words. The category which the text is classified (positive or negative) relies on the averaged sum for the semantic orientation of the sentences. If the semantic orientation value is below a threshold, the text is negative, otherwise positive.

To incorporate the discourse knowledge, an additional step is introduced in the SO-CAL algorithm. After computing the semantic orientation of the individual words, the SO-RST algorithm multiplies this value by a weight ranging from 0 to 5. This weight is distinct for each RST relation in the text. The weight which multiplies each word is the one related with the discourse role for sentence where the word appears.

In order to evaluate the algorithm described above, it is necessary to determine the weight values. These values are computed by an experiment using a corpus of reviews annotated with the RST structure. The corpus is used to train and test a model to determine the best weights. This same model shows which are the most important relations for the algorithm proposed and their values.

A second experiment is conducted using the weights learned in the first experiment. In this experiment, a shallow RST parser module is developed for the algorithm proposed. This module uses the lexicon discourse markers present in the text to identify RST relations. Two corpora are analysed using this new module and the accuracy of this new method reported.

The next section shows the contributions of this work.

1.4 Contributions

The main contribution of this work is the lexicon-based sentiment analysis algorithm proposed. The algorithm, called SO-RST, shows us a manner to incorporate discourse knowledge and how to use it in sentiment classification.

In order to develop SO-RST it is conducted an exploratory study to learn how RST relations can help a lexicon-based sentiment classifier. This study shows a new understanding about the use of particular RST relations in sentiment analysis.

The shallow RST parser module is another outcome for this work. The parser excludes 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.

1.5 Thesis Outline

The dissertation is organised as follows. Chapter 2 introduces and details concepts in sentiment analysis and discourse, detailing the sentiment approaches and the main works in the literature. It also presents a literature review for the existing works in sentiment analysis which use discourse knowledge. Chapter 3 shows the algorithm proposed in this work. Chapter 4 presents two experiments conducted in order to answer the basic questions proposed in this work. Finally, Chapter 5 presents a discussion and concludes.

Chapter 2

Concepts and Literature Review

This chapter introduces concepts from sentiment analysis and discourse in more detail. The chapter also presents the approaches for sentiment analysis and their main works. It is also presented a literature review about sentiment analysis and the use of discourse. Finally, it discusses the important points and makes considerations for the work described in this thesis.

2.1 Sentiment Analysis

Sentiment analysis or opinion mining deals with the computational treatment of opinion, sentiment and subjectivity in text (Pang and Lee, 2008). 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. Although sentiment may be represented in different ways (polarity, mood, feelings), the works reported in this chapter categorize a text between positive or negative categories. Some of the works also provide a previous classification to determine if the text is polar (contain sentiment) or neutral (absent of sentiment).

Formally, a sentiment or opinion, is defined by Liu (2009) as a quintuple (o_j, f_{jk}, oo_{ijkl}, h_i, l_l) where o_j is an object, f_{jk} is a feature of the object o_j, oo_{ijkl} is the orientation or polarity of the opinion on feature f_{jk} of object o_j, h_i is the opinion holder and t_l is the time when the opinion is expressed by h_i.

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 and to determine if the overall opinion is positive or negative. This represents the most difficult level of analysis. It is, at this level, that the author intentions are most important to sentiment classification.

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 next subsections explain in detail and show the main works for these approaches.

2.1.1 Supervised Learning Approach

Sentiment classification is very similar to document classification. In the context of document classification, a set of documents D is mapped into a set of classes C, which in sentiment analysis, represent the sentiment classes. The results of the classification process is a set of pairs $\{d_i, c_j\} \in D \times C$ where d_i is classified into $c_j \in C$ (Sebastiani, 2002).

Supervised machine learning requires two sets of documents D: a training and a test set. A training set is used by the classifier to learn how differentiate a document into a specific class. This differentiation happens in function of a set of features determined in the document. The learning algorithm focuses on discovering and weighting the features which most correlate a document with his class. The test set is used validate the performance of the classifier obtained in the learning step.

Existing supervised learning algorithms like naïve Bayes, decision trees and support vector machines (SVM) can be easily applied to sentiment classification problem. The main effort in the process to build a good sentiment analysis system is not the algorithm decision, although this factor matters, but the choice of a representative training corpus and a good set of features.

Pang et al. (2002), one of the first works in sentiment analysis with machine learning, performed supervised classification in a movie reviews dataset. The corpus[1] used on this work was collect from the *Rotten Tomatoes Website*[2] and consists in 2000 movie reviews where 1000 are positive labelled reviews and 1000 are negative labelled reviews. The decision of labelling each text as positive or negative was determined by the number of stars for each movie. With a five star system, movies with three-and-a-half stars and up are considered positive while movies with two stars and below are considered negative.

Pang et al. (2002) showed that using unigrams, as a bag-of-words feature model, and the relative position for the words, the performance is higher when used either naïve Bayes, maximum entropy or SVM algorithms. The maximum accuracy obtained for her classifier was 81% in the movie reviews dataset. One of the most challenging points highlighted by the author and a common phenomenon

[1] Available at `http://www.cs.cornell.edu/people/pabo/movie-review-data/`
[2] `http://www.rottentomatoes.com/`

in the corpus is the “thwarted expectations” narrative. The author exemplifies the phenomenon in the following text excerpt:

> *This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up" or "I hate the Spice Girls. ...[3 things the author hates about them]... Why I saw this movie is a really, really, really long story, but I did, and one would think I'd despise every minute of it. But... Okay, I'm really ashamed of it, but I enjoyed it. I mean, I admit it's a really awful movie ...the ninth floor of hell...The plot is such a mess that it's terrible. But I loved it.*

In the text above, the reviewer makes use of a lot of deliberate contrast to earlier discussions. This kind of phenomenon, according Pang et al. (2002) turns the classification process difficult for a supervised machine learning approach.

Wilson et al. (2009) present a further study with a more elaborated set of features for supervised machine learning. They present an exploratory study on features for phrase level sentiment analysis with the Multi-Perspective Question Answering (MPQA) corpus (Wiebe et al., 2005). The listing below describes some of the features used in their work as well as in other supervised learning works:

- *Terms and their frequency*: words present in the text, as individual words or n-grams and their frequency counts. In some cases, word positions may also be considered. In this feature it is important to apply a selection filter like TF-IDF weighting scheme that can distinguish the most valuable attributes for each class.

- *Part of speech tags*: part of speech tags are important indicators of subjectivities and opinions. Thus, adjectives, for example, can be selected as special features for the machine learning algorithm.

- *Negation*: negation and its scope is a very important feature. For example, sentences like *I don't recommend* have an opposite polarity to *I recommend*. The ability to know that a word was negated makes all the difference for a classifier.

- *Opinion words and phrases*: opinion words can be inserted as characteristics to express each sentiment or class. Words like beautiful or poor, or even expressions can be used to determine the orientation of a text to positive or negative class.

- *Syntactic dependency*: words dependency based features generated from parsing or dependency trees are also tried by some researchers.

2.1.2 Unsupervised Learning Approach

Unsupervised learning differs from supervised methods in the way that it does not require a corpus with labelled examples. In some aspect, unsupervised learning algorithms are more robust because they can easily adapt to different domains when a labelled corpus is not available. Unsupervised learning methods often use bootstrapping methods, i.e., a small seed of examples is given and the algorithm is able to retrieve by similarity other instances of training. Despite the robustness of this method, it is susceptible to semantic drifts. A semantic drift occurs when, in the bootstrapping process, an example of positive instance is learned as negative or vice versa.

Turney (2002) presents an unsupervised classification of reviews based on some fixed syntactic phrases that are likely to be used to express opinions. The algorithm has three steps:

1. extract phrases containing adjectives or adverbs;

2. estimate the orientation of the extracted phrases using the pointwise mutual information (PMI) measure (Church and Hanks, 1990);

3. compute the average orientation of all phrases and classifies the review into positive or negative.

In the first step, only adjectives and adverbs are targeted since they are good indicators of subjectivity and opinions. The phrases extracted contain two consecutive words, where one of them is an adjective or adverb. For example, in the sentence: "The iPhone has a nice design", the adjective "nice" should keep the noun "design" as feature. Table 2.1 presents the rules applied in order to extract terms. Note that a third word is sometimes necessary to validate the rule, but it is not extracted by the algorithm.

TABLE 2.1: Patterns of tags for extracting phrases from reviews (Turney, 2002)

	First Word	Second Word	Third Word (Not Extracted)
1.	JJ	NN or NNS	anything
2.	RB, RBR, or RBS	JJ	not NN nor NNS
3.	JJ	JJ	not NN nor NNS
4.	NN or NNS	JJ	not NN nor NNS
5.	RB, RBR, or RBS	VB, VBD, VBN, or VBG	anything

After extracting all phrases, the second step is to measure the polarity, or opinion orientation (oo). For this, Turney (2002) measures the Pontwise Mutual Information (PMI) between the phrase and the positive sentences. The same process is done with the phrase and the negative sentences. The similarity coefficient by the algorithm will express how negative or positive is the phrase.

The probabilities are calculated by performing queries to a search engine and collecting the number of hits. Turney (2002) used the AltaVista search engine because it has a NEAR operator that constrain the search to a ten words window. The final equation for PMI adapted with AltaVista is shown in equation 2.1.

$$oo(phrase) = \log_2 \left(\frac{hits(phrase NEAR \text{``}excellent\text{''})hits(\text{``}poor\text{''})}{hits(phrase NEAR \text{``}poor\text{''})hits(\text{``}excellent\text{''})} \right) \tag{2.1}$$

Finally, in Step 3, the algorithm averages the obtained opinion orientation (oo) for each phrase extracted in Step 2. If the averaged oo is a negative number, the algorithm classifies the text as negative, otherwise positive. The authors performed test in a corpus of 410 reviews from the domains of automobiles, banks, movies and travel destinations. The final accuracy ranged from 84% for automobile reviews to 66% for movie reviews.

2.1.3 Lexicon-based methods

The main characteristic of lexicon-based methods is the use of a dictionary with polar vocabulary. This dictionary is used to determine the presence of polar terms in the text. The polarity class assigned by a lexicon-based classifier is determined by computing the average polarity contained in the individual words.

The next subsection shows the process to build dictionaries. Subsection 2.1.3.2 shows how to apply the dictionary in a lexicon-based classifier.

2.1.3.1 Building Dictionaries

To compile a dictionary for opinion words, i.e., words with an associated sentiment, there are three possible approaches: manual, dictionary-based, and corpus-based. The manual approach consists in collecting and building the dictionary manually, which is a very time-consuming task. The dictionary-based approach uses a standard or custom language dictionary in order to determine the polarity of the words. The corpus based approach uses a corpus to extract similarities between positive and negative words.

One of the simplest techniques for the dictionary-based approach is reported by Hu and Liu (2004) and Kim and Hovy (2004). This approach is based on bootstrapping using a small set of seed opinion words and an online dictionary, e.g., WordNet (Fellbaum, 1998). This strategy first collects a small set of opinion words and then uses WordNet to grow this set for their synonyms and antonyms. The new words are included in the dictionary and the process is restarted. The iterative process stops when no more new words are found. Usually, after this process, the words are verified and corrected by hand, finalizing the dictionary construction.

Some researchers have appointed that additional information, like glosses in WordNet and additional techniques (e.g., machine learning) generate better lists (Andreevskaia and Bergler, 2006; Esuli and Sebastiani, 2006*a*,*b*). Despite the efficiency of these methods to construct polarity dictionaries, they are unable to find opinion words with domain specific orientations. For this problem, a corpus-based approach represents a good solution.

The corpus-based approach relies on syntactic or co-occurrence patterns and a list of seeds to find other opinion words in a large corpus. One of the first works is described in Hatzivassiloglou and McKeown (1997) where the authors use a regression model to classify conjoined adjectives into positive or negative categories. They use a set of linguistic constrains or conventions on connectives to identify additional adjective opinion words and their orientations. For example, in the sentence *"this phone is beautiful and small"*, if *beautiful* is known to be positive, so *small* can be categorized as positive as well. Other connectives like *or*, *but*, *either-or* and *neither-nor* are used in the work.

Qiu et al. (2009) propose a propagation approach that exploits the relations between sentiment words and topics or product features that the sentiment words modify. The extraction rules are designed based on relations described in dependency trees.

Ding et al. (2008) explores the idea of intra-sentential and inter-sentential sentiment consistency. They showed that the same word might have different orientations in different contexts. Their method determines opinion words and their orientations together with the object features that they modify.

2.1.3.2 Lexicon-based Sentiment Classifier

The lexicon method uses a dictionary to determine the polarity, or semantic orientation, for the individual words in the text. This 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).

Linguistic features like negation scope (e.g., not good), *irrealis* (e.g., could be good) and intensifiers (e.g., very good) could be easily addressed in this method by the incorporation of steps in the process. Some authors also prefer lexicon methods, in opposite to supervised machine learning methods, since the dictionary used is independent from domain (Taboada et al., 2011).

One of the first works with lexicon methods is reported by Polanyi and Zaenen (2006) with contextual valence shifters. The authors showed that the polarity valence of terms can be modified by the context. They proposed two categories of contextual valence shifters: Sentence Based and Discourse Based. The sentence based valence shifters are listed below:

1. *Negatives:* The negation is the most basic shifter. The use of the *not* flipping the valence of a term has been discussed in some other works (Das and Chen, 2001; Pang et al., 2002). Other negatives have also the same effect: *never*, *none*, *nobody*, *nowhere*, *nothing*, *neither*.

2. *Intensifiers:* Intensifiers have the role to weaken or strengthen the valence around it. For example, in the sentence *the very ugly design*, the word *very* has the function of intensifier.

3. *Modals:* According to Polanyi and Zaenen (2006): "Language makes a distinction between events or situations which are asserted to have happened, are happening or will happen (*realis* events) and those which might, could, should, ought to, or possibly occurred or will occur (*irrealis* events)". For example, the sentence *If Mary were a terrible person, she would be mean to her dog* express an idea of possibility, but not a direct sentiment expressed.

4. *Presuppositional Items:* Some words have the ability to shift the valence of evaluative terms through their presuppositions. For example the adverbs

barely as shown in the examples *It is sufficient* with *It is barely sufficient*. *Sufficient* is a positive term, *barely sufficient* is not. The use of this term presupposes that better was expected.

5. *Irony:* This characteristic of natural language can be exemplified by the sentence *The very brilliant organizer failed to solve the problem*. The irony presupposes some knowledge of the world.

The work by Polanyi and Zaenen (2006) was theoretical and no implementation of such problems was done. Kennedy and Inkpen (2006) concentrated on implementing those ideas. They create features to deal with negatives and intensifiers. In their approach, they flip the polarity of the next word, in the case of negation, or intensify by some amount, in the case of intensifiers. The approach does not deal with the scope of negatives and intensifiers. Other works address this issue. Choi and Cardie (2008) present a work in compositional semantics. Their classifier treats negation from a compositional point of view by first calculating polarity of terms independently, and then applying inference rules to arrive at a combined polarity score.

The most important lexicon-based method is reported by Taboada et al. (2011), where 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:

1. individual words have a prior polarity, which is independent from context;
2. the semantic orientation can be expressed as a numerical value.

Taboada et al. (2006) report a method to build a semantic orientation dictionary similar to those described by Turney (2002). Instead using AltaVista with the operator NEAR, they used the Google search engine to retrieve the words and their polarity coefficients. Different from the proposed by Turney (2002), Taboada et al. (2006) showed not only that adverbs and adjectives matter, but also do nouns and verbs. After the bootstrapping process, their dictionary was revised and the mistakes of the method corrected. The final version 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[3] 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. In these reviews, the user explicitly indicated in the website if the product is "recommended" or "not recommended." This feature was used to determine whether a review is positive or negative.

The SO-CAL algorithm can be summarized as follows:

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

[3]Available at `http://www.sfu.ca/~mtaboada/research/SFU_Review_Corpus.html`

The next section introduces concepts about discourse. It shows how discourse structure is organized around different frameworks and presents the Rhetorical Structure Theory (RST), the discourse theory used in this work. The section also shows how RST is organized into relations and how those relations are hierarchically applied into text spans.

2.2 Discourse

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.

Grosz and Sidner (1986) present a theory to model the intentional aspect, Jordan (1992) and Kehler (2002) propose semantic relations to this structure, Mann (1987) proposes a theory called Rhetorical Structure Theory (RST), which includes the intentional and semantic relations. According to Reiter and Dale (2000), the RST is one of the most complete theories to describe the discourse representation and it is useful for several tasks in natural language processing.

The next section details the Rhetorical Structure Theory, which is also the discourse theory used in this work.

2.2.1 Rhetorical Discourse Theory (RST)

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. Table 2.2 presents the RST relations as described by Mann (1987) and their nuclearity.

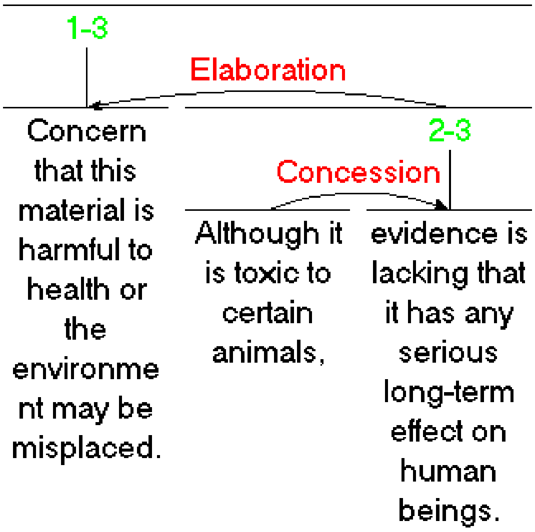
TABLE 2.2: Relations defined by the Rhetorical Structure Theory (Mann, 1987)

Relation	Multinuclear
ANTITHESIS	No
BACKGROUND	No
CIRCUMSTANCE	No
CONCESSION	No
CONDITION	No
ELABORATION	No
ENABLEMENT	No
EVALUATION	No
EVIDENCE	No
INTERPRETATION	No
JUSTIFY	No
MEANS	No
MOTIVATION	No
NON-VOLITIONAL CAUSE	No
NON-VOLITIONAL RESULT	No
OTHERWISE	No
PURPOSE	No
RESTATEMENT	No
SOLUTIONHOOD	No
SUMMARY	No
VOLITIONAL CAUSE	No
VOLITIONAL RESULT	No
CONTRAST	Yes
JOINT	Yes
LIST	Yes
SEQUENCE	Yes

In the literature, one can find some automatic RST parsers for several languages (Marcu, 2000; Pardo and 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.

Figure 2.1 shows a graphical example of the RST organization. In this figure, it can be seen three spans of text related by the relations elaboration and concession. The spans two and three are related by the relation concession. In a mono-nuclear relation, the arrow point indicates which span is the nucleus of a relation. The group of spans two and three is also linked by an elaboration relation into span one.

FIGURE 2.1: Example of RST structure (Mann, 1987, p. 15)



The next section presents work in the literature which uses concepts from discourse structure in sentiment analysis.

2.3 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. The authors proposed two categories of contextual valence shifters: sentence based and discourse based. The discourse based contextual valence shifters are described below (sentence-based contextual valence shifters were already presented on subsection 2.1.3.2):

1. *Connectors*: are words such as *although*, *however*, *but*, *on the contrary*, *notwithstanding*, etc. The connectors can introduce an information or act on some information given elsewhere in the text. For example, in the sentence *Although Boris is brilliant at math, he is a horrible teacher*, the connector *Although* informs the fact that Boris is brilliant at math is less relevant than the fact that he is a horrible teacher.

2. *Discourse Structure and Attitude Assessment:* concerns to the discourse structure itself. The two basic discourse relations of interest to us are: lists and elaborations. Elaborations, for example, give us more detail about another constituent. For example, in the text *John walks a lot. Last month he walked 25 miles on Tuesdays*, the second sentence illustrates the concept expressed in the dominating sentence. When the valence information is introduced in a dominating sentence, the elaborations reinforce its effects, so, even if the elaboration does not introduce a new polarity to the text, it acts as an intensifier.

3. *Multi-entity Evaluation:* the entities (features) that the author mentions. For example, a review can be very negative about some characteristics, but positive about the product itself.

4. *Genre and Attitude Assessment:* terms and expressions used in a text are directly related to the purpose of the text. For example, product reviews have a different language than company reports or political debates.

5. *Reported Speech:* the reported speech does not reflect the direct opinion from the author.

6. *Subtopics:* the author can split his point of view into subtopics. Each subtopic inside the main text receives a particular opinion or sentiment.

7. *Genre Constraints:* for example, some movie reviews present plot summaries before the author's opinion. In the opinion of Polanyi and Zaenen (2006) these kind of constrains for the movie genre should be considered.

Pang et al. (2002) showed that the incorporation of discourse elements to help sentiment classification can happen in a very simple way. In this study, the authors decided to include 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.

The work of Mao and Lebanon (2006) proposes an interesting approach. They model the global sentiment of a document as a trajectory of local sentiments. In their research each sentence in the document receives a local sentiment score and it is mapped into a conditional random field predictor. The flow is then smoothed out through convolution with a smoothing kernel. The idea behind this model is that the two flows should reflect the distances between global sentiments. In the experiments, the authors verified that the sentiment flow (especially when objective sentences are excluded) outperforms a plain bag of words representation in predicting global sentiment with a nearest neighbour classifier.

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.

The next section presents a discussion about the concepts presented in this chapter.

2.4 Discussion

The Lexicon-based method is known for being domain-independent, while the machine learning method tends to adapt to the domain that the classifier learned. Aue and Gamon (2005) show that machine learning method performance drops precipitously (almost to chance) when the same classifier is used in a different domain.

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 as negation scope, *irrealis* and intensifiers are found. As a result, this work uses a lexicon-based method in our sentiment classification.

As described in Section 2.3, some works already cover the use of discourse knowledge in sentiment analysis. Although, none of them focus on the importance of particular discourse relations in the opinionated text. The next chapter presents the ideas proposed in this work. It also presents the algorithm designed.

Chapter 3

SO-RST Algorithm

This chapter describes the SO-RST algorithm proposed in this work. As specified in Chapter 1, this study aims to identify the discourse structure in the text and use it to improve the sentiment classification. The pivot question in this work is how to incorporate such knowledge into a lexicon-based sentiment classifier and how to use it to improve the classification accuracy.

The next section presents the concepts behind the algorithm.

3.1 Lexicon-based Sentiment Analysis Using RST

As described in Chapter 2, lexicon-based methods are useful to incorporate new linguistic features in the classifier algorithm. We have shown 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. To address linguistic phenomena like negation, intensifiers and *irrealis*, the algorithm can modify the semantic orientation for the words around by multiplying it by some factor. The idea of SO-CAL is explained in the example which follows:

> *I like the product appearance. One day, it broke down. Hence, I believe it is a bad product.*

A lexicon-based sentiment classifier split the text into sentences. Each sentence is used to compute the semantic orientation (SO), i.e., the number which indicates the polarity present in the sentence. In the example below, the text presented before is split into sentences. Inside each sentence, the SO is measured based on the polar words matched by the dictionary.

I like(+4) the product appearance.
$SO = 4$

One day it broken(-2) down.
$SO = -2$

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

$TotalSO = 4 + (-2) + (-2)$

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. In the algorithm, this value is applied to the semantic orientation of the words under the scope of those relations. The purpose of the weight is to emphasize or downplay the importance of the sentences under particular relations.

In the example reported, the average semantic orientation was not enough to classify the text as positive or negative. Using RST structure, our algorithm aims to give a higher or lower importance to RST spans and consequently improve the classification.

In this example, 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. 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 below shows how the weights will be assigned in the example.

I like (+4) the product appearance.

$SO = 4 \times w_{none}$

One day it broken (-2) down.

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

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

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

In the example, w_{none} represents the weight multiplying the words in a sentence covered by no RST relation. $w_{ResultNucelus}$ represent the weight multiplying the words in the nucleus span of a result relation. $w_{ResultSatellite}$ represent the weight multiplying the words in the satellite span of the relation result.

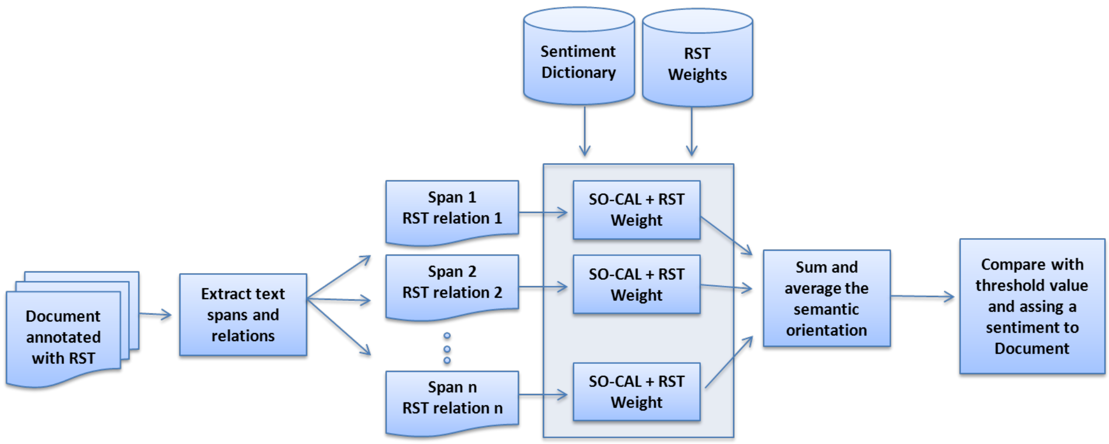
Like the original SO-CAL, the algorithm classifies the text based on the average of the semantic orientation computed. We based our experiment in the work reported by Taboada and Grieve (2004), where the SO-CAL was used with a threshold of 0.62. We also use the same dictionary of sentiment provided by 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 accuracy formula is defined by:

$$Accuracy = \frac{\text{Number of instance correctly classified}}{\text{Total number of instances}} \tag{3.1}$$

The SO-RST diagram is shown in the Figure 3.1. The Algorithm 3.1.1 shows the SO-RST pseudocode.

FIGURE 3.1: Diagram showing the steps followed by the SO-RST algorithm



In the Figure 3.1, the SO-RST is detailed. 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 an RST relation. After calculating the semantic orientation for all sentences, an averaged sum is computed and the algorithm classifies the text as positive or negative, based on the computed value.

Algorithm 3.1.1: SO-RST(*Document*)

```
dictionary ← READ_SENTIMENT_DICTIONARY()
RST_weights ← READ_WEIGHTS_RST()
TOTAL ← 0

for each Sentence ∈ Document
  do { for each word ∈ Sentence
       do { if word ∈ Dictionary
            then { SO ← GET_SEMANTIC_ORIENTATION(dictionary, word)
                   if word is negated
                     then { if SO<0
                              then SO = SO + 3
                              else SO = SO − 3
                   if word is intensified
                     then SO = SO × intensifier_value
                   if word is in irrealis
                     then SO = SO × 0
                   if word is in RST Relation
                     then SO = SO ∗ GET_WEIGHT(RST_weights, relation)
                   TOTAL ← TOTAL + SO

SO_document ← TOTAL/Number_Sentences
if SO_document > 0.62
  then return (Positive)
  else return (Negative)
```

The Algorithm 3.1.1 shows in detail how the semantic orientation for each word is modified in case of negation, intensifiers, *irrealis* or the presence of a discourse relation.

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

The first experiment uses a learning algorithm to determine the best weights in a product reviews corpus annotated with RST. The second experiment presents a simple RST parser based on discourse markers and word clues. This parser is included in the proposed algorithm in the way to eliminate the necessity for a corpus annotated with RST.

Chapter 4

Experiments and Results

This work presents an investigation of the use of discourse structure to improve feature-based sentiment analysis classification. This experimental work aims to corroborate the hypotheses that the discourse structure, represented in the RST relations, helps the classification process.

This chapter details experiments with sentiment analysis and the Rhetorical Structure Theory (RST). There are reported two experiments. The first experiment aims to find the best configuration of weights which maximizes the accuracy of the SO-RST algorithm described. For this, we used the SFU Review Corpus annotated with RST (Taboada et al., 2006; Taboada and Grieve, 2004). In sum, we want to learn which relations are important in a lexicon-based sentiment analysis algorithm and which ones are not.

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 section describes the SFU Review Corpus used in the experiments.

4.1 Corpus

The SFU Review corpus (Taboada and Grieve, 2004)[1] is a collection of 400 reviews from the website Epinions.com downloaded in 2004 from eight different categories: books, cars, computers, cookware, hotels, movies, music, and phones.

Inside each category, the texts are split into 25 positive and 25 negative reviews. The classification into positive and negative is based on the "recommended" or "not recommended" tag present in the website and provided by the reviewer.

The corpus also provides the RST annotation at the sentence level, i.e., only the relations found within sentences were annotated. The texts were annotated by Montana Hay and Maite Taboada[2], using the RSTTool [3].

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 frequency of the relations present in the corpus is described in Table 4.1.

4.2 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 one 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, three parts of the corpus are used for training and the remaining part of the corpus is used

[1] Available at http://www.sfu.ca/~mtaboada/research/SFU_Review_Corpus.html

[2] Please, report to the website http://www.sfu.ca/~mtaboada/research/SFU_Review_Corpus.html for more details

[3] http://www.wagsoft.com/RSTTool/index.html

TABLE 4.1: Number of occurrences for each RST relations in the SFU Review Corpus

Relation	**Number of occurrences**
antithesis	39
background	476
cause	767
circumstance	1019
concession	717
condition	592
elaboration	804
enablement	5
evaluation	200
evidence	101
interpretation	45
justify	22
means	76
motivation	29
nonvolitional-cause	4
nonvolitional-result	2
otherwise	2
preparation	328
purpose	366
restatement	12
result	359
solutionhood	29
summary	5
unconditional	19
unless	50
volitional-cause	3
volitional-result	5
Total	**22185**

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 chapter. Each relation is going to receive the weight learned and the words under the relation scope are going to be changed by these weights.

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 2^{26} 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 an 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. The next subsection shows how the genetic algorithm was configured in the learning process.

4.2.1 Using Genetic Algorithm to Determine the Weights

Genetic algorithm is an approach used to find a set of parameters that lead to the solution for a problem. This technique is applied in a scenario where it is difficult to determine these parameters trough conventional methods, for example, where there is no intuition about the optimal parameters or when the number of possible combinations among the variables does not allow to apply brute force techniques.

The genetic algorithm took its name from the proximity with the way which genes are combined to generate new individuals in nature. In this technique, we have a set of parameters which we call genes that are combined to generate new individuals, or programs. In this combination, a set of adjustable factors can influence the convergence of the algorithm, like mutations and cross-overs. The final idea is that it is possible to generate a family of programs with their own genes, or parameters, which are going through evolution. The programs with the best scores, measured by a fitness function, propagate their genes through the next family generation.

In genetic algorithm approach, it is possible, in the long of n generations, to reach the set of parameters closer to the optimal solution without being necessary to explore all the combinations possible.

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.

The next subsection presents how this heuristic algorithm was applied in our dataset.

4.2.2 Learning the Weights

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. The first method is motivated by the fact that suppressing some relations spans could show to the classifier what is important in the text. The second method is motivated by the goal of finding the relations which should receive more importance by the sentiment classifier.

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 (we previously presented the frequency of relations in the corpus in the table 4.1). 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).

4.2.3 Results

This subsection presents the results obtained by our experiment. We tested two scenarios (weight the whole span, or weight satellite and nucleus distinctly) and two different methods (binary or real weights). The baseline method used for each scenario is the same method but with weight=1 for all relations (no RST distinction). The results obtained by the two tested scenarios are shown in the Table 4.2 and Table 4.4.

In the training set, 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 4.2a). Using real weights the average accuracy for the training set was 78.50% against 72.00% from the baseline (Table 4.2b). These results demonstrate that the learning algorithm achieved its goal and determined which weights maximize the accuracy measure.

In the test set, we run the classifier the same weights learned from the training set on the unseen texts. The average accuracy using binary weights was 71.25% (Table 4.2a) and the average accuracy using real weights was 75.75% (Table 4.2b). The baseline accuracy for both was 72.25%. The conclusion is that the learned

Table 4.2: 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%**

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 they proved to be statistically significant ($P < 0.05$).

An analysis of the weights is shown in the Tables 4.3a and 4.3b. We can see 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 assess, based on the values, the importance they show in the sentiment classification. For example, in the Table 4.3b, the relation elaboration presents a consistent pattern of high weights for all the four folds of our experiment. This signalizes that this relation is very important to lexicon-based sentiment classification. On the other hand, the relation concession presents a consistence patter of low weights for all the four folds. This signalizes that the spans covered by this relation are not important in the classification. Other sentences, like antithesis, shows high importance in some folds and small importance in others. For these relations, the results are not consistent and nothing can be said about their importance.

TABLE 4.3: Best weights in the cross-folding validation learned by the genetic algorithm for the **Scenario 1**

a) **Binary weights**

Relation	1st Fold	2nd Fold	3rd Fold	4th Fold	Average
antithesis	0	1	0	0	**0.25**
background	1	1	1	1	**1**
cause	1	1	1	1	**1**
circumstance	0	0	1	0	**0.25**
concession	1	1	1	1	**1**
condition	1	1	1	1	**1**
elaboration	1	1	1	1	**1**
evaluation	1	0	1	0	**0.5**
evidence	1	1	1	1	**1**
interpretation	0	0	1	1	**0.5**
means	1	1	1	1	**1**
preparation	1	1	1	1	**1**
purpose	1	1	0	1	**0.75**
result	1	1	1	1	**1**
unless	1	1	0	1	**0.75**

b) **Real weights**

Relation	1st Fold	2nd Fold	3rd Fold	4th Fold	Average
antithesis	1.35	0.34	0.15	1.81	**0.91**
background	1.66	2.22	1.86	0.54	**1.57**
cause	1.77	0.69	0.93	0.11	**0.87**
circumstance	1.79	4.15	4.13	3.39	**3.36**
concession	0.2	0.34	0.16	0.09	**0.20**
condition	2.61	2.89	3.58	3.83	**3.23**
elaboration	4.02	4.49	4.53	4.53	**4.39**
evaluation	2.61	3.48	2.25	1.79	**2.53**
evidence	2.61	2.23	1.2	3.42	**2.36**
interpretation	3.57	4.32	2.25	4.19	**3.58**
means	4.02	3.48	4.13	1.26	**3.22**
preparation	1.35	0.69	0.93	0.54	**0.88**
purpose	3.8	2.63	2.25	1.81	**2.62**
result	1.35	0.96	0.93	0.54	**0.95**
unless	2.61	3.42	0.93	2.11	**2.27**

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. In this experiment, the relations circumstance, condition, elaboration, evaluation, evidence, interpretation, means and result showed a consistent pattern of high weights (see Table 4.3a) providing evidence that the spans under those relations are important to be higher weighted 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.

TABLE 4.4: Accuracy measure for cross-folding validation with the weights learned by the genetic algorithm for the **Scenario 2**

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	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%**

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 4.4a). Using real weights, the average accuracy for the training set was 78.92% (Table 4.4b). 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, the average accuracy using binary weights was 70.75% (Table 4.4a) and the average accuracy using real weights was 73.75% (Table 4.4b). The baseline accuracy was 72.25%. The values show that the weights learned improved the average accuracy when used real values for the weights. The values

Table 4.5: Best binary weights in the cross-folding validation learned by the genetic algorithm for the **Scenario 2**

Relation	1st Fold	2nd Fold	3rd Fold	4th Fold	Average
antithesis (nucleus)	0	0	0	0	**0**
antithesis (satellite)	0	0	0	0	**0**
background (nucleus)	1	1	1	1	**1**
background (satellite)	0	1	0	0	**0.25**
cause (nucleus)	1	1	1	1	**1**
cause (satellite)	0	0	1	1	**0.5**
circumstance (nucleus)	1	0	1	1	**0.75**
circumstance (satellite)	0	1	0	0	**0.25**
concession (nucleus)	1	1	1	1	**1**
concession (satellite)	0	1	0	0	**0.25**
condition (nucleus)	1	1	1	1	**1**
condition (satellite)	1	1	1	1	**1**
elaboration (nucleus)	1	1	1	1	**1**
elaboration (satellite)	0	1	1	1	**0.75**
evaluation (nucleus)	1	0	0	0	**0.25**
evaluation (satellite)	1	0	1	0	**0.5**
evidence (nucleus)	0	1	0	1	**0.5**
evidence (satellite)	1	1	1	1	**1**
interpretation (nucleus)	1	0	1	1	**0.75**
interpretation (satellite)	0	0	1	0	**0.25**
means (nucleus)	1	1	1	1	**1**
means (satellite)	1	1	1	1	**1**
preparation (nucleus)	0	1	1	1	**0.75**
preparation (satellite)	0	1	1	0	**0.5**
purpose (nucleus)	0	1	1	0	**0.5**
purpose (satellite)	0	1	0	1	**0.5**
result (nucleus)	1	0	0	0	**0.25**
result (satellite)	1	1	1	0	**0.75**
unless (nucleus)	0	1	0	1	**0.5**
unless (satellite)	1	1	1	1	**1**

reported were also submitted to a two-sample student t-test and their proved to be statistically significant ($P < 0.05$).

An analysis of the weights is showed in the tables 4.5 and 4.6. We can see again that some relations did not present a consistent pattern in all folds and some did. When we look for the binary weights we see some coherence of the importance to use the relations (see table 4.5): background (nucleus), cause

Table 4.6: Best real weights in the cross-folding validation learned by the genetic algorithm for the **Scenario 2**

Relation	1st Fold	2nd Fold	3rd Fold	4th Fold	Average
antithesis (nucleus)	1.2	1.05	0.68	2.01	**1.24**
antithesis (satellite)	2.56	0.14	3.19	1.21	**1.78**
background (nucleus)	3.71	2.12	4.59	0.81	**2.81**
background (satellite)	0.84	0.14	0.69	1.27	**0.74**
cause (nucleus)	0.13	1.47	0.68	2.28	**1.14**
cause (satellite)	2.56	0.28	0.68	1.08	**1.15**
circumstance (nucleus)	3.74	3.44	2.53	3.34	**3.26**
circumstance (satellite)	0.41	4.35	1.44	2.09	**2.07**
concession (nucleus)	4.14	0.57	4.27	2.27	**2.81**
concession (satellite)	1.29	0.01	1.2	0.38	**0.72**
condition (nucleus)	0.03	3.32	2.79	1.26	**1.85**
condition (satellite)	3.89	4.35	4.14	3.73	**4.03**
elaboration (nucleus)	0.24	4.18	4.05	4.68	**3.29**
elaboration (satellite)	4.71	0.1	4.59	3.35	**3.19**
evaluation (nucleus)	0.54	2.19	0.69	4.13	**1.89**
evaluation (satellite)	4.22	1.05	3.08	1.08	**2.36**
evidence (nucleus)	0.84	0.8	0.05	0.81	**0.63**
evidence (satellite)	2.27	1.34	2.15	3.73	**2.37**
interpretation (nucleus)	1.29	3.44	1.2	3.75	**2.42**
interpretation (satellite)	3.89	4.35	3.08	2.19	**3.38**
means (nucleus)	2.73	2.78	4.24	3.34	**3.27**
means (satellite)	2.01	1.34	2.53	3.11	**2.25**
preparation (nucleus)	0.54	1.05	1.19	2.86	**1.41**
preparation (satellite)	0.25	2.12	0.69	2.28	**1.34**
purpose (nucleus)	1.65	1.42	1.45	1.21	**1.43**
purpose (satellite)	4.14	2.12	1.65	3.34	**2.81**
result (nucleus)	0.9	0.8	1.65	0.81	**1.04**
result (satellite)	4.88	2.5	2.8	0.63	**2.70**
unless (nucleus)	1.13	0.22	3.29	3.75	**2.10**
unless (satellite)	0.84	1.13	2.41	3.11	**1.87**

(nucleus), concession (nucleus), condition (nucleus), condition (satellite), elaboration (nucleus), evidence (satellite), means (nucleus), means (satellite), unless (satellite). For the experiment using real weights, we see a consistent pattern of high importance for the relations (see table 4.6): 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.

The next chapter presents an experiment which incorporate discourse markers in the SO-RST algorithm.

4.3 RST Module

This section presents an experiment motivated from the result of the previous section. The previous experiment showed how RST theory can help sentiment analysis classification and presented the particular relations involved in this process. Although the previous method improved over the baseline, 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 objective of this experiment 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 or word clues. This chapter focuses on the relations that helped achieving a good average accuracy in the previously described experiment and explore how to incorporate those relations in our algorithm.

4.3.1 Methodology

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. The following example shows one of the patterns crafted in this process.

rule = 40
relation = "CIRCUMSTANCE"
pattern = "(?P<S>after/.+?,/,)(?P<N>.+)$"

This pattern can match a sentence and isolate the nucleus and satellite spans. In this example, the relation *Circumstance* is defined by a regular expressions that captures two parts of the sentence, the first part as a satellite (defined by the regular expression group marker ?P<S>) and the second as a nucleus (marker ?P<N>). The expression captures all sentences starting with the word *after* which have a comma (","). The segment before the comma is defined as the satellite of the relation and the segment after the comma the nucleus of the relation. In the pattern, the rule is matched against the sentence marked with the Part-of-Speech (POS) tags. In the sentence, each POS is defined after the word by a backslash ("\") and a code. The following example shows the application of this pattern. The POS annotations are omitted in this example.

After its previous mayor committed suicide last year, an investigation disclosed that town officials regularly voted

This sentence, after analysed by the RST parser module, will return the follow information to the sentiment classifier algorithm:

Circumstance Nucleus: [an investigation disclosed that town officials regularly voted]
Circumstance Satellite: [After its previous mayor committed suicide last year,]

Table 4.7 shows the total number of rules crafted for each relation and the number of sentences those rules matched in the SFU Review Corpus. Appendix A presents the entire list of patterns.

TABLE 4.7: Number of rules crafted for each relation and respective number of sentences matched by those rules in the SFU Review Corpus

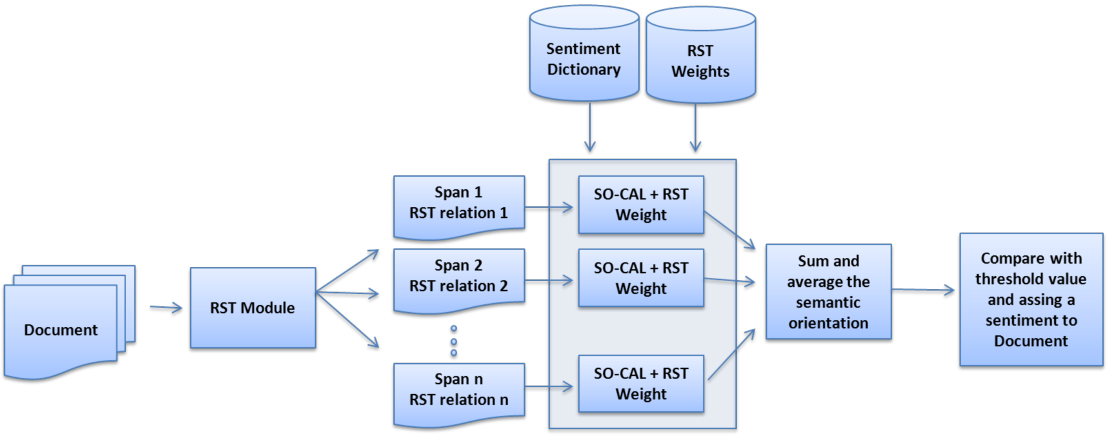
Relation	Number of Rules	Number of Sentences Matched
Anthitesis	6	227
Background	2	1776
Cause	3	388
Circumstance	3	256
Concession	4	206
Condition	3	480
Elaboration	2	76
Means	1	134
Purpose	1	52
Unless	1	35
Total	**26**	**3630**

4.3.2 Experiments and Results

In this experiment we incorporated the RST rules in a new module called RST module which was incorporated in the SO-RST algorithm. This new module substitutes the use of the RST corpus. The new diagram with the RST module for the SO-RST algorithm is shown in the Figure 4.1.

The Figure 4.1 is based on the Figure 3.1 previously described. Instead of the input source is a document annotated with RST followed by a module to extract

FIGURE 4.1: Diagram showing the steps followed by the SO-RST algorithm with the RST Module



these RST structures, this new diagram have a simple document as input followed by the RST module.

In this experiment, 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 (Table 4.3b) 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 (Table 4.6) in the previous experiment. In this scenario, 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.

Table 4.8 presents the relations and the weights attributed in both scenarios.

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

TABLE 4.8: Weights assigned to each RST relation in the algorithm proposed

a) **Scenario 1**

Relation	Weight
Anthithesis	1
Background	1
Cause	1
Circumstance	3.37
Concession	0.2
Condition	3.23
Elaboration	4.39
Means	3.22
Purpose	1
Unless	1

b) **Scenario 2**

Relation	Weight Nucleus	Weight Satellite
Anthitesis	1	1
Background	1	1
Cause	1	1
Circumstance	3.26	1
Concession	1	1
Condition	1	4
Elaboration	3.28	3.18
Means	3.27	1
Purpose	1	1
Unless	1	1

a weight to the RST relation (weight = 1). The results for both scenarios are summarized in the Tables 4.9 and 4.10.

TABLE 4.9: 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%

Our results show inconsistent results for both corpus. In SFU Review Corpus, the SO-RST achieved 74.06% of accuracy with the weights from scenario 1 and

TABLE 4.10: 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%

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.

Some of the factors which leads us to believe the results were not conclusive are:

- the patterns crafted cover only a small set of the discourse phenomena which occurs in the text;
- the patterns crafted do not cover all the important RST relations;
- some relations which received a high weight in the first experiment were not covered by the patterns or had few instances recognized;

This experiment showed an approach which incorporates a shallow discourse analysis module in the lexicon-based sentiment analysis. We believe this work showed the initial directions to take in this sense. Future works may focus on better discourse parser strategies or in the use of automatic RST parsers.

Chapter 5

Discussion and Conclusion

This chapter presents a discussion about the results achieved with this work and outlines directions for future works.

5.1 Summary

The Introduction chapter defined the objective of this work as improving lexicon-based sentiment analysis using the discourse knowledge. The following questions were entitled to be answered:

- Can discourse knowledge help lexicon-based sentiment classifiers?
- Which RST relations are more important for lexicon-based sentiment classification?
- How to incorporate those important relations into the classifier algorithm?

By Question 1, this study wants to determinate if the discourse structure present in text gives additional information to the classifier which helps it in the classification process. By Question 2, this study wants to determinate which relations in the RST theory better contribute to the classification process. The importance of a relation is measured by how likely the sentences under this relation indicates the polarity of the text. Question 3 asks how those important relations can be incorporated in a sentiment classifier.

This work proposed the algorithm SO-RST. This algorithm uses the discourse knowledge of texts in a lexicon-based sentiment classification. In this algorithm, the discourse knowledge was modelled with the Rhetorical Structure Theory (RST). The RST theory defines 26 relations responsible to link hierarchically the discourse structure of a text. Each relation connects a pair of text spans.

The SO-RST algorithm uses the dictionary to compute the semantic orientation of individual words. This orientation is then modified by the presence of negation, intensifiers and *irrealis*. The final semantic orientation of a sentence is computed by the sum of the semantic orientation presented in the individual words. The category which the text is classified (positive or negative) relies on the averaged sum for the semantic orientation of the sentences. If the semantic orientation value is below a threshold, the text is negative, otherwise positive.

To incorporate the discourse knowledge, we introduced an additional step in the SO-CAL algorithm. The SO-RST algorithm assigns a distinct weight for each RST relation. This weight ranges from 0 to 5. In the algorithm, this value is applied to the semantic orientation of the words under the scope of these relations. The purpose of the weight is to emphasize or downplay the importance of the sentences under particular relations.

In order to verify the adequacy of the proposed algorithm we conducted two experiments. The first experiment aimed to discover the best weights for each RST relation. We conducted an experiment with the SFU Epinions Corpus (Taboada et al., 2006; Taboada and Grieve, 2004) annotated with RST to discover which relations have more impact in the SO-RST algorithm proposed. In order to learn this weights we used a genetic algorithm heuristic and a 4-fold cross-validation.

This first experiment showed that RST relations can improve lexicon-based sentiment classification. It also showed which relations are the most important in the classification process. In order to incorporate this knowledge about the discourse in the SO-RST we executed a second experiment.

In the second experiment, a shallow RST parser module was incorporated in the SO-RST. This module uses the lexicon discourse markers present in the text to identify RST relations. The purpose of this module is to provide independence from a RST annotated corpus.

In this second experiment, the SO-RST used the weights learned from the first experiment. Two corpora were used to compute the accuracy measure: SFU Reviews corpus and the Movie Reviews 2 (Pang and Lee, 2004). The results showed the necessity of more tests with the RST module.

5.2 Discussion

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 approach the discourse structure (Somasundaran, 2010; Taboada et al., 2008), this work gave more support to the claim that the discourse structure is relevant to sentiment classification. The novelty here 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 excludes 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.

5.3 Future Work

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.

References

Andreevskaia, A. and Bergler, S. (2006), Mining wordnet for fuzzy sentiment: Sentiment tag extraction from wordnet glosses, *in* 'Proceedings of EACL', Vol. 6, pp. 209–216.

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* .

Choi, Y. and Cardie, C. (2008), Learning with compositional semantics as structural inference for subsentential sentiment analysis, *in* 'Proceedings of the Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, pp. 793–801.

Church, K. and Hanks, P. (1990), "Word association norms, mutual information, and lexicography", *Computational linguistics* , Vol. 16, MIT Press, pp. 22–29.

Das, S. and Chen, M. (2001), Yahoo! for amazon: Extracting market sentiment from stock message boards, *in* 'Asia Pacific Finance Association Annual Conf. (APFA)'.

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.

Ding, X., Liu, B. and Yu, P. (2008), A holistic lexicon-based approach to opinion mining, *in* 'Proceedings of the international conference on Web search and web data mining', ACM, pp. 231–240.

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

Esuli, A. and Sebastiani, F. (2006*a*), Determining term subjectivity and term orientation for opinion mining, *in* 'Proceedings the 11th Meeting of the European Chapter of the Association for Computational Linguistics (EACL-2006)', pp. 193–200.

Esuli, A. and Sebastiani, F. (2006*b*), SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining, *in* 'Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06)', pp. 417–422.

Fellbaum, C. (1998), *WordNet: An electronic lexical database*, The MIT press.

Grosz, B. J. and Sidner, C. L. (1986), "Attention, intentions, and the structure of discourse", *Computational Linguistics* , Vol. 12, MIT Press, pp. 175–204.

Hatzivassiloglou, V. and McKeown, K. R. (1997), Predicting the semantic orientation of adjectives, *in* 'Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics', ACL '98, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 174–181.

Hu, M. and Liu, B. (2004), Mining and summarizing customer reviews, *in* 'KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, New York, NY, USA, pp. 168–177.

Jordan, M. P. (1992), An integrated three-pronged analysis of a fund-raising letter, *in* W. C. Mann and S. A. Thompson, eds, 'Discourse Description Diverse Linguistic Analyses of a FundRaising Text', Amsterdam/Philadelphia: John Benjamins, pp. 171–226.

Kehler, A. (2002), "Coherence, Reference and the Theory of Grammar".

Kennedy, A. and Inkpen, D. (2006), "Sentiment classification of movie reviews using contextual valence shifters", *Computational Intelligence* , Vol. 22, pp. 110–125.

Kim, S. and Hovy, E. (2004), Determining the sentiment of opinions, *in* 'Proceedings of the 20th international conference on Computational Linguistics', Association for Computational Linguistics, pp. 1367–es.

Liu, B. (2009), *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*, 1st ed. 2007. corr. 2nd printing edn, Springer.

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", *Text-Interdisciplinary Journal for the Study of Discourse* , Vol. 8, Walter de Gruyter, Berlin/New York Berlin, New York, pp. 243–281.

Mao, Y. and Lebanon, G. (2006), Sequential models for sentiment prediction, *in* 'ICML Workshop on Learning in Structured Output Spaces'.

Marcu, D. (2000), "The theory and practice of discourse parsing and summarization", *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 classification using machine learning techniques, *in* 'Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10', EMNLP '02, Association for Computational Linguistics, Morristown, NJ, USA, pp. 79–86.

Pardo, T. and 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", *Computing attitude and affect in text: Theory and applications* , Springer, pp. 1–10.

Qiu, G., Liu, B., Bu, J. and Chen, C. (2009), Expanding domain sentiment lexicon through double propagation, *in* 'International Joint Conference on Artificial Intelligence (IJCAI-09)'.

Reiter, E. and Dale, R. (2000), *Building Natural Language Generation Systems*, Vol. 27 of *Studies in Natural Language Processing*, Cambridge University Press.

Sebastiani, F. (2002), "Machine learning in automated text categorization", *ACM computing surveys (CSUR)* .

Siegler, M. (2010), 'Eric Schmidt: Every 2 Days We Create As Much Information As We Did Up To 2003'.
URL: *http://techcrunch.com/2010/08/04/schmidt-data/*

Snow, S. (2011), 'How much content is on the web?'.
URL: *http://contently.com/blog/how-much-content-is-on-the-web/*

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 Computational 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.

Turney, P. D. (2002), 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 '02, Association for Computational Linguistics, Morristown, NJ, USA, pp. 417–424.

Wiebe, J., Wilson, T. and Cardie, C. (2005), "Annotating Expressions of Opinions and Emotions in Language", *Language Resources and Evaluation* , Vol. 39, Springer, pp. 165–210.

Wilson, T., Wiebe, J. and Hoffmann, P. (2009), "Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis", *Computational Linguistics* , Vol. 35, MIT Press, pp. 399–433.

Apendix A - Discourse Patterns Crafted

```
Example: [Maggie Gyllenhaal delivers an excellent performance]
         [although she is not the scene-stealer many are saying .]
rule = 10
relation = "ANTITHESIS"
pattern = "(?P<N>.{20,})(?P<S> although/.+)$"
```

```
Example: [Instead of separate shampoo and conditioner,]
         [the Benson offers a " conditioning shampoo " ,
          which is evidently a bad thing.]
rule = 11
relation = "ANTITHESIS"
pattern = "(?P<S>instead/.+?,)(?P<N>.+)$"
```

```
Example: [These are factoids]
         [, NOT descriptive passages !]
rule = 12
relation = "ANTITHESIS"
pattern = "(?P<N>.+)(?P<S>,/, not/.+)$"
```

```
Example: [I thought I needed a new stove]
         [, instead a new set of pans was a better deal.]
rule = 13
relation = "ANTITHESIS"
pattern = "(?P<N>.+)(?P<S>,/, instead/.+)$"
```

```
Example:  [Rather than relying on conventional clues,]
          [the mystery follows a trail of riddles and symbols.]
rule = 14
relation = "ANTITHESIS"
pattern = "(?P<S>rather/rb than/.+?,/,)(?P<N>.+)$"
```

```
Example: [General Motors uses metric terms for its automobile bodies
          and power trains,]
         [however, items such as wheelbases are still described in inches.]
rule = 15
relation = "ANTITHESIS"
pattern = "(?P<N>.{20,})(?P<S> however/.+)$"
```

```
Example: [except that Byrnes is an outsider to politics and to Washington ]
         [ ( he is a former CEO of Ford or GM or something ) ;]
rule = 20
relation = "BACKGROUND"
pattern = "(?P<N>.+)(?P<S>\(.+\)[^a-z]*)$"
```

```
Example:  [I 've had the phone for about a week]
          [and so far I 'm very impressed .]
rule = 21
relation = "BACKGROUND"
pattern = "(?P<S>.+)(?P<N>,/, and/.+)$"
```

```
Example: [I enjoyed it]
         [because many of the clues involved two of my favorite subjects :
         art history and theology .]
rule = 30
relation = "CAUSE"
pattern = "(?P<N>.+)(?P<S> because/.+)$"
```

```
Example: [We only missed one day of seeing her]
         [ due to snow]
rule = 31
relation = "CAUSE"
pattern = "(?P<N>.+)(?P<S> due/jj to/.+)$"
```

```
Example: [Since founding the company,]
         [the charismatic Vietnam vet, who is still only 46 years old,
          has fostered an ethos of combat.]
rule = 32
relation = "CAUSE"
pattern = "(?P<S>since/.+?,/,)(?P<N>.+)$"
```

```
Example: [After its previous mayor committed suicide last year,]
```

```
         [an investigation disclosed that town officials regularly voted]
rule = 40
relation = "CIRCUMSTANCE"
pattern = "(?P<S>after/.+?,/,)(?P<N>.+)$"


Example: [ Our All-Clad have taken a real beating]
         [since we purchased them .]
rule = 41
relation = "CIRCUMSTANCE"
pattern = "(?P<N>.+)(?P<S> since/.+)$"


Example: [once you get up to speed,]
         [the car functions decently , though not perfectly .]
rule = 42
relation = "CIRCUMSTANCE"
pattern = "(?P<S>once/.+?,/,)(?P<N>.+)$"


Example:  [Despite their considerable incomes and assets,]
          [40% of the respondents in the study don't feel
           financially secure]
rule = 50
relation = "CONCESSION"
pattern = "(?P<S>despite/.+?,/,)(?P<N>.+)$"


Example: [Although shot in the chest,]
         [he manages to disrobe]
rule = 51
```

```
relation = "CONCESSION"
pattern = "(?P<S>although/.+?,/,)(?P<N>.+)$"


Example: [Bakingrowning is also possible]
         [though I would never bake in them myself .]
rule = 52
relation = "CONCESSION"
pattern = "(?P<N>.+)(?P<S> though/.{20,})$"


Example: [Though not quite as good as a HDTV,]
         [but not too bad either .]
rule = 53
relation = "CONCESSION"
pattern = "(?P<S>though/.+?,/,)(?P<N>.+)$"


Example: [forgiven on a monthly pro-rata basis]
         [as long as the owner remains the occupant.]
rule = 60
relation = "CONDITION"
pattern = "(?P<N>.+)(?P<S> as/in long/rb as/.+)$"


Example: [and you can go to the observation deck for free]
         [ if you eat there .]
rule = 61
relation = "CONDITION"
pattern = "(?P<N>.{20,})(?P<S> if/.+)$"
```

```
Example:  [ordered the institution to stop paying common stock dividends]
          [until its operations were on track.]
rule = 62
relation = "CONDITION"
pattern = "(?P<N>.+)(?P<S> until/.+)$"
```

```
Example: [Part of its charm as a film is its very charming
          cast of characters,]
         [especially the female leads . ]
rule = 70
relation = "ELABORATION"
pattern = "(?P<N>.+)(?P<S>,/, especially/.+)$"
```

```
Example: [The students began to ponder their own direction in life]
         [including Betty]
rule = 71
relation = "ELABORATION"
pattern = "(?P<N>.+)(?P<S> including/.+)$"
```

```
Example: [Maybe she could step across the Plaza to the Met and
          help out her Czech compatriot/NN]
         [by singing the slow parts of Traviata.]
rule = 120
relation = "MEANS"
pattern = "(?P<N>.+/nn.?)(?P<S> by/.+)$"
```

```
Example: [the director explained that her fondest artistic wish
          was to find a way to play Somewhere Over the Rainbow]
         [so/IN that the song's original beauty comes through,
          surmounting the cliche.]
rule = 150
relation = "PURPOSE"
pattern = "(?P<N>.+)(?P<S> so/in.{20,})$"
```

```
Example:  [engine lacks torque to hold it]
          [unless road is flat .]
rule = 170
relation = "UNLESS"
pattern = "(?P<N>.+)(?P<S> unless/.+)$"
```