UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

DANNY SUAREZ VARGAS

Detecting Contrastive Sentences for Sentiment Analysis

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação

Orientador: Prof. Dr. Viviane Moreira

Porto Alegre
2016

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Suarez Vargas, Danny

Detecting Contrastive Sentences for Sentiment Analysis / Danny Suarez Vargas. – Porto Alegre: PPGC da UFRGS, 2016.

66 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2016. Orientador: Viviane Moreira.

1. Sentiment Analysis. 2. Contradiction Analysis. I. Moreira, Viviane. II. Título.

"If I have seen farther than others,
it is because I stood on the shoulders of giants."
— Sir Isaac Newton

ACKNOWLEDGMENTS

Firstly, I want to thank every member of my family. My father Juan for preparing me for the challenges of life. My mother Fortunata for being the example of person who I follow. My niece Paula for having been the happiness of my home in difficult days. And my sisters for their words of encouragement.

I would also like to thank the teachers who are and have been part of my academic life. To my Professor Julio Garcia Rivadeneyra for appearing at the right time, for believing in me, and for helping me start this long journey. My advisor Dr. Viviane P. Moreira, for having placed their trust in me and for helping me to achieve my goals.

I thank the good friends which life has given me. Thanks to Jose Martin Lozano Aparicio for being like a brother for me. Thanks to Edimar Manica for the companionship, and for the countless conversations during these years.

Finally, I thank the academic institutions that have allowed me to achieve my goals. The National University San Antonio Abad of Cusco and the Federal University of Rio Grande do Sul.

ABSTRACT

Contradiction Analysis is a relatively new multidisciplinary and complex area with the main goal of identifying contradictory pieces of text. It can be addressed from the perspectives of different research areas such as Natural Language Processing, Opinion Mining, Information Retrieval, and Information Extraction. This work focuses on the problem of detecting sentiment-based contradictions which occur in the sentences of a given review text. Unlike other types of contradictions, the detection of sentiment-based contradictions can be tackled as a post-processing step in the traditional sentiment analysis task. In this context, we make two main contributions. The first is an exploratory study of the classification task, in which we identify and use different tools and resources. Our second contribution is adapting and extending an existing contradiction analysis framework by filtering its results to remove the reviews that are erroneously labeled as contradictory. The filtering method is based on two simple term similarity algorithms. An experimental evaluation on real product reviews has shown proportional improvements of up to 30% in classification accuracy and 26% in the precision of contradiction detection.

Keywords: Sentiment Analysis. Contradiction Analysis.

Detecção de Sentenças Contrastantes através de Análise de Sentimentos

RESUMO

A análise de contradições é uma área relativamente nova, multidisciplinar e complexa que tem por objetivo principal identificar pedaços contraditórios de texto. Ela pode ser abordada a partir das perspectivas de diferentes áreas de pesquisa, tais como processamento de linguagem natural, mineração de opinioes, recuperação de informações e extração de Informações. Este trabalho foca no problema de detectar contradições em textos – mais especificamente, nas contradições que são o resultado da diversidade de sentimentos entre as sentenças de um determinado texto. Ao contrário de outros tipos de contradições, a detecção de contradições baseada em sentimentos pode ser abordada como uma etapa de pós-processamento na tarefa tradicional de análise de sentimentos. Neste contexto, este trabalho apresenta duas contribuições principais. A primeira é um estudo exploratório da tarefa de classificação, na qual identificamos e usamos diferentes ferramentas e recursos. A segunda contribuição é a adaptação e a extensão de um framework de análise contradição existente, filtrando seus resultados para remover os comentários erroneamente rotulados como contraditórios. O método de filtragem baseia-se em dois algoritmos simples de similaridade entre palavras. Uma avaliação experimental em comentários sobre produtos reais mostrou melhorias proporcionais de até 30 % na acurácia da classificação e 26 % na precisão da detecção de contradições.

Palavras-chave: Análise de Sentimentos, Análise de Contradições.

LIST OF ABBREVIATIONS AND ACRONYMS

CBOW	*Continuous Bag-of-words*
COS	*Contrastive Opinion Summarization*
NLP	*Natural Language Processing*
POS	*Part-of-Speech*
RNTN	*Recursive Neural Tensor Network*
WEKA	*Waikato Environment for knowledge Analysis*

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Consulting the opinion of others during the decision-making process has always been a common practice in people’s lives. The goal is to confront points of view in the search for the best decision. At present, with more than a third of the world population having access to the Internet (MEEKER, 2015), this practice has moved to the virtual context, in which people interact with others through opinions. These opinions are usually expressed in the form of product reviews available on the Web. Sentiment Analysis (also known as Opinion Mining) focuses on this context in order to help people manage these reviews and produce or extract useful information. Review summaries regarding rating stars(1-5), or with respect to polarity orientation(positive, negative), or with respect to different attributes of a specific product or service on popular websites such as Amazon, or Tripadvisor are some typical application of sentiment analysis. Polarity classification (also called polarity detection or sentiment polarity classification) is one of the most important tasks in the Sentiment Analysis area. It can be viewed as a two or three-class classification problem in which the classes are {positive, negative} and {positive, neutral, negative} respectively. Furthermore, the classification can be performed at different levels of granularity – document, sentence, clause, or aspect level (LIU, 2012).

Another important field of study is *Contradiction Analysis* (also called *Contradiction Detection*). It is a relatively novel multidisciplinary and complex area which aims to solve the problem of detecting contrastive or contradictory texts among the texts of a given collection of texts. The difficulty of this problem arises mainly from three reasons. The absence of a clear definition of contradiction for each context in which it appears, the high diversity of features that contribute to the presence of contradictions, as well as the scarcity of annotated data to solve the problem through new approaches and for different contexts.

Contradiction analysis was addressed on the literature mainly from two approaches, *Machine Learning* (MARCU; ECHIHABI, 2002; HILLARD; OSTENDORF; SHRIBERG, 2003; GALLEY et al., 2004) and *Textual Entailment*. (MARNEFFE; RAFFERTY; MANNING, 2008; PADÓ et al., 2008; MIZUNO et al., 2012). The solutions proposed from these approaches are mainly characterized by their input data which consists of pairs of sentences. Each pair of sentences is processed by the solutions in order to determine whether there exists a contradiction between them. Only with these two approaches and their annotated data, it is not possible to solve all types of contradictions such as the contradictions that arise from the opposite sentiments in the user reviews (also called sentiment-based contradictions).

In this work, we propose a framework to detect sentiment-based contradictions in order to prove that it is possible to find contradictions from sentiment analysis. This solution addresses the problem of contradiction detection from the perspective of sentiment analysis which defines the contradictions as the diversity of sentiments in a given text. This type of contradiction can be identified by employing a method which we proposed as an adaptation of the work by Tsytsarau, Palpanas e Denecke (2011) which was built in three steps. The first step is polarity classification of reviews at the sentence-level. The second step measures the diversity of sentiments on the polarity classification results, and the final step is labeling the review as contradictory or not. We grouped these steps in two distinct processes: (i) classification and (ii) contradiction analysis. In relation to the first process, we performed an exploratory study that aims at identifying and using different resources and tools (such as Wordnet, Stanford NLP Toolkit, Weka). Regarding the second process, we adapted and extended Tsytsarau *et al.,*'s work to our context adding a filtering step. Filtering is based on simple similarity algorithms (which we also proposed) combined with an existing polarity classifier, namely the Recursive Neural Tensor Network (RNTN) (SOCHER et al., 2013)) which is the state-of-the-art in polarity classification.

In order to evaluate the performance of our proposed solution, two groups of experiments were performed. The first experiments evaluate the classification process, while the second group assesses contradiction analysis. The results represent promising improvements in relation to the baseline. More specifically, our similarity algorithms achieved improvements in accuracy ranging from 16 to 19% compared to a widely used baseline (*i.e.,* RNTN (Recursive Neural Tensor Network) (SOCHER et al., 2013). For contradiction analysis, the use of our additional filtering method brought proportional precision improvements of up to 30%.

The remainder of this work is organized as follows, Chapter 2 defines the main concepts of sentiment analysis and contradiction analysis; Chapter 3 revises the related literature; Chapter 4 presents the contributions of this dissertation; Chapter 5 details the experimental evaluation; Chapter 6 concludes the paper, pointing to future work.

2 BASIC CONCEPTS

This chapter is organized into three sections. The first section introduces the main concepts, tasks, and approaches to sentiment analysis. The second section introduces the problem, the most important features, the classification, as well as the approaches to solving the contradiction analysis problem. Finally, the third section presents a summary of the tools and resources that were used in the present work.

2.1 Sentiment Analysis

Sentiment Analysis is the field of study that deals with the sentiments expressed in a given text. Sentiments can appear in different ways. For instance, sentiments can be conveyed as opinions, evaluations, appraisals, or emotions towards entities such as products, services and their attributes (LIU, 2012).

Despite the terms Sentiment Analysis and Opinion Mining having first appeared in different areas (Natural Language Processing (DAS; CHEN, 2001) and Information Retrieval (DAVE; LAWRENCE; PENNOCK, 2003)), they have been used interchangeably as, in essence, they refer to the same task, the automatic processing of opinionated data such as on-line reviews in order to help users to understand vast amounts of information and make decisions over it. Polarity Classification, Subjectivity classification, Opinion Extraction, or Opinion Spam Detection are some of the well-known tasks that are under the umbrella of Sentiment Analysis or Opinion Mining. However, due to the fact that the earlier works developed in the area were focused mainly on the polarity classification task, there is a strong relation between the area and this task. Initially, the relation was of equivalence, but nowadays the polarity classification task is only one of the many tasks in the area. (PANG; LEE, 2008).

2.1.1 Subjectivity, Opinion, and Sentiment

In a scenario in which people share their opinions and care about the opinions of others, comes the need of dealing with terms like *opinion*, *sentiment*, and *subjectiv-*

ity (PANG; LEE, 2008). These terms need to be clearly defined in order to allow for an understanding of the subsequent definitions of the present work. In the online Oxford dictionary (OXFORD. . . , 2016), *opinion* is defined as “A view or judgment formed about something, not necessarily based on fact or knowledge", *sentiment* is defined as “A view of or attitude toward a situation or event", and *subjectivity* is defined as “The judgment based on individual personal impressions and feelings and opinions rather than external facts". Based on these definitions and from the research point of view, we can say that *opinion* is a personal impression about some topic or their aspects, generally written in subjective language, where *subjective language* is the language used to express evaluations, emotions, and speculations (WIEBE et al., 2004); *the sentiment* can be considered as the orientation of a given opinion which can be positive, negative, or neutral; and *the subjectivity* is the concept over which the main source of data for sentiment analysis is obtained and also the concept over which the *subjectivity classification* is defined. Subjectivity classification allows us to separate the opinionated-data from the non-opinionated-data. Opinionated data is characterized by the presence of opinion statements inside it. There is plenty of opinionated-data on the Web in the form of user reviews. These reviews represent the main source of data for the works developed in the area of Sentiment Analysis. In the present work, we also use this type of data in our experiments. Below we present three examples of reviews, for a given topic, over which we will identify the main elements in the context of Sentiment Analysis.

1. “I am in love with this app! The best app ever."
   Very-positive –> 5 stars

2. “Bad new version does not appear to offer any new useful capabilities."
   Negative –> 2 stars

3. “The app is not bad."
   Neutral –> 3 stars

The first element of a review is the *topic* over which the opinion is expressed, which is also called as entity or target by Liu (2012). In the three previous examples, the topic is the same “The app". The second element of a review is the sentiment, or polarity, that it express (LIU, 2012). This sentiment can take one of three labels (negative, positive, or neutral) or one of five labels (very-negative, negative, neutral, positive, very-positive). Numeric rating scores are also possible. In example (1) the sentiment is the label “very-positive" or the rating score “5 stars". In example (2) the sentiment is the label “Negative" or the rating score “2". And in example (3) the sentiment is the label “Neutral" or the rating score “3 stars". There are other important elements that a given review contains, as defined by Liu (2012), such as the aspects or attributes of the entity or the date in which the review was published. However, these elements are not considered in the present work.

After defining some important terms and identifying the elements of a given user review, we present a classification of works developed in the area. This classification is based on the level of the analysis and was proposed by Liu (2012). The author identified three levels of granularity based on language constructs: document, sentences, and clauses and one level of granularity that is regardless of these constructs by looking directly at the opinion itself.

Document level.- This level of analysis has the language construct *document* as the minimal unit of the analysis and assumes that the opinions expressed in a given document refer to a single topic (TURNEY, 2002).

Sentence level.- This level of the analysis has the language construct *sentence* as the minimal unit of the analysis and assumes that it is possible to assign an overall sentiment value to each sentence of the given input data (RILOFF; WIEBE; PHILLIPS, 2005).

Clause level.- This level of analysis considers the possibility that a given sentence may contain more than one opinion on it or may contain opinionated as well as non-opinionated information. So, in order to address this type of sentences, the language construction *clause* is considered as the minimal unit of the analysis at this level (WILSON; WIEBE; HWA, 2004).

Aspect level.- This level of the analysis is also called as feature level. The main difference between this and the previous levels is that, instead of considering only the overall topic as a target of the expressed opinions, the analysis considers the topic and their attributes in order to generate a more fine-grained analysis (HU; LIU, 2004).

2.1.2 Polarity Detection

The polarity detection (also called polarity classification or sentiment polarity classification) is one of the most important tasks in Sentiment Analysis. Assuming that the overall opinion in an input document is about one single topic, polarity detection can be defined as a two, three, or five class classification task.

Two-Class Classification.- The task is defined as a binary classification which assigns one of two possible labels to each document of the input data(positive or negative) (PANG; LEE, 2008).
Three-Class Classification.- The number of classes also considers the *neutral* (RAVI; RAVI, 2015) class which is characterized by the absence of opinion on the input document (LIU, 2012) or by the presence of opposite opinions which return a neutral polarity value when they are aggregated (TSYTSARAU; PALPANAS; DENECKE, 2011).
Five-class Classification.- This task considers two degrees of negative, the neutral, and two degrees of positive as classes (ACAMPORA; COSMA, 2014). This type of work can also be considered as rating prediction task in which the classes: emotional negative (-2), rational negative (-1), neutral (0), rational positive (1), and emotional positive (2) are obtained based on the two types of user evaluations (rational and emotional) (LIU, 2012).

2.1.3 Approaches for Sentiment Analysis

Tsytsarau e Palpanas (2012) states that the proposed techniques to address the Sentiment Analysis tasks can be formulated as a three-step method : *Identify*, *Classify*, and *Aggregate*. Furthermore, it classifies the works developed on the area into four different approaches: *Machine-Learning*, *Dictionary*, *Statistical*, and *Semantic* which are defined as follows.

Machine-Learning Approach.- Given a dataset D which is divided into two different groups: *training data* and *test data*, the machine learning solutions are characterized by two sequential steps. The first step consists of learning a model based on the training data and the second step consists in classifying the test data based on the learned model. The machine learning algorithms and the datasets developed for them are important factors that contributed to the growth of the research on the Sentiment Analysis Area (PANG; LEE, 2008). However, the results of the solutions proposed from this approach are strongly dependent on the quantity and quality of training data. This dependency represents a problem specifically when the labeled data (*i.e.,* data annotated with the expected class) is scarce compared to the unlabeled data (TSYTSARAU; PALPANAS, 2012).

Dictionary Approach.- This approach proposes solutions relying on pre-built dictionaries (or lexicons) containing the polarities of opinion words. The General Inquirer[1], and Sentiwordnet (ESULI; SEBASTIANI, 2006) are well-known resources of this approach. Given a sentence S of a document D which contains a set of words W, the solutions proposed from this approach generally obtain the polarity value of a sentence S by averaging the polarities of each word of W. The limitation of this approach when dealing with domain-specific contexts can be overcome by using a combination of dictionaries and machine learning methods (TSYTSARAU; PALPANAS, 2012).

Statistical Approach.- An important problem with the two previous approaches is that they are not adaptable to different domains. So, in order to achieve the adaptability, the statistical approach relies on the construction of corpus-specific-dictionaries. These dictionaries can be built, for example, by calculating posterior polarities using word co-occurrences (FAHRNI; KLENNER, 2008). Solutions proposed from this approach generally rely on the frequency and the context in which each word appears. This means if a given word occurs more frequently among positive/negative texts, then it has a positive/negative polarity. A well-known work developed with this approach is Turney (2002) which is based on the observation that similar opinion words frequently appear together in a corpus.

Semantic Approach.- The solutions developed from this approach are based on the different semantic relationships that exist between words and by using the sense information of words to overcome the problem of context. The most known semantic relations are Synonymy, Antonymy, and Meronymy (*i.e.,* a part-of relationship). The lexical database for English, *Wordnet* (UNIVERSITY, 2005; MILLER, 1995), is the most widely used resource in this approach and is introduced in more detail in Section 2.3.

2.2 Contradiction Analysis

The first approach to the contradiction analysis problem appeared in the speech recognition area (HILLARD; OSTENDORF; SHRIBERG, 2003; GALLEY et al., 2004) trying to find agreements and disagreements over audio files. The analysis of contradictions in text was addressed for the first time in Harabagiu, Hickl e Lacatusu (2006). The

[1] Available at:<http://www.wjh.harvard.edu/~inquirer/>

authors identified contradictions using lexical, negation, and contrast features as well as an alignment text tool. Later, Marneffe, Rafferty e Manning (2008) contributed with a definition of contradiction for the Natural Language Processing area and described a classification of contradictions based on the features which characterize them.

Given two texts T_1, T_2 and a topic or event E, Harabagiu, Hickl e Lacatusu (2006) stated that the incompatibility between the information conveyed by T_1 and the information conveyed by T_2 is enough to affirm the existence of contradiction between T_1 and T_2. Two years later, Marneffe, Rafferty e Manning (2008) improves the previous definition by adding the condition of the event co-reference which says that the contradiction only occurs when the information conveyed by two different texts refers to the same event (or entity).

Contradiction Analysis is a relatively novel multidisciplinary and complex area which combines Natural Language Processing (NLP), Opinion Mining, Information Retrieval (IR) and Information Extraction (IE). The literature has diverse definitions of Contradiction Analysis, as each author defined it according to the specific problem that they were trying to solve. For example, Padó et al. (2008) which implement the contradiction detection system developed by Marneffe, Rafferty e Manning (2008), define the problem as a *textual entailment problem* using textual alignment scores, co-referent events, and a logistic regression algorithm to decide whether the given two texts contradict each other or not. Ennals et al. (2010) address the problem as a search for conflicting topics on the Web through texts patterns like “It is not correct that...". Tsytsarau, Palpanas e Denecke (2011) define the contradiction as a form of *sentiment* diversity and stated that there is a contradiction regard topic T, when there are conflicting opinions on T.

A more generic definition for contradiction was established in Tsytsarau e Palpanas (2012) which stated that, given a topic T, two sets of documents D_1, D_2, and a document collection D, where $D_1 \cap D_2 = \emptyset$ and D_1,$D_2 \subset D$. There is a contradiction on T when the information conveyed about T is more different between D_1 and D_2 than within each of them. However, the previous definition is only valid for contradictions that occur through different documents which in this work we call as *inter-document contradictions*.In this work, we address the problem we call *intra-document contradictions*, in which we search contradictions through the sentences of a given review or document. So, we use the definition that unlike other definitions, it is characterized by the use of the polarity of sentences as a main cue to determine whether there exists a contradiction or not.

2.2.1 Contradiction Detection Features

In this section, we use the term *feature* in a broader sense, meaning the typical characteristics of contradictions (and not necessarily the attributes used by a machine learning classifier). There are different features that are used in the literature in order to detect contradictions in texts. Some of them are easy to detect while others are difficult to detect even for the humans.

Number, Date, and Time Features.- Given two sentences S_1, and S_2 which contain numeric, date, or time information, the mismatch of this information over the sentences can represent the existence of a contradiction (MARNEFFE; RAFFERTY; MANNING, 2008).

1. “The house has three floors"

2. “The house has four floors"

Assuming that the two sentences refer to the same house, there is a numeric mismatch problem regarding the number of floors.

Lexical Relation Features. There are different types of lexical relations, such as Synonymy, Antonymy, Hyponymy, Hypernymy, Meronymy, etc. Among these relations, antonymy, synonymy, and meronymy have been used in the literature as important cues to detect contradictions (MARNEFFE; RAFFERTY; MANNING, 2008; HARABAGIU; HICKL; LACATUSU, 2006; RITTER et al., 2008; KAWAHARA; KUROHASHI; INUI, 2008).

3. “James starts his homework"

4. “James finishes his homework"

In the example above, there is certainly a contradiction between sentences. The opposite meanings of the words “start" and “finish" represent a feature to take into account on the contradiction analysis.

Structural Features.- Given two sentences and their syntactic structures, it is possible to determine the role of each word within the sentences. Based on the role of each word, it is possible to determine whether there exists a conflict between sentences (MARNEFFE; RAFFERTY; MANNING, 2008).

5. “Jose defeated Pedro"

6. “Pedro defeated Jose"

Sentences 5 and 6 have the same structure but the roles of subject and direct object are inverted. Thus, the meanings of the sentences are not the same and represent a contradiction.

Factive Features.- These features appear in a sentence S which contains a factive construction F such as forget/remember/know... that or an implicative construction I such as forget/remember/manage/bother ... to. It is important to take their properties into account of them because they are capable of changing the meaning of "S" (NAIRN; CONDORAVDI; KARTTUNEN, 2006; MARNEFFE; RAFFERTY; MANNING, 2008).

7. “David *forgot to* send me a confirmation message"

8. “David sent me a confirmation message"

9. “David did not *forget to* send me a confirmation message"

10. “David sent me a confirmation message"

In the sentences (7), (8), (9), and (10) the presence and absence of the factive construction *forget to* changes the meaning of the sentences.

Modality Features.- Given a sentence S which contains a modal verb MV such as can, could, may, etc. MV is a good cue to infer the meaning of S. Handling MV correctly on the analysis of S can allow us, for example, to determine whether the statement of S is possible or not (MARNEFFE; RAFFERTY; MANNING, 2008).

11. “"The Facebook users may retrieve their personal information"

12. “The Facebook users may not retrieve their personal information"

The presence of the modal verb may and its negation help us to find the contradiction between (11) and (12).

Relational Features.- Given two sentences S_1 and S_2. The relations between elements of S_1 and the relations between the elements of S_2 are important to determine whether the sentences entail or contradict each other (MARNEFFE; RAFFERTY; MANNING, 2008).

13. “Diego lives in Liverpool"

14. “Diego dreams of traveling to Liverpool"

In the sentences above,“Diego" is the subject, “Liverpool" is the Location, and the relation “Live" of the sentence (13) is negated in the sentence (14). So, sentences (13) and (14) contradict each other.

Negation Features.- This group of features contains negation terms and phrases. Each of them is capable of reversing the polarity of the sentence or statement which contains it (KENNEDY; INKPEN, 2006; KAWAHARA; KUROHASHI; INUI, 2008). The negation features can be classified by their number of words: individual word such as ‘no’, ‘not’, ‘without’, ‘barely’, or multiword phrase such as ‘no where’, ‘by no means’, ‘at no time’ (JIA; YU; MENG, 2009) and by the ease with which they can be recognized: directly licensed such as overt negated markers, negative quantifiers, and strong negative adverbs and indirectly licensed such as negated events, negated entities, and negated states (HARABAGIU; HICKL; LACATUSU, 2006).

15. “The app is good"

16. “The app is not good"

Sentences (15) and (16) contradict each other by the presence of the negation term ’not’ in the sentence (16).

Intensifiers and Diminishers Features.- Given a sentence S and its polarity P, the intensifiers/diminishers increase/decrease the strength of S (KENNEDY; INKPEN, 2006). If the polarity value P is positive, the intensifier can change P to very positive and if the polarity value P is negative, the diminisher can change P to very negative.

17. “The app is good" –> 4 stars

18. “The app is very good" –> 5 stars

The valuation of products through a number of stars is a typical example of the impact of the intensifiers and diminishers in a given text. The intensifier word “very" in the sentence (18) increases its strength.

Contrasting Features.- This group of features is originated from the discourse relation of contrast. The contrast relation between two sentences appears when the situations presented in them are the same in many respects, different in some respects, and these situations can be compared with respect to the differences (MARCU; ECHIHABI, 2002).

19. “Carlos doesn’t have enough money to buy a ticket to see the show."

20. “Jose can buy a ticket to see the show."

The situation in the sentences above is the same respect to “buy a ticket" and different with regards to “Carlos cannot buy a ticket while Jose can". The most used way to identify the discourse relation of contrast is through the contrastive discourse markers, but it is also possible to identify the discourse relation in the absence of these markers (HARABAGIU; HICKL; LACATUSU, 2006) through semantic interpreters of Natural Language Processing such as discourse-relation classification system (MARCU; ECHIHABI, 2002). The discourse markers also called discourse connectives, discourse operators, pragmatic connectives, sentence connectives, and cue phrases are lexical expressions commonly represented by the syntactic classes of conjunctions, adverbs, and prepositional phrases (FRASER, 1999).

World Knowledge Features.- Some occurrences of contradiction can only be detected through world-knowledge. In order to detect this type of contradictions, one has to construct a knowledge base that allows inferring implicit information over the given sentences, statements or texts. The system performance will be directly proportional to the size of the world-knowledge base (MARNEFFE; RAFFERTY; MANNING, 2008; RITTER et al., 2008).

21. “Diego was born in France"

22. “Diego was born in Paris"

In order to decide whether exist a contradiction or not, it is necessary to have the knowledge that Paris is the capital of France.

2.2.2 Classification of Contradictions

Contradictions can be classified *based on their features*, based on the time in which they occur, and based on the context in which they are analyzed. The first classification was proposed by Marneffe, Rafferty e Manning (2008), who defined two groups of contradictions according to the complexity of the analysis which is required to detect them. The first group includes contradictions that can be identified by finding **antonyms, negation, or numeric mismatches**. The second group of contradictions can be identified by an analysis of **structure of assertions, discrepancies based on word-knowledge, and**

lexical contrast. For Marneffe, Rafferty e Manning (2008), analyzing and detecting contradiction from the second group is more complex than on the first group.

The second classification was proposed by Tsytsarau e Palpanas (2012), Tsytsarau, Palpanas e Denecke (2011) and implemented by (TSYTSARAU; PALPANAS; DENECKE, 2011). It defined two types of contradictions *based on the time* at which the contradiction arises. **Asynchronous contradiction** that arises through the time and **synchronous contradiction** that arises on a fixed time.

Finally, contradictions can be analyzed in *two contexts*. The first one is when the contradictions arise among different documents which we called as *inter-document contradictions* and the second one is when contradictions arise within a given document which we called *intra-Document contradictions* (TSYTSARAU; PALPANAS; DENECKE, 2011). Despite this terminology having not been used in the literature, we will use it in order to clearly establish the scope of our work.

2.2.3 Approaches

There are some frameworks, tools, and systems (HARABAGIU; HICKL; LACATUSU, 2006; KAWAHARA; KUROHASHI; INUI, 2008; MIZUNO et al., 2012) that were developed to solve the contradiction detection problem. Each of them tries to highlight the resources that were used in order to differentiate itself from others. These resources can be steps, methods, algorithms or features and it is possible to use them to group the solutions in approaches.

Machine Learning.- The solutions that pertain to this approach have as the main resource one or more machine learning algorithms of all available on the literature (MARCU; ECHIHABI, 2002; HILLARD; OSTENDORF; SHRIBERG, 2003; GALLEY et al., 2004).

Textual Entailment.- In this approach, the problem of contradiction detection is addressed as a textual entailment problem. Textual entailment is the task that, for two given text fragments, determines whether the meaning of one text can be inferred (entailed) from another text (DAGAN; GLICKMAN; MAGNINI, 2006). Some works use directly the output of textual entailment solutions (HARABAGIU; HICKL; LACATUSU, 2006), while others perform modifications of them (MARNEFFE; RAFFERTY; MANNING, 2008; PADÓ et al., 2008; MIZUNO et al., 2012)

Others.- Some works do not fit in the previous types of approaches such as *pattern-*

based (ENNALS et al., 2010; ENNALS; TRUSHKOWSKY; AGOSTA, 2010), *knowledge-based* (RITTER et al., 2008), and *sentiment analysis-based* (TSYTSARAU; PALPANAS; DENECKE, 2011; VARGAS; MOREIRA, 2015).

2.3 Tools and Resources

In order to develop the present work, some tools and resources were used. The lexical database Wordnet, The Stanford NLP toolkit, The implementation of Wor2vec, and the data mining resource Weka are introduced below.

2.3.1 Wordnet

Wordnet is a lexical database for English. It contains information about the words (word forms) and their meanings (senses)-more than 118K different words and more than 90K different word senses grouped based on their syntactic categories such as nouns, verbs, adjectives, adverbs and linked through the semantic relations such as synonymy, antonymy, hyponymy, meronymy, troponymy(see the table 2.1). The synonymy is the most important relation that is included in the Wordnet. There are 117K synsets linked between them by other relations such as hyponymy, meronymy or troponymy where a synset is a set of words that denote the same concept and are interchangeable in different contexts (UNIVERSITY, 2005; MILLER, 1995).

Table 2.1: Semantic Relations in WordNet

Semantic Relation	**Syntactic Category**	**Examples**
Synonymy (similar)	Noun, Verb, Adjective, Adverb	(pipe, tube), (rise, ascend)
Antonymy (opposite)	Adjective, Adverb, (Noun, Verb)	(wet, dry), (powerful, powerless)
Hyponymy (subordinate)	Noun	(sugar maple, maple maple)
Meronymy (part)	Noun	(brim,hat)
Troponymy (manner)	Verb	(march,walk)
Entailment (similar)	Verb	(drive,ride)

Source: Miller (1995)

2.3.2 Stanford NLP Toolkit

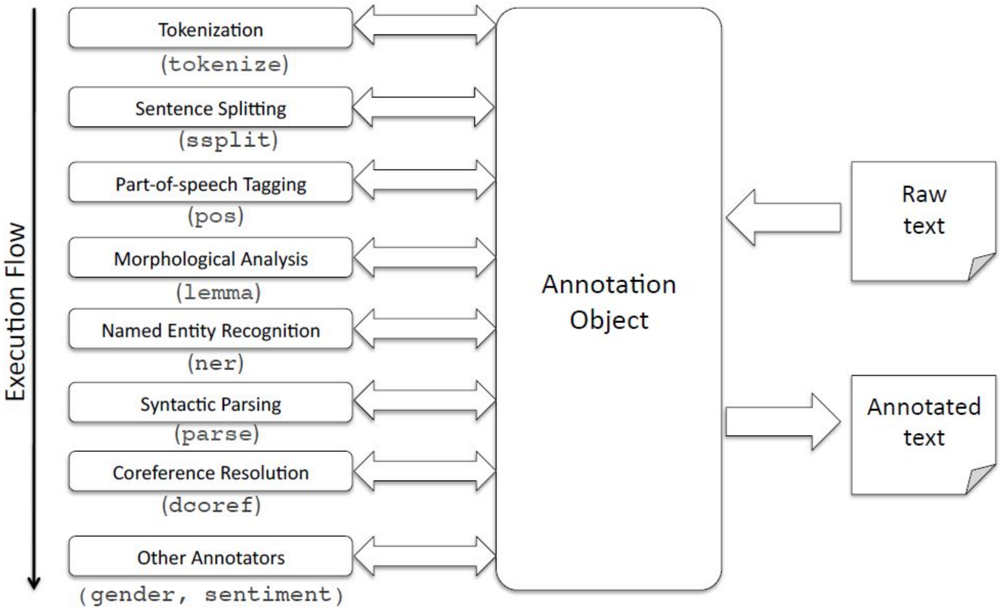
The Stanford CoreNLP toolkit (MANNING et al., 2014) is a Java API based on a straightforward pipeline architecture(see the figure 2.1). It executes most of the common Natural Language Processing tasks. These tasks are the well-known linguistic analysis processes such as tokenizing, splitting, part-of-speech tagging which are available as a form of sequential annotators. Each annotator is designed to perform a specific task and having as its output the information obtained through the process. This information can be essential for performing the next annotator of the pipeline, for example, it is necessary to perform the process of tokenization before performing the process of POS-tagging. The API provides a total of 12 annotators and some of them are defined below.

- **Tokenize** tokenizes a given text into sequence of tokens.
- **Cleanxml** removes the XML tags from a given document.
- **Ssplit** splits a given text into sentences.
- **Pos** assigns a syntactic category to each token of the input.
- **Lemma** lemmatizes the given input data.
- **Gender** adds gender information to names.
- **Sentiment** It performs the sentiment analysis process on a given input data and returns a score of five possible scores.

2.3.3 Word2Vec

Word2Vec (MIKOLOV et al., 2013) is the tool that results of the implementation of two model architectures: Continuous Bag-of-words model (CBOW) and Continuous skip-gram model. These models are used for computing continuous vector representations of words learned by neural networks. The first model allows us to predict the current word based on the *n* previous and future words and the second model allows us to predict neighboring words (words that appear before and after the current word) based on the current word (see Figure 2.2).

Figure 2.1: Architecture of the Stanford NLP Toolkit



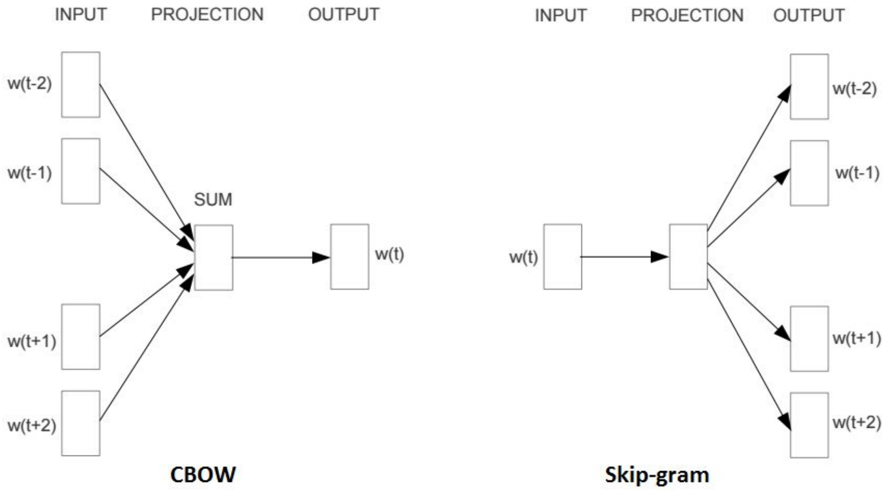
Source: Manning et al. (2014)

One important property of these models is that they preserve better the linear regularities between words compared to other models such as Latent Semantic Analysis (LSA) and reduces computational costs when working with large datasets with billions of words. Another important property is that it is possible to find many different types of similarities such as syntactic and semantic word relationships, for example, the question “What is the word that is similar to *cold* in the same sense as *hottest* is similar to *hot*?" can be formulated using simple algebraic operations of the vector representation of words as X = vector(“hottest") - vector(“hot") + vector(“cold") where X is the vector representation of the answer for the formulated question. So, based on the cosine distance, the word to answer the question is selected, for the example the expected answer is the word “coldest".

2.3.4 Weka

The Waikato Environment for knowledge Analysis (WEKA) (HALL et al., 2009) is an open source software designed to help data mining researchers and practitioners alike. WEKA has widespread acceptance through the academic and business circles mainly due to the fact that it is a modular extensible open source tool. It provides re-

Figure 2.2: Continuous Bag-of-words and Skip-gram model architectures



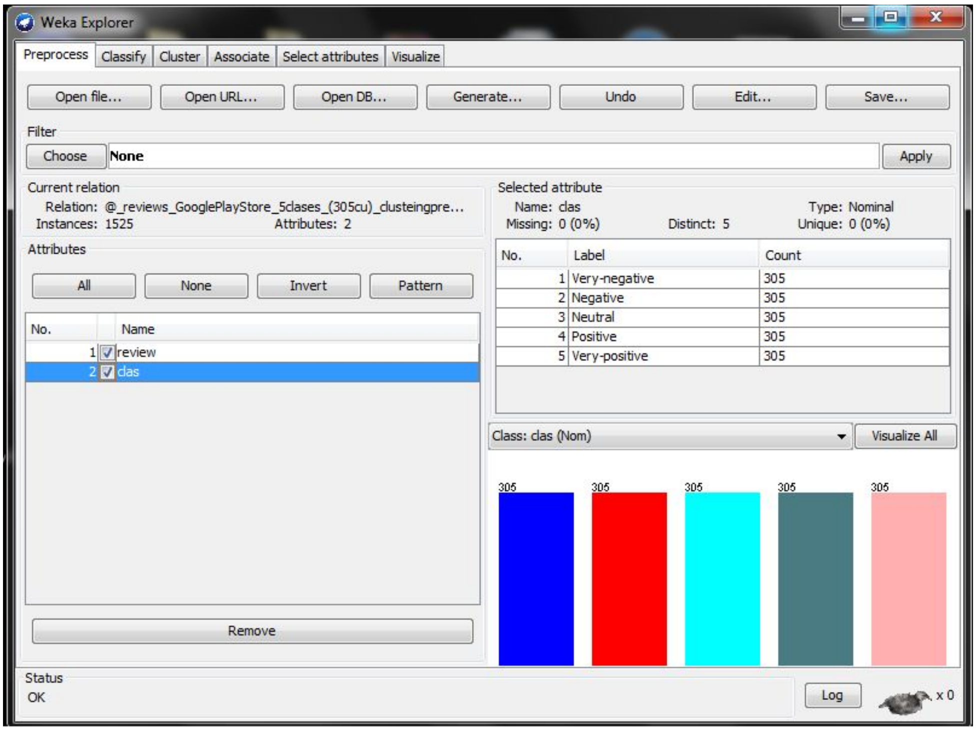
Source: Mikolov et al. (2013)

sources for data preprocessing and visualizing, machine learning algorithms, as well as the options to evaluate the performance of algorithms. WEKA has a well-structured and user-friendly interface. The main graphical user interface is the “Explorer" as we can see in Figure 2.3. It contains six graphical panels: *“Preprocess"*, *“Classify"*, *“Cluster"*, *“Associate"*, *“Select Attributes"*, and *“Visualize"* which allow performing a number of tasks needed in the data mining pipeline. The other important graphical user interface in WEKA is the “Experimenter". It aims at facilitating the automatic comparison of the performance of algorithms. This comparison can be performed on different datasets and based on different evaluation criteria that are available in WEKA.

2.3.5 RNTN Classifier

The Recursive Neural Tensor Network (RNTN) model (SOCHER et al., 2013) aims to capture the compositional effects of longer phrases in the task of sentiment detection. Similar to other models such as RNN (Recursive Neural Network), RNTN is a recursive neural model. This type of model is characterized by performing two operations in common, word vector representation, and classification. More specifically, this type of model, for a given n-gram input, parses it into a binary tree, represents each leaf node (corresponding to a word) through a vector, classifies each vector, uses these vectors to compute the parent vectors in a bottom-up fashion by using some compositional function,

Figure 2.3: The main graphical user interface of WEKA: Explorer



Filter: None

Current relation
Relation: @_reviews_GooglePlayStore_5clases_(305cu)_clusteingpre...
Instances: 1525 Attributes: 2

Attributes

No.		Name
1	☑	review
2	☑	clas

Selected attribute
Name: clas Type: Nominal
Missing: 0 (0%) Distinct: 5 Unique: 0 (0%)

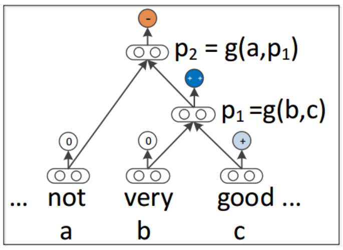
No.	Label	Count
1	Very-negative	305
2	Negative	305
3	Neutral	305
4	Positive	305
5	Very-positive	305

Class: clas (Nom)

Status
OK

performs again the classification for each parent vector, and so on recursively. We can see this process in Figure 2.4 and an RNTN classification example in Figure 2.5.

Figure 2.4: Approach of Recursive Neural Network models for sentiment

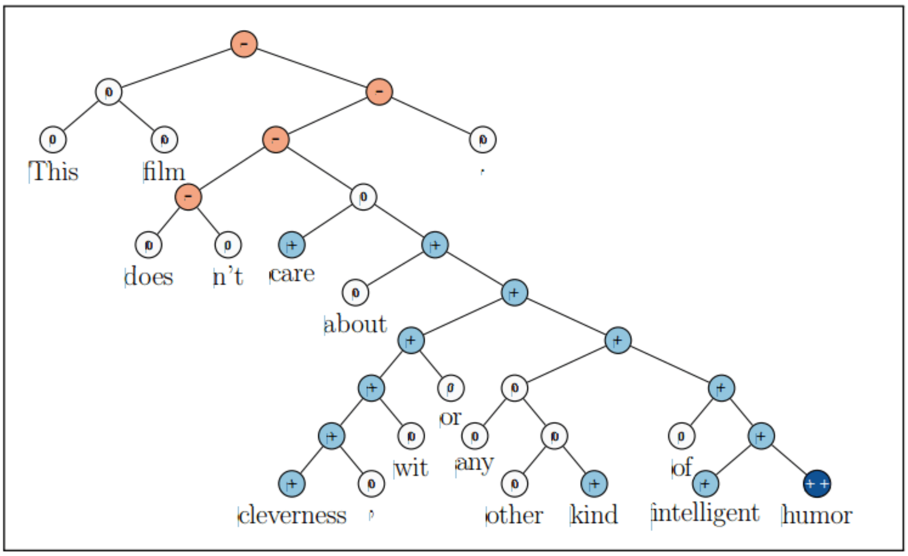


Source: Socher et al. (2013)

RNTN performs better than other neural networks that ignore word order and establishes the state of the art in the polarity classification task at sentence level by using the Stanford sentiment treebank. The Stanford sentiment treebank is a large and labeled compositional resource which consists in fine-grained sentiment labels for 215,154 phrases in the parse trees of 11,855 sentences about movie reviews.

In this chapter were introduced the main concepts, features, classification, and approaches of the sentiment analysis and contradiction analysis areas. Furthermore, the tools and resources such as Wordnet, Stanford NLP Toolkit, Word2vec, Weka, and RNTN

Figure 2.5: Example of the Recursive Neural Tensor Network for predicting 5 sentiment classes



Source: Socher et al. (2013)

classifier are briefly described as they are used in the subsequent chapters.

3 RELATED WORKS

This chapter describes the works that are closely related to this dissertation. These works are organized in two sections. The first section describes the existing literature related to sentiment analysis and the second section describes the works that address the contradiction analysis.

3.1 Works on Sentiment Analysis

Liu (2012) defines the problem of sentiment analysis, detailing it into sub-areas and levels of analysis. Then it presents a survey of the existing solutions in the literature, comparing the results obtained by them. In addition, it also introduces new and challenging research problems. Among the sections that are related to this dissertation are the polarity classification at sentence level and the opinion summarization. Tsytsarau e Palpanas (2012) is a little more specific, it structured well the sentiment analysis problem as a three-step process, classified the existing solutions into four approaches, and introduced opinion aggregation problems. Identify, Classify, and Aggregate are the three steps considered in Tsytsarau's survey. The first step aims at identifying topics or opinions. The second step aims at classifying documents, sentences, or words. And, the third step aims at performing an overall analysis, for example, the average or prevalent opinion of a group of people about some specific topic. The four approaches considered in Tsytsarau's survey are Machine learning, Dictionary, Statistical, and Semantic approach. The machine learning approach characterized by the availability of its different algorithms(supervised, unsupervised), its two main steps(training, and testing), and its high dependency on the quantity and quality of training data. The dictionary approach characterized by the use of pre-built dictionaries such as the General Inquirer[1] or Sentiwordnet (ESULI; SEBASTIANI, 2006), its methods that aggregate polarities of individual words in order to determine the polarity of a sentence or document, and its high domain dependency. The statistical approach characterized by the use of frequency and co-occurrence of words, the construction of domain-specific dictionaries, and the dependency on large enough corpus to overcome the problem of unavailability of words that exists in the previous approach. And the semantic approach characterized by its principle that semantically close words should receive similar sentiment values. A well-known ma-

[1] Available at:<http://www.wjh.harvard.edu/~inquirer/>

chine learning approach work was the proposed by Pang, Lee e Vaithyanathan (2002). In this work, the sentiment analysis was addressed as a two-topic text categorization problem or two-class classification problem (negative, positive). The Naive Bayes, Maximum Entropy, and Support Vector Machine algorithms were considered to perform the classification. The presence and frequency of unigrams, bigrams, part of speech, adjectives, and position were considered as features. From the experiments, the support vector machine algorithm by considering the presence of unigrams and bigrams obtained the best accuracy for three-fold cross-validation. Turney (2002) proposed a statistical solution to the classification of reviews as recommended or not recommended (positive, negative). This work is based on the hypothesis that if two words frequently appear together within the same context, they are likely to have the same polarity. This principle was implemented by considering the mutual information measure (CHURCH; HANKS, 1990). For a given phrase P, the semantic orientation(positive, negative) of P was calculated as the mutual information of P and the word “excellent" minus the mutual information between P and the word “poor". The words “excellent" and “poor" were selected as it represents the reviews with rating values of five stars and one star respectively, while the phrase P consists of two consecutive words. The first word can be an adjective, an adverb or a noun, while the second word can be a noun, an adjective or a verb. From the experiments performed on reviews of different domains, the best accuracy was achieved on the automobiles domain. Hu e Liu (2004) It is the well-known semantic approach work. The objective of this work was the creation of features-based summaries for product reviews. This work was structured in three main steps. The first step aimed to discover all features for a given product. The second step aimed to identify opinion words and sentences as well as their semantic orientation. And the final step aimed to summarize the results. The authors, in step 2, took advantage of the structure of the lexical database for English Wordnet which groups their words based on their syntactic categories (noun, adjective) and links them through the semantic relations (synonymy, antonymy). As we will see in the next chapter, we use the concepts of *intensifiers*, *diminishers*, and *negation* shifters, as well as *a lexicon-based method* in our proposed framework. Taboada et al. (2011) proposed a lexicon-based method for Sentiment Analysis. The authors identify intensification and negation as important features to be considered when extracting sentiments from reviews. Kennedy e Inkpen (2006) examined the contextual valence shifters (or only shifters): Negation, Intensifiers, and Diminishers, showing their impact on the sentiment classification task. The study found that by considering these shifters, classification accuracy can increase up to 85%. There are also works that address the impact of each type of shifter independently. Jia, Yu e Meng (2009) studied the impact of the negation shifters in the sentiment classification task. They identified the scope of negation and carried out experiments that showed this brings improvements in the accuracy of the classification. The main difference of our work in relation to these sentiment analysis works is that our goal is not strictly the construction of a classifier. Instead, we are looking for reviews with contrastive or contradictory sentences and the classifier can be considered as an important preliminary step to our objective.

3.2 Works on Contradiction Analysis

In this section, we describe in detail the two works on contradiction analysis that are most similar to this dissertation, and give a brief overview of other complementary works.

The work by Kim e Zhai (2009) defined a novel problem called *Contrastive Opinion Summarization (COS)* and proposed a framework to deal with it. This framework takes as input two sets of positive and negative sentences and returns as output a list of contrastive sentence pairs that are the most representative of the two input sets. COS was addressed as an optimization problem in which the objective function is based in two criteria *representativeness* and *contrastiveness*. In order to calculate the numerical value of the two criteria, two similarity functions were defined:*content similarity* and *contrastive similarity*. Furthermore, COS, the two criteria, the similarity functions as well as the optimization problem were mathematically defined.

Based on these definitions and for the set of positive and negative sentences X and Y, two algorithms were proposed to solve the optimization problem. The ***Representativeness First Approximation*** algorithm initially retrieves the k most *representative* sentences U, V from X and Y respectively by using a clustering algorithm. Meanwhile, the ***Contrastive first Approximation*** algorithm retrieves the k most *contrastive* sentences of each group of sentences X and Y by using a content similarity function.

Two datasets were used in their experiments. The first dataset consists of Amazon's reviews that were obtained from the literature[2].In order to assign the correct format to this dataset, two human annotators identified representative contrastive sentence pairs. Furthermore, the authors prepared another dataset by using the Yahoo! search engine.

[2] Available at:<https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html>

The data was obtained by querying for the topic “Aspartame". Then, the results were processed to construct positive and negative matching sentences. The *contrastiveness* and the *representativeness* of the resulting summary was evaluated by the *precision* and the *aspect coverage*, respectively. Both of them were based on the agreement between the human annotators and the algorithms.

From the results of the experiments, the authors concluded that it is easier to achieve high representativeness than to achieve high contrastiveness. So, the *contrastiveness-first approximation* algorithm should be selected in order to maximize the contrastiveness of the resulting summary. The highest precision and aspect coverage values were 0.540 and 0.804 respectively. The main difference between Kim’s work and ours is that we are not interested in creating any summary, instead, we process all input reviews looking for those with contradictions between their sentences.

The second most closely related work was performed by Tsytsarau, Palpanas e Denecke (2011) who proposed a novel approach to contradiction detection. This work was adapted as we explain later to be used as the baseline of our proposed framework. Unlike other works that define contradiction analysis as pairwise comparisons of texts (text, hypothesis), in that work, it was defined as the search for sentiment diversity on document collections related to one or more topics. Furthermore, contradictions were classified based on the time (Synchronous, Asynchronous) and on the context (Intra-Document, Inter-Document) in which they arise. In order to present their approach, the authors proposed a framework that defines concepts of aggregated sentiment (mean value), sentiment variance (variance), and contradiction. The sentiment S with respect to a topic T was defined as a real number in the range $[-1, 1]$ that indicates the polarity of the author’s opinion on T expressed in a text. The aggregated sentiment μ_s expressed in a collection of documents D on topic T, is defined as the mean value over all individual sentiments assigned in that collection. The contradiction on a given topic, T, between two groups of documents, D_1, $D_2 \subset D$ is defined in function of the information conveyed about T. From these definitions, the authors create a novel contradiction measure based on the mean value and variance, shown below.

$$C = \frac{nM_2 - M_1^2}{(\vartheta n^2 + M_1^2)} W \tag{3.1}$$

where n is the cardinality or the number of documents of the given document collection D. $M_1 = \sum_{i=1}^{n} S_i$ and $M_2 = \sum_{i=1}^{n} S_i^2$ are the first and second order moments of the topic sentiment which are based on the mean value μ_s and on variance σ^2 respectively. The small value $\vartheta \neq 0$ is used to limit the level of contradiction when μ^2 is close to zero; and W is a weight function which takes into account n of D to calculate C.

$$W = \left(1 + \exp(\frac{\bar{n} - n}{\beta})\right)^{-1} \tag{3.2}$$

where $\bar{n}$ is the average number of topic documents involved in the analysis and β is a scaling factor.

Based on these definitions, a three-step framework for contradiction detection was proposed. The first step of this framework consists in detecting topics for each sentence of the input data. The second step assigns a sentiment to each sentence-topic pair. Then, contradiction analysis is performed in the final step. An experimental analysis attempted to find contradictions on the topic “internet government control" considering reviews published in a time window of ten days. The authors show plots for the mean, variance and the contradiction measure over time. On an evaluation with human subjects, the authors found that users were able to identify contradictions faster with their method than when using a visual method proposed by Chen et al. (2006).

Among the differences between Tsytsarau’s work and ours, is the fact that while they look for contradictions that occur across different documents (inter-document), we look for contradictions that occur inside a single document (intra-document). The other difference is that, instead of only relying on the contradiction measure to detect contradictions, we consider an additional filtering process which is detailed later in Section 4.

Harabagiu, Hickl e Lacatusu (2006) proposed a framework for recognizing contradictions as a Textual Entailment problem. Features with information about contrast, semantic, pragmatic, and negation are considered to cast the text entailment as a classification problem. Marneffe, Rafferty e Manning (2008) mainly provides a definition of contradiction for the NLP area, a general classification of contradictions, and an available corpora to contradiction analysis systems.

The main difference of our work in relation to these works is the approach that they use and the type of contradictions that they were addressed.

4 DETECTING CONTRAST AND CONTRADICTION IN SENTIMENT ANALYSIS

This chapter presents the main contributions of this work, which can be summarized as :

1. An exploratory study of resources for the classification task.
2. An adapted and extended contradiction analysis framework which is based on the algorithms listed below.
   - An algorithm to determine the polarity orientation of texts at the sentence level, which relies on simple similarity algorithms combined with an existing polarity classifier.
   - A filtering algorithm to remove reviews that are labeled erroneously as contradictory, which improves the precision of the contradiction detection task.

The main differences of this work in relation to the existing approaches are the type of contradictions that are the goal of our work (intra-document sentiment-based contradictions), while other works address inter-document contradictions and/or do not focus on sentiment-based contradictions. The only work that deals with this type of contradictions was presented by Tsytsarau, Palpanas e Denecke (2011) which represents the baseline of this dissertation. Compared to this, our work has some differences such as the additional filtering process, the similarity algorithms, and our polarity orientation algorithm.

4.1 Problem Definition and Solution Overview

The detection of sentiment-based contradictions based on a contradiction measure was addressed earlier by Tsytsarau, Palpanas e Denecke (2011). Here, we adapt the definition of contradiction as well as the contradiction measure to our context as follows.

4.1.1 Sentiment-Based Contradiction

For a given review R, which contains two or more sentences $\{S_1$, S_2,...,$S_n\}$, and their polarity orientation values $\{P_1$, P_2,...,$P_n\}$ where $S_1 \neq S_2 ... \neq S_n$, R is considered a contrastive/contradictory review or contains contrastive/contradictory sentences when the *contradiction Measure* C of R exceeds a certain threshold ρ.

4.1.2 Contradiction Measure C

This measure assigns a contradiction value C to R as follows.

$$C = \frac{nM_2 - M_1^2}{(\vartheta n^2 + M_1^2)} W \tag{4.1}$$

where n is the cardinality or the number of sentences of R. $M_1 = \sum_{i=1}^{n} P_i$ and $M_2 = \sum_{i=1}^{n} P_i^2$ are the first and second order moments of the polarity values which are based on the mean value μ_s and on variance σ^2 respectively. The small value $\vartheta \neq 0$ is used to limit the level of contradiction when μ^2 is close to zero. W is a weight function which takes into account n of R to calculate C.

$$W = \left(1 + \exp(\frac{1-n}{\beta})\right)^{-1} \tag{4.2}$$

where β is a scaling factor.

4.1.3 Contradictory versus Contrastive

A given review R consisting of two or more sentences with opposite polarity orientations, it is considered as having a *contradiction* if the sentences refer to the same topic or attribute, whereas if the divergence in polarities refer to different attributes of the overall topic, the review is considered to have a *contrast*. Table 4.1 shows examples of contradiction and contrast.

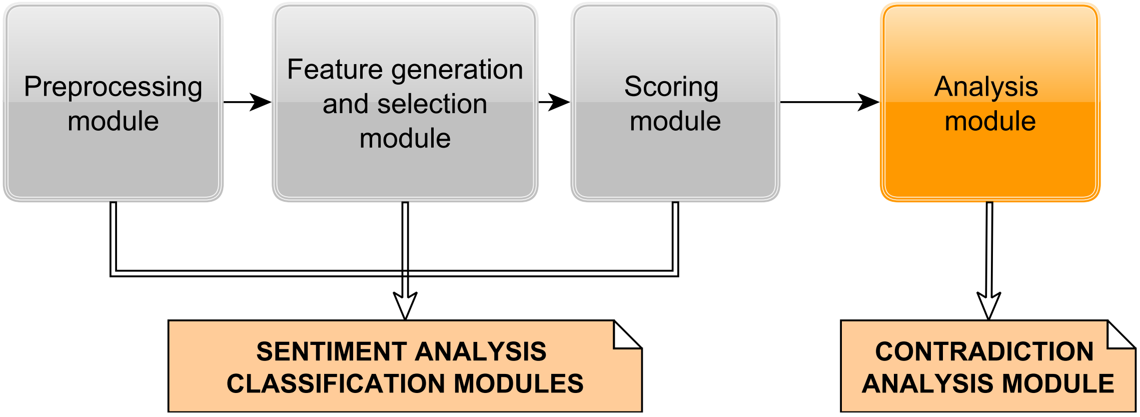
Table 4.1: Contradiction vs Contrast

Sentence 1	**Sentence 2**	**Type of review**
“update made it worse"(-)	*“but I still enjoy using the app"*(+)	**Contradictory**
“good site and content"(+)	*“bad app hard application to navigate"*(-)	**Contrastive**

Source: Vargas e Moreira (2015)

In this work, we are looking for intra-document synchronous contradictions in text from the sentiment analysis approach (sentiment-based contradictions). More specifically, we are looking for reviews that contain contrastive/contradictory sentences us-

Figure 4.1: Classification and contradiction analysis modules


Preprocessing module
Feature generation and selection module
Scoring module
Analysis module
SENTIMENT ANALYSIS CLASSIFICATION MODULES
CONTRADICTION ANALYSIS MODULE

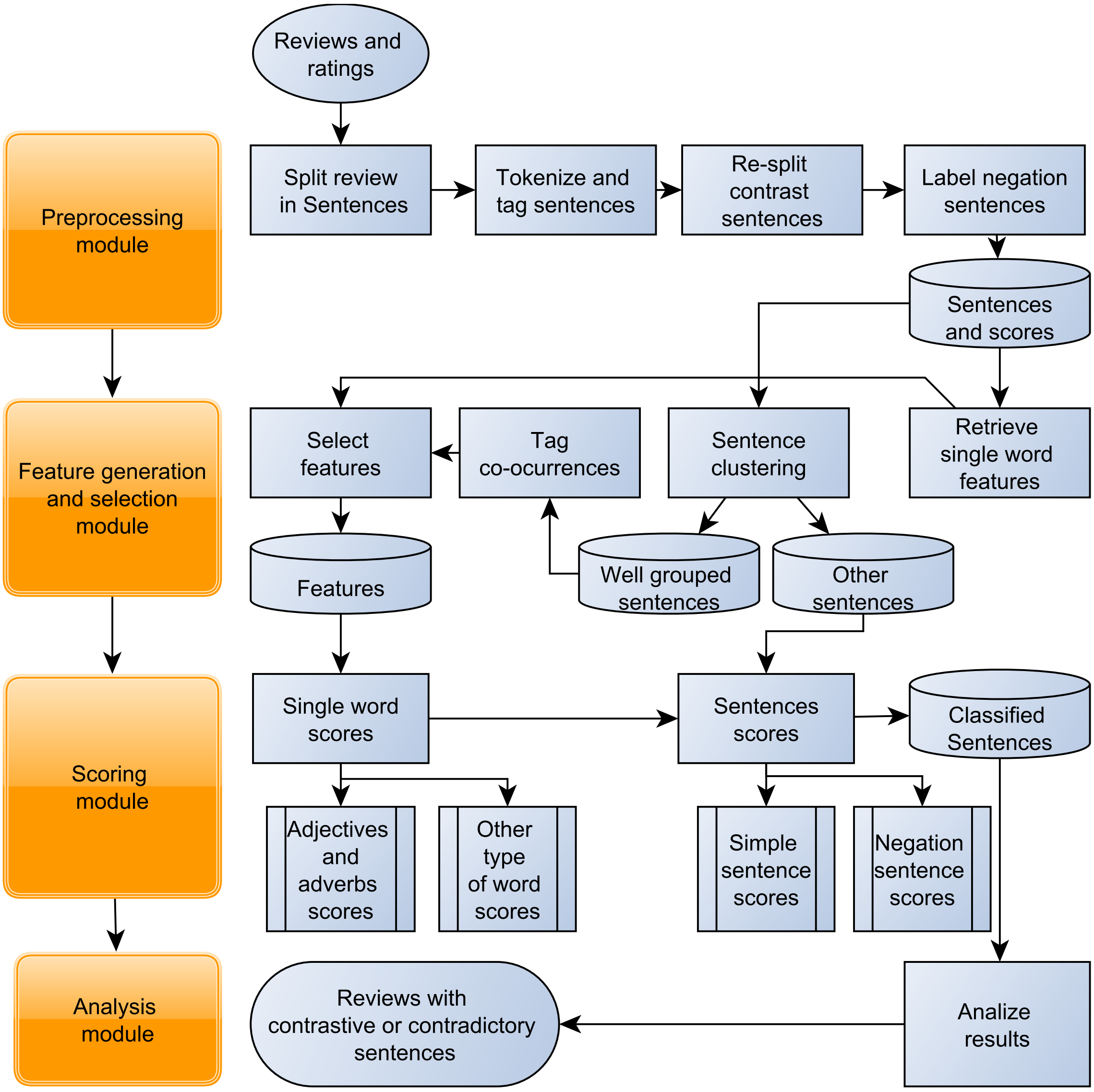
ing the polarity orientation of the sentences to decide whether a review contains contrastive/contradictory sentences.

Among the different definitions of contradiction, this work adopts the one from a sentiment analysis perspective given in 4.1.1 as it is the only one that fits into the kind of contradictions that we seek in the present work. Based on this definition, we propose a framework to detect sentiment-based contradictions. The proposed framework takes a review as input. The review is then processed by four modules: Preprocessing, Generation and Selection of Features, Scoring, and Analysis. The first three modules perform the classification task, and the fourth module performs contradiction analysis. The classification task is an exploratory study that aims at identifying and using different resources (Wordnet, Stanford NLP Toolkit , Weka) in each of the three proposed modules. On the other hand, in the contradiction analysis task, we aim to adapt and improve the results of an existing sentiment-based contradiction detection framework. These two tasks are represented in Figure 4.1. The output of our framework is a list of reviews that contain contrastive or contradictory sentences. Figure 4.2 shows the architectural overview of our framework. The four modules of our proposed framework are described next.

4.2 Preprocessing

This module receives a set of reviews R as input. Each review $r \in R$ may consist of one or more sentences. Then, each review r is split into k sentences which are submitted to a part-of-speech (POS) tagger to assign a grammatical class (noun, verb, adjective, etc.) to each word. We are interested in words tagged as adjectives (JJ), verb (VB), modal verb (MD), adverbs (RB), nouns (NN), adjective comparative (JJR), preposition (IN),

Figure 4.2: Framework for identifying contrastive sentences



Source: Vargas e Moreira (2015)

conjunction (CC)[1], as these are the classes of most interest ofr polarity identification. The next step is to determine whether a given sentence S contains contrastive/negated words. So, we assign two attributes to each sentence, “Contrast" and “Negated". These are Boolean attributes which take the value “0/1" in the absence/presence of contrastive or negated words, respectively. In order to assign a Boolean value to each sentence, we restricted our analysis to the set of terms listed below.

Contrast Attribute.- This attribute takes as the default value “0" and it changes its value to “1" when it contains terms such as *no, not (or its contraction n't), never, less, without, barely, rarely, no longer, no more, no way, no where, by no means, at no time*.

[1] For a complete list of POS tags, please refer to <https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html>.

Negation Attribute.- This attribute takes as the default value “0" and it changes its value to “1" when it contains terms like *although, however, differ, unlike, yet, but, instead, whereas, on the contrary, on the other hand, contrary to*. After determining the value of the *contrast attribute* in each sentence, the sentences in which this value is 1 are divided into $(m-1)$ sequential sub-sentences, where m is the number of contrast terms that the sentence contains. The contrast attribute is important to manipulate sentences that can be separated into even smaller units, while the negation attribute permits us to prevent the effect of negation terms into a given sentence. Finally, we can group the sentences based on their contrast and negated attribute. The resulting groups are defined below.

Simple tagged Sentences.- Sentences for which both the value of contrast and negated attribute is “0" belong to this group.

Tagged Sequences of Contrast Sentences.- Sentences that were divided by their contrast terms into sequential sub-sentences pertain to this group.

Sentences with Negation Terms.- Sentences in which the value of their negated attribute is “1" pertain to this group.

A step by step example of how the preprocessing module works is shown below. The input review (a) is divided into four sentences (b), (c), (d), and (e). Each of these sentences is tagged. Sentence (f) is the result of tagging sentence (b), which is the only that presents a contrast term “but", and a negation term “not". So, it is split again by the contrast term and gives us as the result the sentences (g) and (h). At the end of this step we have three simple sentences (c), (d) and (e), two contrast sentences (g) and (h), and one sentence with a negation term (h).

(a) “Prettier but not as smart! I’ve signed in 4x in 15 minutes! Sick of this!! I just want to listen..."

(b) “Prettier but not as smart!"

(c) “I’ve signed in 4x in 15 minutes!"

(d) “Sick of this!!"

(e) “I just want to listen..."

(f) “prettier$_{(\mathrm{JJR})}$ but$_{(\mathrm{CC})}$ not$_{(\mathrm{RB})}$ as$_{(IN)}$ smart$_{(\mathrm{JJ})}$!"

(g) “prettier$_{(\mathrm{JJR})}$"

(h) “not_RB as$_{(\texttt{IN})}$ smart$_{(\texttt{JJ})}$!"

4.3 Feature Generation and Selection

This module consists of three sub-modules: Retrieve Single words Features, Sentences clustering, and Tag co-occurrences. The first and second sub-modules deal with all sentences from the preprocessing module, but the third sub-module takes only the sentences that were correctly grouped by the clustering step.

4.3.1 Retrieve Single Word Features

In the works that deal with sentiment analysis, the words that belong to the syntactic categories adjective, adverb, or noun are frequently taken into account. The adjectives are very important because they directly convey the sentiment expressed by the user about some entity (noun). The adverbs are generally used as *intensifiers* or *shifters*. The intensifiers increment or decrement the strength of a given adjective. On the other hand, the shifters can change the meaning of a given adjective. In this work, we also use these features to do the classification task. The output of this sub-module is a *list of words, which can be adjectives, adverbs, nouns* and their variants.

4.3.2 Sentence Clustering

We use a clustering algorithm to find other important types of words, which could be used as features when there are no adjectives, adverbs, or nouns for the classification task. Adjectives, adverbs, and nouns are the most widely used words as features to represent a given sentence or review. However, there are sentences which do not contain neither of them. For this type of sentences, we have no way of predicting their class, if only this type of words are considered for classification. So, we make use of a clustering algorithm to find other important types of words, which could be used as features. The Expectation-Maximization clustering algorithm (DEMPSTER; LAIRD; RUBIN, 1977) is used to group the data into five classes: Very positive (5-star), Positive (4-star) Neutral (3-star), Negative (2-star), and Very Negative (1-star). Since we have the information about the number of stars assigned by the reviewer, we can select the subset of sentences that was grouped correctly. Words from the correctly grouped sentences are selected as features. Each word is a feature of the class in which it occurs most. Finally, for each group, we select the most representative features. Some word examples are shown below. We decide to use clustering by the availability of their algorithms. More specifically, by the easy use of the algorithm EM through Weka.

Table 4.2: Examples of words from the output of the clustering sub-module.

Word	**Syntactic Category**	**Group**	**Occurrences**
recommend	verb(VB)	very-positive(5-stars)	50
forgive	verb(VB)	negative(2-stars)	80
must	modal-verb(MD)	very-negative(1-star)	40

Source: Vargas e Moreira (2015)

The output of this sub-module is a list of words, each of them with its tag (syntactic category), class in which it appears (group), and the number of occurrences.

4.3.3 Tag Co-occurrence

For the simple tagged sentences, we only need the representation by tags of the sentences as the goal is to find the tags that co-occur frequently (*i.e.,* that appear frequently together). We consider adjectives (JJ), adverbs (RB), nouns (NN), verbs (VB) and their variants as the most important unigram features. Co-occurrences of two and three tags are considered. They take into account their number of occurrences and how close they are to the unigram features. The most frequent co-occurrences are: $\langle \texttt{JJ}, \texttt{NN} \rangle$, $\langle \texttt{NN}, \texttt{JJ} \rangle$, $\langle \texttt{VB}, \texttt{NN} \rangle$, $\langle \texttt{RB}, \texttt{JJ} \rangle$, $\langle \texttt{DT}, \texttt{JJ}, \texttt{NN} \rangle$, $\langle \texttt{JJ}, \texttt{JJ}, \texttt{NN} \rangle$, $\langle \texttt{VB}, \texttt{RB}, \texttt{JJ} \rangle$. We decide to take into account the co-occurrence of tags in order to consider the context in which a given word frequently appears. The result of this sub-module is a list of *two and three co-occurrence tags*. Some examples of the output of this sub-module are presented in the table 4.3.

Table 4.3: Examples of the Tag co-occurrence sub-module output

Tag co-occurrence	**Number of tags**
veryy$_{(\texttt{RB})}$, good$_{(\texttt{JJ})}$	2
mostly$_{(\texttt{RB})}$, unuseable$_{(\texttt{JJ})}$	2
selection$_{(\texttt{NN})}$, is$_{(\texttt{VBZ})}$,nice$_{(\texttt{JJ})}$	3
can$_{(\texttt{MD})}$, not$_{(\texttt{RB})}$, see$_{(\texttt{VB})}$	3
really$_{(\texttt{RB})}$, annoying$_{(\texttt{JJ})}$, popup$_{(\texttt{NN})}$	3

Source: Vargas e Moreira (2015)

From the results of the three sub-modules, we select the most relevant features in order to achieve a good representation of the sentences. In the sentences that follow, sentence (a) is a sentence which will be processed by this module and sentence (b) is the representation of the sentence (a) through its most relevant features (words and sequences of words).

(a) ‘almost$_{\texttt{(RB)}}$ perfect$_{\texttt{(JJ)}}$ this$_{\texttt{(DT)}}$ app$_{\texttt{(NN)}}$ in$_{\texttt{(IN)}}$ general$_{\texttt{(JJ)}}$ is$_{\texttt{(VBZ)}}$ just$_{\texttt{(RB)}}$ as$_{\texttt{(IN)}}$ useful$_{\texttt{(JJ)}}$ as$_{\texttt{(IN)}}$ the$_{\texttt{(DET)}}$ website$_{\texttt{(NN)}}$"

(b) “almost$_{\texttt{(RB)}}$,perfect$_{\texttt{(JJ)}}$ general$_{\texttt{(JJ)}}$ useful$_{\texttt{(JJ)}}$"

Finally, as the output of this module, we have three different results: (i) *a list of single word features* which is composed of two groups of words; the first group contains adjectives, adverbs, and nouns, while the second group contains other types of words such as verbs, modal verbs; (ii) *The list of co-occurrences of two and three tags*; and (iii) *The sentences represented by their features*.

4.4 Scoring

The scoring module has two sub-modules. The first one takes the list of single-word features and the second uses the co-occurrences of tags and the features of the sentence.

4.4.1 Single Word Scores

The Wordnet was used to determine the polarity and the strength of the adjectives and to calculate the semantic relatedness of word senses. We decide to use this resource in order to exploit the context information that it provides through the semantic relationships between its words. The relationships between words can be antonym, entailment, holonym, meronym, and similar_to. We disregarded the antonym relatedness, as it increases the errors in the classification task. In order to assign a class to a given word w, we measure four relatedness values: relatedness between w and “best" (a), relatedness between w and “good" (b), relatedness between w and “bad"(c), and relatedness between w and “worst" (d). The words *best, good, bad*, and *worst* represent the classes Very Positive

(5-stars), Positive (4-stars), Negative (2-stars) and Very Negative (1-star), respectively. The class assigned to the word w is the class that obtains the highest value among (a), (b), (c), and (d). In the exceptional case when (a), (b), (c), and (d) all have zero as a relatedness value, the *neutral* class is assigned to the word w. Finally, by taking into account some rules, we can convert from adverb to adjective (*i.e.,* by removing the suffix *ly* we convert *badly* to *bad*). The classification of the adjectives in classes and synonym relatedness helps us assign a class to adverbs. When a given word w is not an adjective or an adverb, we assign w to class C_1 such that w occurs more frequently in C_1.

4.4.2 Sentences Scores

The features for the list of co-occurrences of two and three tags and the sentences represented by their features together with the scores for single words are used to assign a class to the sentences. The adjectives and words obtained with the clustering step help us assign a class to the sentences. For the simple sentences, the adverbs are used only as intensifiers. They can convert a given class to the next higher or lower class. For instance, from Negative to Very Negative, from Positive to Very Positive, or from Positive to Neutral. For the Sentences with negation words (shifters), an additional analysis is necessary to find out the feature that is negated by a shifter. The shifters can convert a given class to the completely opposite class. For instance, Negative to Positive or Very Negative to Very Positive. Next, we show an example of assigning a class to the given sentence which is represented by their features(single words and sequences of words).

(c) “almost$_{(\mathrm{RB})}$ → perfect$_{(\mathrm{JJ})}$ general$_{(\mathrm{JJ})}$ useful$_{(\mathrm{JJ})}$"

(d) “almost$_{(\mathrm{RB})}$ (3) → perfect$_{(\mathrm{JJ})}$(4) general$_{(\mathrm{JJ})}$(3) useful$_{(\mathrm{JJ})}$(4)"

(e) “almost$_{(\mathrm{RB})}$ → perfect$_{(\mathrm{JJ})}$(4) general$_{(\mathrm{JJ})}$(3) useful$_{(\mathrm{JJ})}$(4)"

(f) “almost$_{(\mathrm{RB})}$ → perfect$_{(\mathrm{JJ})}$ general$_{(\mathrm{JJ})}$ useful_JJ" → (4) Positive

4.5 Analysis

This step can be considered as a post-processing step or a contradiction analysis step. The results of the previous module are used here in order to determine whether there exists a contradiction or not in a given set of reviews (input data). The analysis is based on the polarity information of the sentences that compose each review. In this module, we describe the process of adapting and extending the original framework proposed by Tsytsarau, Palpanas e Denecke (2011) which was introduced in Section 3.

4.5.1 Adapting the Framework

Based on the original framework, we perform some modifications in order to adapt it to our context as follows.

Identification of Topics.- Since we are looking for intra-document contradictions and considering that the input reviews are about a single overall topic, the step in which topics are identified is not necessary for our context.

Detection of Sentiments.- The goal of this step is to assign sentiment values (*i.e.,* positive, negative and neutral) to each sentence-topic pair. Since we are dealing with a single topic, we only need to perform the assignment of sentiment values to each sentence, which can be achieved by the polarity classification. In order to perform the polarity classification, we used RNTN (SOCHER et al., 2013) which was introduced in 2.3.5.

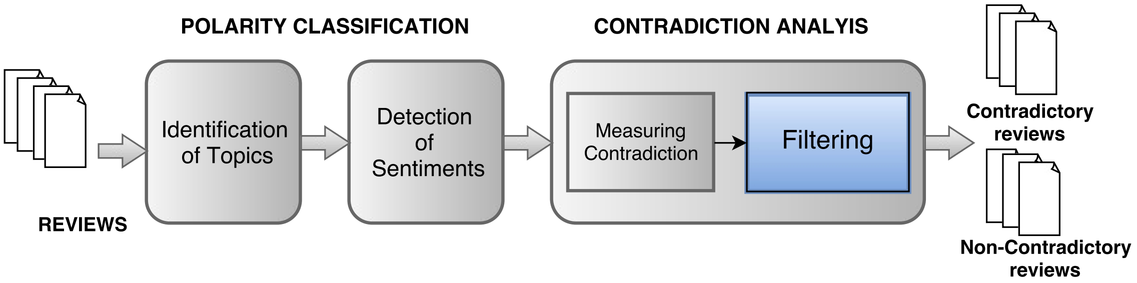
Measuring Contradictions.- In the original framework, this step aims to find the contradictory opinions across documents based on the contradiction measure C. We also perform this step by considering the adapted version of the measure as it was presented previously in 4.1.2. At this point, we have the sentences classified as positive or negative. Furthermore, based on these classified sentences, the contradiction value C is calculated for each review. So, we select the reviews with the highest C value, labeling them as contradictory.

4.5.2 Extending the Framework

In this step, we extend the original framework as we can see in the figure 4.3 by adding a filtering step (see Figure 4.4) that aims to remove the reviews erroneously labeled as contradictory. Furthermore, we use the C value assigned in the previous step to each review R of the input data in order to label R as a review that contains contradictory sentences or not based on two criteria.

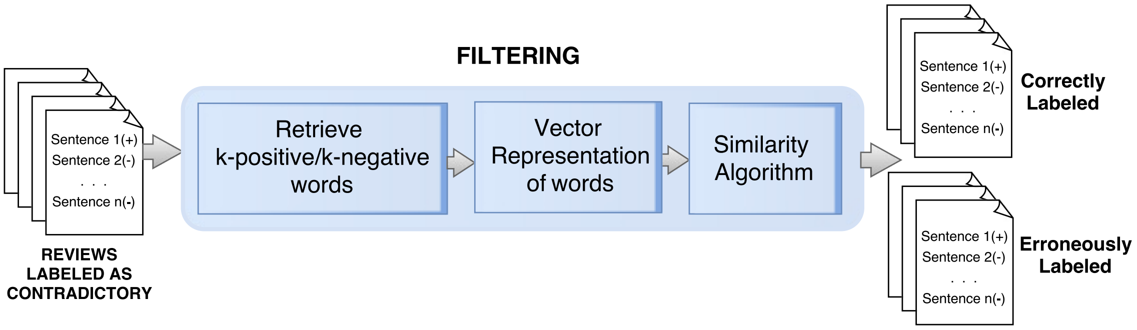
First criterion.- The reviews with the highest C value are labeled as the reviews that contain contradictory sentences.

Figure 4.3: Extended sentiment-based contradiction analysis framework



Source: Vargas e Moreira (2016)

Figure 4.4: Similarity-based filtering method



Source: Vargas e Moreira (2016)

Second criterion.- Instead of only relying on the C value of the reviews, it tries to filter the errors of the previous module (polarity classification of sentences) by using the proposed filtering step method. Method that is based on the similarity of words, more specifically, on the cosine between the vector representation of two groups of words (group of k-positive/negative words and words of a given input sentence). The way to retrieve the k-positive and k-negative words, the vector representation of words, the similarity algorithms as well as the process of filtering errors are detailed next.

Retrieve k-positive/k-negative Words.- In this step, we select the k-most representative positive and negative words. This selection can be manually, automatically or semi-automatically performed. The manual selection requires domain knowledge (LIU et al., 2004). The automatic selection can be performed, for example, by relying on results of clustering algorithms or on the results of regression models (SANGANI; ANANTHANARAYANAN, 2013), while the semi-automatic selection combines manual and automatic selection methods.

Vector Representation of Words.- This step is responsible for providing a vector representation of words. This vector representation plays an important role in the effectiveness of our proposed algorithms. At this point, we need to select the best resource that permits us to represent words through vectors for our similarity algorithms. So, we decide to use the state of the art for the word similarity task, Word2vec which was introduced in 2.3.3.

Similarity Algorithm.- From the vector representation of words, we decide to use the well-known cosine distance which measures the similarity of two vectors by relying on the cosine of their angle. Furthermore, we formally define this words similarity as follows. Given a word w and a set of words V={v_1,v_2,...,v_k}, the similarity function SW assigns a value $d \in$ [-1,1] to w based on the cosine distance algorithm ϕ of the vector representation of word w regard to the vector representation of each v_i with $i \in$ {1,2,...,k}

$$SW\left(w,V\right) = \phi\left(CosSimil\left(w,v_i\right)\right)$$

We propose two algorithms 1 and 2 to measure the similarity between two sets of words. These algorithms are used to calculate the similarity between the set of k-positive/negative words and the set of words of a given sentence S by obtaining two real values. Values that can be interpreted as the positive and negative orientation of the given sentence S. The difference between the two proposed algorithms is the way that they average the similarity values. In our experiments, we used the maximal and mean values but it is possible to test other alternatives such as the minimal values.

Algorithm 1: Measuring the mean-similarity between two sets of words

```
input  : A set of $n$-words $A$ and a set of $m$-words $B$
output: A real value *resp* that represents the similarity between $A$ and $B$
$first\_array \leftarrow []$;
for $i \leftarrow 0$ to $n-1$ do
    $second\_array \leftarrow []$;
    for $j \leftarrow 0$ to $m-1$ do
        $simil \leftarrow cos\_distance(Word2Vec(A[i]), Word2Vec(B[j]))$ ;
        if $(simil >= -1)$ then
            $second\_array.add(simil)$;
        end
    end
    $second\_array\_without\_outliers \leftarrow remove\_outliers(second\_array)$;
    $mean\_value \leftarrow mean(second\_array\_without\_outliers)$;
    $first\_array.add(mean\_value)$;
end
$resp \leftarrow mean(first\_array)$;
```

Algorithm 2: Measuring the max-similarity between two sets of words

```
input  : A set of $n$-words $A$ and a set of $m$-words $B$
output: A real value *resp* that represents the similarity between $A$ and $B$
$first\_array \leftarrow []$;
for $i \leftarrow 0$ to $n-1$ do
    $second\_array \leftarrow []$;
    $max\_simil \leftarrow -2$;
    for $j \leftarrow 0$ to $m-1$ do
        $simil \leftarrow cos\_distance(A[i], B[j])$ ;
        if $(simil > max\_simil)$ then
            $max\_simil \leftarrow simil$;
        end
    end
    if $(max\_simil >= -1)$ then
        $first\_array.add(max\_simil)$;
    end
end
$resp \leftarrow mean(first\_array)$;
```

For each our previous similarity algorithms, we can combine it with an existing classifier on the literature in order to implement our polarity orientation algorithm. In our work, we decide to use the state of the art for the polarity classification at sentence level for movie reviews (this classifier can be replaced by others), the RNTN classifier which was introduced in 2.3.5.

So, we combine the RNTN results with the results of each of our similarity algorithms to determine the polarity orientation of sentences. Furthermore, we formally define the algorithm below.

For a given sentence S, the two real values obtained with the previous algorithms, the sentiment orientation $sent_{RNTN}$ assigned by the RNTN classifier, and a threshold value t, we can determine the polarity orientation of S by using Algorithm 3.

Algorithm 3: Determining the polarity orientation of a given sentence

```
input : A sentence $S$, the real value $SP$ w.r.t. the $k$-positive words, the real
        value $SN$ w.r.t. the $k$-negative words, the sentiment orientation
        $sent_{RNTN}$ assigned by the RNTN classifier, and a threshold $t$
output: the polarity orientation $sent$ of $S$ based on $SP$, $SN$, and $t$
diff $\leftarrow SP - SN$;
if $||diff|| > t$ then
    if diff $> 0$ then
        $sent \leftarrow positive$;
    else
        $sent \leftarrow negative$;
    end
else
    sent $\leftarrow sent_{RNTN}$;
end
```

Where the threshold value t allows us to opt for using/not using the neutral class in the polarity orientation algorithm. Where t=0 means that the neutral class is not considered.

Filtering Errors.- This step performs the filtering process of reviews that were erroneously labeled as contradictory. The input of this step consists on a given review R that was labeled as contradictory in the Measuring Contradiction step, two real values for each sentence of R which represent the positive and negative orientation, the sentiment orientation $sent_{RNTN}$ assigned by the RNTN classifier, and the threshold t which is used in the sentiment orientation algorithm 3. So, the polarity value obtained from our sentiment orientation algorithm are used in order to determine the real orientation of each sentence. Finally, It determines if the review R is really contradictory or not(it should be filtered) by relying on the number of positive and negative sentences in R.

Algorithm 4: Determining if a given review should be filtered

```
input : Array of $n$ sentences that represent the current review, array *SP* with
        $n$ similarity values regard to the $k$-positive words, array *SN* with $n$
        similarity values regard to the $k$-negative words, array *RNTN* with $n$
        sentiment orientation values,and a threshold $t$
output: review $R$ labeled as contradictory or not
for $i \leftarrow 0$ to $n - 1$ do
    $sentence \leftarrow sentences[i]$;
    $sentim \leftarrow$
      $sentiment\_orientation(sentence\_a, SP[i], SN[i], RNTN[i], t)$;
    if $sentim == positive$ then
        $array\_positives.add(sentence)$;
    else
        $array\_negatives.add(sentence)$;
    end
end
*diff* = $length(array\_positives) - length(array\_negatives)$;
if $||$*diff*$||$ == *0* **or** $||$*diff*$||$ == *1* then
    *The review is contradictory*
else
    *The review is not contradictory (it should be filtered out)*
end
```

4.5.3 Summary

In this chapter, we described the proposed framework to detect sentiment-based contradictions in reviews, which is the main contribution of this dissertation. This framework can be divided into two important phases: Classification and Contradiction Analysis. For the first phase, we described three modules that perform the classification task, while for the second phase, we describe an existing contradiction analysis module based on the contradiction measure C which was adapted to our context. Furthermore, the contradiction module was extended by our filtering process that aims to remove the erroneously labeled reviews as contradictory.

5 EXPERIMENTAL EVALUATION

This chapter describes the experiments that were carried out in order to test our proposed framework. The experiments are organized in two phases. The first phase deals with our three-module classifier, while the second phase is evaluates polarity classification and contradiction detection. The dataset, the evaluation metrics, the experimental procedure, as well as the obtained results are described next. It is important to make clear that we do not use any supervised machine learning algorithms, so the steps of training and testing are not performed in the experiments.

5.1 Dataset

Our dataset is composed of users' reviews about Android applications. The reviews were collected from the Google Play Store (SANGANI; ANANTHANARAYANAN, 2013). The data is divided into seven groups according to the application they refer to. Each group contains 4500 reviews in English about a different Android application. Each review contains information on reviewer ID, creation time, rating (from 1 to 5), and review text. For the experiments, we only used the review text and its rating. It is important to recall that a review may consist of one or more sentences. So, the number of sentences for each review group is not the same. The distribution of sentences per class is unbalanced and is detailed in Table 5.1. The positive reviews (4 and 5 stars are the most frequent).

Table 5.1: Distribution of Sentences

Class	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	All groups
5-stars	3734	4894	3843	4444	3836	4574	4904	29903
4-stars	1813	1289	1164	1196	1030	1247	2091	9129
3-stars	1499	462	741	343	1035	665	790	4704
2-stars	1239	264	469	98	721	542	421	2933
1-star	3098	794	1021	196	1446	1537	883	8161
Total reviews	**11383**	**7703**	**7238**	**6277**	**8068**	**8565**	**9089**	**54830**

5.2 Evaluation Metrics

The well-known evaluation metrics of accuracy, precision, recall, and F-measure are used to measure the performance of our classification and the contradiction detection solutions. They are calculated according to the following equations.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{5.1}$$

$$Precision = \frac{TP}{TP + FP}, \tag{5.2}$$

$$Recall = \frac{TP}{TP + FN}, \tag{5.3}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5.4}$$

where TP, TN, FP, and FN stand for True positive, True negative, False positive, and False negatives, respectively.

5.3 Contradiction Analysis on the three-module Classifier

In this section, we implement a classification system based on the three first modules (three-module classifier), classify the input reviews at the sentence level, and perform the contradiction analysis over the classification results.

5.3.1 Three-module Classifier

The classifier receives reviews as input data, splits them into sentences, and returns the sentences labeled with one of five possible classes. The classes are Very-positive (5-stars), Positive (4-stars), Neutral (3-stars), Negative (2-stars) and Very-negative (1-star). Finally, the class predicted by the classifier is compared against the star rating assigned by the user in order to allow for the calculation of the evaluation metrics.

One of the problems observed in the data was the presence of unclassifiable sentences which cannot be represented by the attributes considered in our framework. For example, some sentences were composed solely by emoticons, or by expressions such as “yup" (which we do not handle at the moment as they are not in the dictionary). Nevertheless, in most cases the review was not composed exclusively of unclassifiable sentences. Table 5.2 details the quantity of classifiable and unclassifiable sentences in our data.

Table 5.2: Distribution of classifiable and unclassifiable sentences

Type	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	All groups
Unclassifiable	3848(34%)	3240(42%)	3788(52%)	2915(46%)	3158(39%)	3456(40%)	3430(38%)	22743(41%)
Classifiable	7535(66%)	4463(58%)	3450(48%)	3362(54%)	4910(61%)	5109(60%)	5659(62%)	32097(59%)
Total sentences	11383	7703	7238	6277	8068	8565	9089	54830

Table 5.3: Classification Results

Data		Negative-Class	Neutral-class	Positive-class	Average
Group 1	Precision	0.38	0.33	0.44	0.38
	Recall	0.50	0.13	0.56	0.40
	F-measure	0.43	0.19	0.49	0.39
Group 2	Precision	0.30	0.39	0.49	0.40
	Recall	0.20	0.08	0.85	0.38
	F-measure	0.24	0.13	0.62	0.39
Group 3	Precision	0.34	0.34	0.47	0.39
	Recall	0.35	0.11	0.72	0.39
	F-measure	0.34	0.17	0.57	0.39
Group 4	Precision	0.35	0.24	0.62	0.40
	Recall	0.11	0.05	0.93	0.36
	F-measure	0.17	0.08	0.74	0.38
Group 5	Precision	0.33	0.32	0.48	0.38
	Recall	0.36	0.15	0.65	0.39
	F-measure	0.34	0.20	0.55	0.38
Group 6	Precision	0.33	0.36	0.49	0.39
	Recall	0.34	0.10	0.74	0.39
	F-measure	0.33	0.16	0.59	0.39
Group 7	Precision	0.32	0.38	0.43	0.37
	Recall	0.19	0.11	0.79	0.36
	F-measure	0.24	0.17	0.56	0.36
All groups	Precision	0.34	0.34	0.49	0.39
	Recall	0.31	0.10	0.76	0.39
	F-measure	0.32	0.15	0.60	0.39

5.3.2 Classification Results

In order to evaluate our classification results as three-class classification problem {Negative, Neutral, Positive}, sentences which have one or two stars are labeled as Negative. Sentences which have three stars are labeled as Neutral and the sentences which have four or five stars rating are labeled as Positive. We evaluated our classification system for each group of reviews and for the union of groups. The results for the different groups of reviews were very similar. In Table 5.3, the results obtained in the classification task for the different groups of reviews and for the union of groups are detailed. Positive sentences were classified more accurately than negative sentences, and the neutral sentences were the hardest to classify. Overall, the recall was noticeably superior for the positive class.

5.3.3 Contradiction Analysis

The analysis was performed on the output of the three-module classifier. This output consists of sentences with the rating assigned by the user and the rating assigned by the classifier. Then, we label all sentences as follows. The sentences are labeled as contrastive/contradictory when the class assigned by the three-module classifier is very-positive (5-stars)/very-negative (1-star) and the class assigned by the reviewer (user) is very-negative (1-star)/very-positive (5-stars). The sentences are labeled as non-contrastive/non-contradictory when the class assigned by the three-module classifier and the class assigned by the reviewer are the same or the difference between them is one class, for example the case in which a given sentence was classified as very-positive (5-stars) by the three-module classifier and as positive (4-stars) by the reviewer. Finally, the sentences are labeled as undefined in all other cases. The distribution of contrastive/contradictory sentences across groups was similar. Table 5.4 details the distribution of contrastive and non-contrastive/non-contradictory sentences retrieved by our system. We can observe that the contrastive sentences represent approximately 20% of the total number of sentences.

Table 5.4: Contrastive/Contradictory and Non-contrastive/Non-contradictory sentences

Type	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	All groups
Non-Contrastive	4151	2472	1877	2263	2690	2787	3070	17749
Contrastive	1309(17%)	883(20%)	556(16%)	354(11%)	869(18%)	967(19%)	1054(19%)	5708(18%)
Undefined	2075	1108	1029	747	1352	1361	1535	8645
Total sentences	7535	4463	3462	3364	4911	5115	5659	32102

5.3.4 Contradiction Analysis Results

From the analysis of the sentences that was labeled as the contrastive/contradictory sentences by our three-module classifier, we can observe two possible cases. The first case appears when the disagreement of the two rating values is the result of a misclassification in our system. The second case happens when a sentence presents a contradiction between its two rating values or presents contrastive sentences. In order to distinguish between contrast and contradiction in the current experiment, the other sentences of the same review should be considered. A contrast happens when the sentence in question has a polarity that diverges from the polarity of the other sentences in the same review. A contradiction happens when the sentences have all the same polarity but it diverges from the star-rating.

Finally, we selected a random sample of 360 sentences from our system's output in order to do a manual analysis and find out whether there is really a contradiction or misclassification. The results of this experiment are detailed in Table 5.5.

Table 5.5: Occurrences of contrastive or contradictory sentences over a random sample of 360 sentences

	Reviewer	**Annotator**	**# Sentences(%)**
Case 1	Negative/Positive	Neutral	40 (11,11%)
Case 2	Negative/Positive	Positive/Negative	32 (8,89%)
Total -	-	-	72 (**20,00%**)

5.4 Polarity Classification and Contradiction Detection

In this Section, we carry out two experiments: Polarity Classification, and Contradiction Detection. The first one is based on Algorithm 3, and the second experiment is based on Algorithm 4.

Pre-processing and RNTN Classification.- A pre-processing step was performed in order to remove incomplete reviews such as those that did not contain star ratings. This step reduced the number of reviews from 31500 to 31482. Then, polarity classification was performed using RNTN. This classification takes the text of the reviews as input, splits them into sentences and assigns one of five possible values to each sentence (1,2,3,4,5). This values can be organized in three groups ((1,2), (3), (4,5)) that represent the negative, neutral and positive orientation, respectively. In this step, the number of reviews is reduced from 31482 to 30228. The main reason for this reduction is the presence of non-English or single-emoticons sentences which cannot be classified.

Measuring Contradictions.- The first step in this analysis consists in measuring the contradictions. Thus, for each review , we calculate its contradiction value C with the small value ϑ fixed in $0,0005$. After that, we perform the selection of the reviews with the highest C value. C ranges from 0.00 to $2.98e-06$. It takes the minimum value when all sentences of a given review have the same polarity value, and assumes the maximum value when the sentences of a given review have the same number of positive and negative sentences.

Data Annotation.- For our two experiments, we selected the reviews which have the maximum C value (2.98e-06), which resulted in 840 reviews, all with two sentences each. Furthermore, we manually annotated the sentences in order to allow for subsequent analysis. The first annotation consists in labeling each of the 1680 selected sentences as positive, negative, or neutral. The second annotation consists in labeling each of the 840 reviews as contradictory or not by relying on the polarity of the first annotation. If a given sentence S_1 in review R has the polarity orientation assigned by RNTN different from their manually assigned polarity orientation, the review is considered as erroneously labeled as contradictory.

Selection of k-positive and Negative Words.- In this step, word selection was done by manually picking the most significant words from a list of 30 words assembled by (SANGANI; ANANTHANARAYANAN, 2013). In our experiments, the value of k was 19. *Negative-Words(N) = {“update", “open", “sucks", “phone", “uninstall", “ads", “play", “bad", “poor", “crap", “crashes", “useless", “uninstalled", “force", “terrible", “horrible", “uninstalling", “waste", “annoying" }*

Positive-Words(P) = {“love", “great", “good", “awesome", “best", “excellent", “nice", “game", “cool", “fast", “easy", “fun", “amazing", “addictive", “perfect", “super", “helpful", “fantastic", “better"}.

Filtering Method.- In order to determine which part of the reviews should be filtered, we performed the steps detailed below. We calculated the similarity of the 1680 selected sentences regarding each group of selected words by using the vector representations provided by Word2Vec in our similarity algorithms 1, and 2 (max_similarity, and mean_similarity). Furthermore, we calculated the polarity orientation of each sentence based on our algorithm 3 with the parameter value t fixed in 0.05 and the k-positive/negative selected words. Finally, we use Algorithm 4 to determine which reviews should be filtered.

5.4.1 Polarity Classification Experiment

For the 1680 sentences previously selected and using the polarity_orientation Algorithm 4, the polarity classification is conducted with the parameter value t fixed at 0. We performed two experiments, one for each of our two similarity algorithms(max_similarity, and mean_similarity) which take as the parameter each sentence of the 1860 selected sentences and the k-positive/negative selected words (Word2Vec is used in our algorithms in order to retrieve the vector representation of words).

5.4.2 Contradiction Detection Experiment

This experiment consists in two parts. The first part is the contradiction detection by using the adapted framework without our filtering method (relying only on the C value) and the second part is the contradiction detection by considering our filtering method.

5.4.3 Evaluation

The evaluation is performed for the two groups of experiments: Polarity Classification and Contradiction Detection. For Polarity Classification, we calculated precision, recall, accuracy, and F1 values for the RNTN classifier. For the Contradiction Detection, we calculated the accuracy of the adapted framework without our filtering method and the accuracy of the framework with our filtering method. We were not able to calculate the recall for the filtering method as we do not know the total number of reviews with contradictions.

5.4.4 Results

Results for the polarity_orientation algorithm using the mean_similarity algorithm and the max_similarity algorithm are shown in Table 5.6, and the results for Contradiction Detection are shown in Table5.7. The best results are shown in bold.

Table 5.6: Results on the classification task.

	Precision	**Recall**	**Accuracy**	**F1**
RNTN(Baseline)	**0.87**	0.60	0.63	0.71
Mean_Similarity (Alg. 1)	0.81 (-7%)	0.94 (+57%)	0.78 (+24%)	0.87 (+23%)
Max_Similarity (Alg. 2)	0.83 (-5%)	**0.95** (+58%)	**0.82** (+30%)	**0.88** (+24%)

Polarity Classification.- The polarity-orientation algorithm (Algorithm 3) using the two similarity measures (Algorithms 1 and 2) was compared with the RNTN classifier. The results showed that with both the max and the mean, there are gains in recall, accuracy, and F1. The proportional improvement is shown between brackets in Table 5.6. These gains were much larger than the loss in precision that the methods brought. This was a consequence of a large reduction in the number of false negatives but with a (smaller) increase in the number of false positives. Comparing the two proposed similarity measures, we observed a slight difference in favor of Algorithm 2 (max_similarity). A Wilcoxon signed-rank test on the accuracy of each method has shown that both improvements are statistically significant, yielding p-values < 0.0001. The same test applied to our two proposed versions show that Alg 2. is significantly superior to Alg 1. (p-value = 0.0016). We attribute the gains to the effective vector representation of the words achieved by Word2Vec, which is based on a very large corpus ($\approx$50Gb).

Contradiction Detection.- For detecting contradictions, we employed our filtering Algorithm (Alg. 4) with the two variations of the polarity-orientation algorithm. Both variations achieved an improvement in precision, however, the biggest advantage was yielded by max_similarity (Alg. 1). We believe that the mean_algorithm suffered from cases in which words that were not significant reduced the mean value impacting negatively on the classification results. The improvements in this task are directly dependent on the results achieved in the classification task.

Table 5.7: Results on the Contradiction Detection task.

	Precision
Without_Filtering(Baseline)	0.19
Filtering_mean (Alg. 1)	0.21 (+10%)
Filtering_max (Alg. 2)	**0.24 (+26%)**

5.4.5 Error Analysis

The polarity-orientation algorithm (Alg. 3) using Alg. 1 does not work well with sentences that start with an overall (positive/negative) evaluation followed by some (negative/positive) evaluations such as *“great app but it’s lacking the feature to play audio while taking notes in bookmark"*. In this type of sentences, the overall sentiment (which should be taken as polarity orientation) is lost when it is averaged with the other additional evaluations. Furthermore, the polarity-orientation algorithm (Alg. 3) using the two similarity measures (Algs. 1 and 2) may fail for words that can take positive or negative orientation depending on the context. For example the word *“simple*" takes a negative orientation in sentences like *“The app is sometimes slow and too simple"*, while, takes the positive orientation in sentences like *“The subscription to the app is free and simple"*. On the other hand, the filtering (Alg. 4) depends on the results of Alg. 1 and 2, so the improvements on the classification represent also an improvement on the contradiction detection. Finally, we did not perform any computational performance analysis (time or memory) as this was not the focus of the present dissertation.

5.5 Discussion

Here we discuss the results obtained in the experiments.

5.5.1 Contradiction Analysis on the Three-module Classifier

As we can see in Table 5.3, which shows us the results for the *classification* task performed with the three-module Classifier, the results for all groups of reviews present similar average values. The average *precision* ranges from 0.37 to 0.40, the average *recall* ranges from 0.36 to 0.40, and the average *F-measure* ranges from 0.36 to 0.39. We can see also that positive sentences were classified more accurately than negative sentences and recall was noticeably superior for the positive class. This happened because the positive class had more instances and thus dominated the classification model. The neutral sentences were the hardest to classify. This happened because there were no features to represent the neutral class – a sentence was classified as neutral when it did not contain evidences of being positive or negative. Even though the results of our classification system do not show improvements compared to existing published results over the metrics used, the implementation of it permits us to know how it works and how the classification systems can help us to address the contradiction analysis problem.

From the classification results, we performed the contradiction analysis in order to find contrastive or contradictory sentences. Table 5.4 shows the results of this analysis. The minimum percentage of contrastive sentences is 11% and the maximum is 20%. However, the sentences that were labeled as contrastive up to this point may not really be contrastive. So, a second experiment was performed consisting of the selection of a random sample of 360 sentences from the sentences that were labeled as contrastive. A manual analysis was employed to find out whether there is really a contrastive/contradictory or if it was a case of misclassification. The results in Table 5.5 show that 20% of sentences were really contrastive or contradictory. This analysis shows us that it is possible to detect contrastive or contradictory sentences directly from the results of a classifier. Furthermore, this analysis represents a basic way to address the contradiction analysis.

5.5.2 Polarity Classification and Contradiction Detection

As we can see in Table 5.6, which shows the results for our *polarity classification* experiment, our polarity-orientation algorithm (Algorithm 3) using the two similarity measures (Algorithms 1 and 2) improves the results of the RNTN classifier. The improvement arises mainly to the fact that our algorithm uses the results of the RNTN classifier in the cases in which it can not determine the polarity orientation of a given sentence. More specifically, when our similarity algorithms cannot determine the polarity orientation of a given sentence, we use the RNTN classifier to determine it. Even though this experiment is not our main goal, it proves the effectiveness of our similarity and polarity orientation algorithms.

Based on the polarity classification results, we performed the *contradiction detection* experiment. In this experiment, it is compared the results obtained from the original framework (adapted framework without the filtering process) with the results obtained from our proposed framework (adapted and extended framework with the filtering process). This comparison attested the importance of our filtering method as it improves the precision in the contradiction detection task shown in Table 5.7.

6 CONCLUSION

In this work, we proposed a framework to detect reviews that contain contrastive or contradictory sentences. The framework is based on the definition of contradiction from the sentiment analysis perspective. The framework is divided in two tasks. The first task is the classification of reviews at the sentence level and includes the modules of preprocessing, feature generation selection, and scoring. The analysis module represents the second task of this work, the contradiction analysis. In accordance to that, the experiments of this work were organized in two groups. The first group of experiments evaluated the implementation of a sentiment classifier. Even though the results of our sentiment classifier do not show improvements over the metrics used regarding to the existing classifiers on the literature, its implementation allowed us to understand how the classification systems can aid addressing the contradiction analysis problem. The second group of experiments assessed the implementation of a contradiction analysis system. This system was implemented based on the contradiction measure proposed by Tsytsarau, Palpanas e Denecke (2011). We adapted this measure to our context and improved its results by the addition of a process of misclassification filtering. The filtering process that is based on the similarity of words. Our results have shown that filtering increases the precision in the contradiction analysis task in all of the considered cases. In the best case, precision increases from 19.0% to 24.0% which represents a proportional improvement of 26%.

From our experiments and their results, we can affirm that the contradiction analysis system is a hard problem that needs specific solutions for each type of contradiction. Furthermore, for the contradictions that were addressed in the present work, the polarity orientation of sentences as well as the similarity between words are important features that allow for the detection of contrastive or contradictory sentences. Since the proposed algorithms are based on the similarity of isolated words without considering the proximity with other words, we did not cover some cases such as the existence of negation terms nor phrasal words.

As future work, we plan to design an automatic method for choosing the k most representative words. This could be implemented using logistic regression or clustering. We could also explore other ways to compare sets of words. For example, instead of comparing the words of a sentence with two independent sets (k-positive, k-negative), we could make sure beforehand that there is an antonymy relationship between the elements of the two sets. We also plan to test our framework with other datasets in order to generalize its scope.

As part of this dissertation, two papers were written. The first (VARGAS; MOREIRA, 2015) was published as a short paper in SBBD 2015, and the second (VARGAS; MOREIRA, 2016) is a full paper, which is currently under review at SBBD 2016.

REFERENCES

ACAMPORA, G.; COSMA, G. A hybrid computational intelligence approach for efficiently evaluating customer sentiments in e-commerce reviews. In: INTELLIGENT AGENTS (IA), 2014 IEEE SYMPOSIUM ON - [s.n], Piscataway, NJ, USA, 2014. **Proceedings...** Orlando, FL, USA: IEEE, 2014. p. 73–80.

CHEN, C. et al. Visual analysis of conflicting opinions. In: VISUAL ANALYTICS SCIENCE AND TECHNOLOGY (VAST), IEEE SYMPOSIUM ON - [s.n], Baltimore, MD, USA, 2006. **Proceedings...** Baltimore, MD, USA: IEEE, 2006. p. 59–66.

CHURCH, K. W.; HANKS, P. Word association norms, mutual information, and lexicography. **Computational linguistics**, MIT Press, v. 16, n. 1, p. 22–29, 1990.

DAGAN, I.; GLICKMAN, O.; MAGNINI, B. The pascal recognising textual entailment challenge. In: THE FIRST INTERNATIONAL CONFERENCE ON MACHINE LEARNING CHALLENGES: EVALUATING PREDICTIVE UNCERTAINTY VISUAL OBJECT CLASSIFICATION, AND RECOGNIZING TEXTUAL ENTAILMENT - 1st, Southampton, UK, 2006. **Proceedings...** Berlin, Heidelberg: Springer-Verlag, 2006. p. 177–190.

DAS, S.; CHEN, M. Yahoo! for amazon: Extracting market sentiment from stock message boards. In: THE ASIA PACIFIC FINANCE ASSOCIATION ANNUAL CONFERENCE (APFA) -[s.n], Bangkok, Thailand, 2001. **Proceedings...** Bangkok, Thailand: [s.n.], 2001. p. 43.

DAVE, K.; LAWRENCE, S.; PENNOCK, D. M. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: THE 12TH INTERNATIONAL CONFERENCE ON WORLD WIDE WEB - [s.n], Budapest, HUNGARY, 2003. **Proceedings...** New York, NY, USA: ACM, 2003. p. 519–528.

DEMPSTER, A.; LAIRD, N.; RUBIN, D. Maximum likelihood from incomplete data via the em algorithm. **Journal of The Royal Statistical Society, series B**, n. 7, p. 1–38, 1977.

ENNALS, R. et al. What is disputed on the web?. In: THE 4TH WORKSHOP ON INFORMATION CREDIBILITY - 4th, Raleigh, NC, USA, 2010. **Proceedings...** New York, NY, USA: ACM, 2010. p. 67–74.

ENNALS, R.; TRUSHKOWSKY, B.; AGOSTA, J. M. Highlighting disputed claims on the web . In: THE 19TH INTERNATIONAL CONFERENCE ON WORLD WIDE WEB - 19th, Raleigh, NC, USA, 2010. **Proceedings...** New York, NY, USA: ACM, 2010. p. 341–350.

ESULI, A.; SEBASTIANI, F. Sentiwordnet: A publicly available lexical resource for opinion mining. In: THE 5TH CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION - 5th, Genoa, Italy, 2006. **Proceedings...** Genoa, Italy: [s.n.], 2006. p. 417–422.

FAHRNI, A.; KLENNER, M. Old wine or warm beer: Target-specific sentiment analysis of adjectives. In: THE SYMPOSIUM ON AFFECTIVE LANGUAGE IN HUMAN

AND MACHINE - [s.n], Aberdeen, Scotland, 2008. **Proceedings...** Aberdeen, Scotland: AISB, 2008. p. 60–63.

FRASER, B. What are discourse markers? **Journal of pragmatics**, Elsevier, v. 31, n. 7, p. 931–952, 1999.

GALLEY, M. et al. Identifying agreement and disagreement in conversational speech: Use of bayesian networks to model pragmatic dependencies. In: THE 42ND ANNUAL MEETING ON ASSOCIATION FOR COMPUTATIONAL LINGUISTICS - 42nd, Barcelona, Spain, 2004. **Proceedings...** Stroudsburg, PA, USA: ACL, 2004. p. 669.

HALL, M. et al. The weka data mining software: an update. **ACM SIGKDD explorations newsletter**, ACM, v. 11, n. 1, p. 10–18, 2009.

HARABAGIU, S.; HICKL, A.; LACATUSU, F. Negation, contrast and contradiction in text processing. In: THE 21ST NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE - 21st, Boston, Massachusetts, 2006. **Proceedings...** Boston, Massachusetts: AAAI Press, 2006. p. 755–762.

HILLARD, D.; OSTENDORF, M.; SHRIBERG, E. Detection of agreement vs. disagreement in meetings: Training with unlabeled data. In: THE 2003 CONFERENCE OF THE NORTH AMERICAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS ON HUMAN LANGUAGE TECHNOLOGY - [s.n], Edmonton, Canada, 2003. **Proceedings...** Stroudsburg, PA, USA: ACL, 2003. p. 34–36.

HU, M.; LIU, B. Mining and summarizing customer reviews. In: THE 10TH ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING - 10th, Seattle, WA, USA, 2004. **Proceedings...** New York, NY, USA: ACM, 2004. p. 168–177.

JIA, L.; YU, C.; MENG, W. The effect of negation on sentiment analysis and retrieval effectiveness. In: THE 18TH ACM CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT - 18th, Hong Kong, China, 2009. **Proceedings...** New York, NY, USA: ACM, 2009. p. 1827–1830.

KAWAHARA, D.; KUROHASHI, S.; INUI, K. Grasping major statements and their contradictions toward information credibility analysis of web contents. In: WEB INTELLIGENCE AND INTELLIGENT AGENT TECHNOLOGY - s.n, Sydney, Australia,2008. **Proceedings...** New York, NY, USA: IEEE, 2008. p. 393–397.

KENNEDY, A.; INKPEN, D. Sentiment classification of movie reviews using contextual valence shifters. **Computational intelligence**, v. 22, n. 2, p. 110–125, 2006.

KIM, H. D.; ZHAI, C. Generating comparative summaries of contradictory opinions in text. In: THE 18TH ACM CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT - 18th, Hong Kong, China, 2009. **Proceedings...** New York, NY, USA: ACM, 2009. p. 385–394.

LIU, B. Sentiment analysis and opinion mining. **Synthesis lectures on human language technologies**, v. 5, n. 1, p. 1–167, 2012.

LIU, B. et al. Text classification by labeling words **AAAI**. [S.l.: s.n.], 2004. v. 4, p. 425–430.

MANNING, C. D. et al. The Stanford CoreNLP natural language processing toolkit. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS (ACL) SYSTEM DEMONSTRATIONS - v. 1, Baltimore, MD, USA,2014. **Proceedings...** Stroudsburg, PA, USA: ACL, 2014. p. 55–60.

MARCU, D.; ECHIHABI, A. An unsupervised approach to recognizing discourse relations. In: THE 40TH ANNUAL MEETING ON ASSOCIATION FOR COMPUTATIONAL LINGUISTICS - 40th, Philadelphia, Pennsylvania ,2002. **Proceedings...** Stroudsburg, PA, USA: ACL, 2002. p. 368–375.

MARNEFFE, M. catherine D.; RAFFERTY, A. N.; MANNING, C. D. Finding contradictions in text. In: THE 46TH ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTIC - 46th, Columbus, OH, USA, 2008. **Proceedings...** Stroudsburg, PA, USA: Association for Computational Linguistics, 2008.

MEEKER, M. Internet trends 2015-code conference. **Glokalde**, v. 1, n. 3, 2015.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. **CoRR - Computing Research Repository - arXiv.org**, abs/1301.3781, 2013.

MILLER, G. A. Wordnet: a lexical database for english. **Communications of the ACM**, ACM, v. 38, n. 11, p. 39–41, 1995.

MIZUNO, J. et al. Organizing information on the web through agreement-conflict relation classification **Information Retrieval Technology**. [S.l.]: Springer, 2012. p. 126–137.

NAIRN, R.; CONDORAVDI, C.; KARTTUNEN, L. Computing relative polarity for textual inference. In: THE FIFTH INTERNATIONAL WORKSHOP ON INFERENCE IN COMPUTATIONAL SEMANTICS - 5th, Buxton, England, 2006. **Proceedings...** Stroudsburg, PA, USA: ACL, 2006. p. 20–21.

OXFORD Dictionaries. Jun 2016. Available from Internet: <http://www.oxforddictionaries.com/>.

PADÓ, S. et al. Deciding entailment and contradiction with stochastic and edit distance-based alignment. In: THE 1ST TEXT ANALYSIS CONFERENCE - 1st, Gaithersburg, Maryland, USA,2008. **Proceedings...** [S.l.]: NIST, 2008.

PANG, B.; LEE, L. Opinion mining and sentiment analysis. **Foundations and trends in information retrieval**, Now Publishers Inc., v. 2, n. 1-2, p. 1–135, 2008.

PANG, B.; LEE, L.; VAITHYANATHAN, S. Thumbs up?: sentiment classification using machine learning techniques. In: THE ACL-02 CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING - [s.n], Philadelphia, PA, USA, 2002. **Proceedings...** Stroudsburg, PA, USA: ACL, 2002. p. 79–86.

RAVI, K.; RAVI, V. A survey on opinion mining and sentiment analysis: tasks, approaches and applications. **Knowledge-Based Systems**, Elsevier, v. 89, p. 14–46, 2015.

RILOFF, E.; WIEBE, J.; PHILLIPS, W. Exploiting subjectivity classification to improve information extraction. In: THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE - [s.n], Pittsburgh, Pennsylvania,2005. **Proceedings...** [S.l.]: Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005. p. 1106.

RITTER, A. et al. It's a contradiction—no, it's not: a case study using functional relations. In: THE CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING - [s.n], Waikiki, Honolulu, Hawaii, USA, 2008. **Proceedings...** Stroudsburg, PA, USA: ACL, 2008. p. 11–20.

SANGANI, C.; ANANTHANARAYANAN, S. **Sentiment Analysis of App Store Reviews**. 2013. Available from Internet: <http://cs229.stanford.edu/proj2013/CS229-ProjectReport-ChiragSangani-SentimentAnalysisOfAppStoreReviews.pdf>.

SOCHER, R. et al. Recursive deep models for semantic compositionality over a sentiment treebank: THE CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING - [s.n], Seattle, USA, 2013. **Proceedings...** Stroudsburg, PA, USA: ACL, 2013. v. 1631, p. 1642.

TABOADA, M. et al. Lexicon-based methods for sentiment analysis. **Comput. Linguist.**, MIT Press, Cambridge, MA, USA, v. 37, n. 2, p. 267–307, jun. 2011. ISSN 0891-2017. Available from Internet: <http://dx.doi.org/10.1162/COLI_a_00049>.

TSYTSARAU, M.; PALPANAS, T. Survey on mining subjective data on the web. **Data Mining and Knowledge Discovery**, Kluwer Academic Publishers, v. 24, n. 3, p. 478–514, 2012.

TSYTSARAU, M.; PALPANAS, T.; DENECKE, K. Scalable detection of sentiment-based contradictions. **DiversiWeb, WWW**, Citeseer, v. 2011, 2011.

TURNEY, P. D. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: THE 40TH ANNUAL MEETING ON ASSOCIATION FOR COMPUTATIONAL LINGUISTICS - 40th, Philadelphia, Pennsylvania, 2002. ACL. **Proceedings...** Stroudsburg, PA, USA, 2002. p. 417–424.

UNIVERSITY, P. **What is WordNet?** 2005. Available from Internet: <https://wordnet.princeton.edu/wordnet/>.

VARGAS, D. S.; MOREIRA, V. P. Detecting contrastive sentences for sentiment analysis. In: THE BRAZILIAN SYMPOSIUM ON DATABASES - 30th, Quitandinha, Petrópolis, BR, 2015. **Proceedings...** [S.l.: s.n.], 2015.

VARGAS, D. S.; MOREIRA, V. P. Identifying sentiment-based contradictions. In: THE BRAZILIAN SYMPOSIUM ON DATABASES - 31th, Salvador, Bahia, BR, 2016. **Proceedings...** [S.l.: s.n.], 2016.

WIEBE, J. et al. Learning subjective language. **Computational linguistics**, MIT Press, v. 30, n. 3, p. 277–308, 2004.

WILSON, T.; WIEBE, J.; HWA, R. Just how mad are you? finding strong and weak opinion clauses. In: THE 19TH NATIONAL CONFERENCE ON ARTIFICAL INTELLIGENCE 19th, San Jose, California, USA 2004. **Proceedings...** [S.l.]: AAAI Press, 2004. p. 761–769.