UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN DE WEB OPINION MINING PARA IDENTIFICAR EL SENTIMIENTO DE USUARIOS DE TWITTER CON RESPECTO A UNA COMPAÑÍA DE RETAIL

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL

JORGE-ANDRÉS JEAN-MICHEL BALAZS THENOT

PROFESOR GUÍA:
JUAN VELÁSQUEZ SILVA

MIEMBROS DE LA COMISIÓN:
FRANCISCO MOLINA JARA
EDISON MARRESE TAYLOR

Parcialmente financiado por el proyecto INNOVA CORFO 13IDL2-23170 – OpinionZoom

---

SANTIAGO DE CHILE
SEPTIEMBRE 2015

Resumen Ejecutivo

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INDUSTRIAL
POR: JORGE BALAZS THENOT
FECHA: SEPTIEMBRE 2015
PROF. GUÍA: JUAN VELÁSQUEZ SILVA

DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN DE WEB OPINION MINING PARA IDENTIFICAR EL SENTIMIENTO DE USUARIOS DE TWITTER CON RESPECTO A UNA COMPAÑÍA DE RETAIL

Los contenidos disponibles en la Web están creciendo a velocidades que hacen que la tarea de analizarlos sea humanamente imposible. Una de las disciplinas que hace frente a este problema es la Minería de Opiniones, también conocida como el Análisis de Sentimientos, responsable de procesar texto automáticamente, con el fin de extraer y analizar las opiniones que contiene para generar información valiosa y accionable.

El objetivo principal de este trabajo es crear una aplicación de Minería de Opiniones capaz de explotar tweets en español que mencionen a la empresa de retail Falabella. En primer lugar, se investigó el impacto que las redes sociales tienen en Chile. En segundo lugar, se creó un estado del arte que englobara los últimos avances en Minería de Opiniones y en Procesamiento del Lenguaje Natural. En tercer lugar, se creó un *Web Crawler* capaz de obtener los tweets que mencionanaran a la compañía. Posteriormente se implementó varios algoritmos de Procesamiento del Lenguaje Natural para pre-procesar los tweets previamente mencionados, e incorporar los datos resultantes al proceso de extracción de opiniones. Este proceso se desarrolló como un enfoque de Minería de Opiniones no supervisado basado en lexicones, dependiente de un analizador de dependencias encargado de detectar ciertas estructuras gramaticales que permitieran identificar fenómenos linguísticos comunes, tales como la negación, intensificación, y oraciones subordinadas adversativas. La identificación de dichos fenómenos permitió mejorar la calidad de la clasificación. Finalmente se creó una página Web para mostrar los resultados que luego fueron utilizados para realizar un análisis exploratorio de la compañía.

Adicionalmente, los algoritmos fueron validados con el corpus TASS, obteniendo valores-F de un 61,88% negativo y 71,88% positivo. A pesar de que el rendimiento de los algoritmos no fue tan alto como una aplicación en producción lo requeriría, se consideró lo suficientemente bueno como para realizar el análisis exploratorio. Con éste fue posible confirmar la intuición de que las cuentas corporativas suelen publicar contenido positivo, las cuentas de noticias contenido neutral, y los usuarios comunes contenido irrelevante o quejas. Además fue posible probar que los usuarios más activos frecuentemente publican contenido totalmente irrelevante. Por otra parte, se logró replicar varios resultados obtenidos por instituciones nacionales reconocidas, entre los cuales destaca el hecho que el momento más controversial del año para Falabella fue cuando se intentó llevar a cabo el Cyber Monday, período en el cual el sentimiento generalizado en Twitter alcanzó los niveles más negativos. Dicho todo esto, la aplicación desarrollada demostró ser útil al momento de utilizar una gran cantidad de datos para extraer información que podría ser potencialmente útil para la firma de retail.

Finalmente, el desarrollo de la aplicación permitió crear un artículo que contuviera parte considerable del transfondo teórico en el cual ésta se basó, además de beneficiar a otros estudiantes en el desarrollo de sus memorias.

Abstract

ABSTRACT OF THE THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
INDUSTRIAL ENGINEER
BY: JORGE-ANDRÉS BALAZS THENOT
DATE: SEPTEMBER 2015
ADVISOR: JUAN VELÁSQUEZ SILVA

DESIGN, DEVELOPMENT AND IMPLEMENTATION OF A WEB OPINION MINING APPLICATION FOR IDENTIFYING TWITTER USER'S SENTIMENT TOWARDS A CHILEAN RETAIL COMPANY

The contents available on the Web are growing at rates that make the task of analyzing them humanly impossible. Opinion Mining, also know as Sentiment Analysis, is a field that attempts to tackle this problem by automatically processing and analyzing raw text, in order to extract the opinions it contains for generating insightful and actionable information.

The main objective of this study is to create an Opinion Mining application capable of exploiting Spanish tweets mentioning the retail company Falabella. First, a research to evaluate the impact social networks have in Chile was carried out. Second, a state of the art encompassing the latest Opinion Mining and Natural Language Techniques was created. Third, a Web Crawler capable of obtaining tweets mentioning the retail company was developed. Next, different Natural Language Processing algorithms were implemented for pre-processing these tweets and feeding the resulting data to the opinion extraction process. This process was developed as an unsupervised lexicon-based approach for Opinion Mining relying on a dependency parser for detecting certain grammatical structures, which would allow to identify frequent linguistic phenomena such as negation, intensification and adversative clauses. Accordingly, the identification of these phenomena improved classification performance. Later, a webpage for displaying the results was created, and finally, the data were exploited to perform an exploratory analysis concerning the retailer.

Additionally, the algorithms were validated with the TASS corpus, obtaining 61.88% negative and 71.88% positive F-measures. Even though algorithm performance was not as high as a production-level application would require, it was deemed high enough for performing the presented exploratory analysis. With it, it was possible to confirm the intuition that corporate accounts often post positive content, news accounts post neutral content, and common users post either irrelevant content or complaints. Further, it was possible to prove that the top most active users often post content that is totally irrelevant. Moreover, several results from studies performed by renowned national institutions were replicated, namely, the most controversial time of the year for the company occurred when it hosted the Cyber Monday shopping event, which is when the overall Twitter sentiment for Falabella reached its most negative levels. All this being said, the developed application proved to be useful in exploiting a great a mount of data for extracting information that could be potentially useful for the retail firm.

Finally, the development of the application spun off the creation of a paper containing a considerable part of the theoretical foundations upon which it was built, and it benefited some students with the development of their own theses.

Computers bootstrap their own offspring, grow so wise and incomprehensible that their communiqués assume the hallmarks of dementia: unfocused and irrelevant to the barely-intelligent creatures left behind. And when your surpassing creations find the answers you asked for, you can't understand their analysis and you can't verify their answers. You have to take their word on faith – or you use information theory to flatten it for you, to squash the tesseract into two dimensions and the Klein bottle into three, to simplify reality and pray to whatever Gods survived the millennium that your honorable twisting of the truth hasn't ruptured any of its load-bearing pylons.

– Peter Watts, Blindsight

Agradecimientos

Quiero agradecer a mi profesor guía, Juan Velásquez, por su apoyo incondicional y su confianza infalible.

A Francisco Molina por haberme orientado en los inicios del trabajo.

A todos mis compañeros de La Salita, Gaspar Pizarro, Patricio Moya, Yerko Covacevich, Gino Slanzi, Eduardo Neira, Luis Valdés y Victor Hernández, sin quienes el trabajo hubiera sido considerablemente menos llevadero.

A los amigos con los que compartí la experiencia universitaria, Fernando Galdames, Matía de Luiggi, Matías del Rey, Diego Granger e Ignacio Canales.

A mi familia, por haberme apoyado en la decisión de embarcarme en este proyecto y por haber sido una motivación constante.

Y finalmente a Lía, mi compañera, guía, y fuente de inspiración, quien me brindó su apoyo durante todo el proceso y me ayudó a cumplir la meta.

Table of Contents

1 Introduction **1**
- 1.1 Context . . . 2
  - 1.1.1 OpinionZoom Project Description . . . 2
  - 1.1.2 Chile and the Social Networks . . . 2
  - 1.1.3 Chilean Retail Industry Presence in the Web . . . 3
  - 1.1.4 Opportunities . . . 5
- 1.2 Objectives . . . 6
  - 1.2.1 General Objective . . . 6
  - 1.2.2 Specific Objectives . . . 6
- 1.3 Research Hypothesis . . . 7
- 1.4 Methodology . . . 7
- 1.5 Expected Results . . . 7
- 1.6 Contributions . . . 8
- 1.7 Thesis Structure . . . 8

2 Conceptual Framework **10**
- 2.1 Opinion Mining . . . 10
  - 2.1.1 Definitions . . . 11
  - 2.1.2 Opinion Mining Pipeline . . . 13
  - 2.1.3 Opinion Mining Surveys . . . 17
- 2.2 Opinion Mining Core Process . . . 19
  - 2.2.1 Different Levels of Analysis . . . 19
  - 2.2.2 Different Approaches . . . 21
- 2.3 Natural Language Processing . . . 23
  - 2.3.1 Text Preprocessing in NLP . . . 24
  - 2.3.2 Lexical Analysis . . . 28
  - 2.3.3 Syntactic Analysis . . . 31
  - 2.3.4 Semantic Analysis . . . 39
  - 2.3.5 Natural Language Processing in Opinion Mining . . . 41
- 2.4 Twitter . . . 42
  - 2.4.1 Overview . . . 42
  - 2.4.2 Companies and Twitter . . . 44
  - 2.4.3 Opinion Mining in Twitter . . . 48

3 Design **53**
- 3.1 Software Requirements . . . 53
- 3.2 General Architecture . . . 54
- 3.3 Data Characteristics . . . 55
- 3.4 Data Extraction Module . . . 57
- 3.5 Preprocessing Module . . . 58
- 3.6 Polarity Classification Module . . . 60
  - 3.6.1 Dependency Parser . . . 61
  - 3.6.2 Rule Applier . . . 63
  - 3.6.3 Aggregator . . . 67
- 3.7 Visualization Module . . . 67

3.7.1 Architecture . . . 67
3.7.2 Web Platform Prototype . . . 68

4 Implementation **73**
4.1 Third-Party Resources . . . 73
4.1.1 Environment . . . 73
4.1.2 MongoDB . . . 74
4.1.3 TreeTagger . . . 75
4.1.4 MaltParser . . . 76
4.1.5 Natural Language Toolkit (NLTK) . . . 77
4.1.6 Django . . . 77
4.2 Data Extraction . . . 78
4.3 Preprocessing . . . 80
4.3.1 Cleaner . . . 80
4.3.2 Corrector . . . 81
4.3.3 Sentence Segmenter . . . 82
4.3.4 Tokenizer . . . 82
4.3.5 Tagger . . . 83
4.4 Polarity Classification . . . 84
4.4.1 Obtaining the Dependencies . . . 84
4.4.2 Application of the Rules . . . 86
4.4.3 Overall Tweet Polarity . . . 90
4.5 Visualization . . . 90
4.6 Implementation Architecture . . . 92

5 Validation and Case Study **94**
5.1 Validation . . . 94
5.1.1 Evaluation Metrics . . . 95
5.1.2 Validation Corpus Description . . . 96
5.1.3 Validation Process and Results . . . 98
5.2 Retail Case Study . . . 103
5.2.1 Dataset Characterization . . . 104
5.2.2 Polarity Analysis . . . 108

6 Conclusions **117**
6.1 Synthesis . . . 117
6.2 Limitations . . . 119
6.3 Implications . . . 119
6.4 Future Work . . . 120
6.5 Closing Remarks . . . 122

Bibliography **123**

Appendix **135**
A Paper: "Opinion Mining and Information Fusion: A survey" . . . 135
B AnCora Tags for Syntactic Dependency . . . 152
C AnCora Constituents . . . 153
D AnCora Relationship Between Dependencies and Constituents . . . 153
E CoNLL File Example . . . 154
F CoNLL Columns . . . 155
G Some Abbreviations Obtained from 50000 tweets . . . 156
H Example of a Sentence Represented as a NLTK Dependency Graph . . . 156
I Full Algorithm for Applying the Heuristics . . . 157

J Interpretation of Kappa 158

List of Tables

1.1 Top 10 Most Popular Social Networking Sites in Chile. 3
1.2 Top 10 Most Popular E-Commerce Sites in Chile (Thousands of Unique Visitors). 4

2.1 Some Emoticons and Their Most Common Associated Emotion 49

4.1 Regular Expressions used for deleting unwanted text. 80
4.2 Regular Expressions used for replacing text. 81

5.1 Confusion Matrix Example. 95
5.2 TASS Corpus Characteristics. 97
5.3 TASS Topic Distribution. 98
5.4 TASS Polarity Distribution. 98
5.5 Confusion Matrix for 5 Classes. 99
5.6 Confusion Matrix for 3 Classes. 101
5.7 Performance Metrics for 3 Classes. 101
5.8 Confusion Matrix for 2 Classes, ignoring `NONE` and `NEU` tags. 101
5.9 Performance Metrics for 2 Classes, ignoring `NONE` and `NEU` tags. 101
5.10 Confusion Matrix for 2 Classes, ignoring `NONE` and `NEU` tags, and assuming perfect filtering of neutral tweets. 102
5.11 Performance Metrics for 2 Classes, ignoring `NONE` and `NEU` tags, and assuming perfect filtering of neutral tweets. 102
5.12 Performance metrics with different treatments for `NEU` and `NONE` tags. 103
5.13 Baseline Comparison. 103
5.14 The 20 Most Active Users and the Type of Content They Usually Post. . . . 106
5.15 24 Frequent Retail-related Keywords. 115

List of Figures

2.1 Opinion Mining Pipeline. . . . 13
2.2 Natural Language Processing stages. . . . 24
2.3 Natural Language Processing in the text preprocessing step of the Opinion Mining process. . . . 24
2.4 Parse trees representing two noun phrases. . . . 35
2.5 Parse tree of a sentence in English and Spanish. . . . 36
2.6 Dependency structure for an English sentence with Stanford-typed dependencies. 38
2.7 Dependency structure for an English sentence represented as a tree. . . . 38
2.8 Dependency structure for an English sentence with adjectives. . . . 38
2.9 Dependency structure for an English sentence with adjectives represented as a tree. . . . 39
2.10 Dependency structure for a Spanish sentence with AnCora-ES typed dependencies. . . . 39
2.11 How microblogging affects branding components. . . . 47

3.1 System General Architecture. . . . 55
3.2 Data Extraction Module Architecture. . . . 57
3.3 Preprocessing Module Architecture. . . . 58
3.4 Polarity Classification Module Architecture. . . . 61
3.5 Dependency trees representing the sentences of a tweet. . . . 62
3.6 Dependency tree of a sentence with intensification. . . . 63
3.7 Polarity propagation of an intensified sentence. . . . 64
3.8 Subjective parent rule representation. . . . 65
3.9 Subject Complement – Direct Object Rule representation. . . . 65
3.10 Adjunct Rule representation. . . . 65
3.11 Default Rule representation. . . . 65
3.12 Example of the application of the Subjective Parent rule. . . . 66
3.13 Example of a sentence with a restrictive conjunction. . . . 67
3.14 Visualization Module Architecture. . . . 68
3.15 Website Landing Page. . . . 69
3.16 View of the tweets tagged as “Very Positive.” . . . 70
3.17 View of the tweets tagged as “Very Negative.” . . . 70
3.18 View of the graph search result of the keyword “servicio.” . . . 71
3.19 View of the list search result of the keyword “servicio.” . . . 71
3.20 Statistics View: Polarity Distribution. . . . 72
3.21 Statistics View: Polarity Label Distribution. . . . 72

4.1 Implementation Architecture. . . . 92

5.1 % of Users and % of Tweets With Respect to the Amount of Tweets per User. 104
5.2 Cumulative % of Tweets With Respect to Cumulative % of Users. . . . 105
5.3 Top 20 Most Active Users. . . . 106
5.4 The Most Recurrent Content Posted by the Top 100 Most Active Users. . . . 107
5.5 Daily Tweet Frequency. . . . 107

5.6 Average Polarity With Respect to the Content Type of the 100 Most Active Accounts. 109
5.7 Content Type Polarity Distribution. 110
5.8 User Average Polarity Distribution. 111
5.9 Normalized Daily Tweet Frequency and Polarity. 112
5.10 24 Frequent Retail-related Keywords. 116

Chapter 1

Introduction

The contents available on the Web have been exponentially growing since it first reached the general public, and even more today with the appearance of the Web 2.0 where the users are granted the ability to modify the contents of web pages and collaborate with others to become the publishers. The platforms where users can contribute with their own content are many, possess varied structures and pursue different goals. Blogs, microblogs, forums, news sites, e-commerces and review sites for movies, books, restaurants, travel places and games, are just a few of the available channels for users to produce insightful content, and each represents a rich source of information from which others can greatly benefit.

However, when the content grows at rates that are being observed today, it becomes humanly impossible to read it entirely, let alone understand it and obtain significant information from its entirety. This is why automated systems become increasingly relevant since they have, or should have, the ability to parse this unstructured content, somewhat understand it, summarize it and make it human-readable in a time scale that allows to incorporate the insights obtained to the decision-making process of a human, organization, business or even another automated system.

Fields such as *Data Mining* are responsible for processing great amounts of structured data and producing meaningful and actionable information from them. Nevertheless, new challenges related with the massive amount of unstructured data produced by users in the many aforementioned platforms are becoming more evident. Indeed, computers have yet to understand the meaning of what humans are saying on the Web. *Natural Language Processing* (NLP) is the field charged with the task of making human language understandable by computers and, as such, is vital to communicate Data Mining algorithms with human generated input. The challenge lies in creating systems able to handle varying qualities of input, since, on the Web, users don't always feel compelled to abide by orthographic or grammatical rules.

One of the many applications for NLP is understanding human speech to extract subjective opinions from it. The usefulness of this application rests on the fact that it is common for humans to look for opinions from other humans before making a decision. The same

could be said for commercial organizations, since in order to make larger profits they have to better understand their clients' needs which are usually expressed as opinions. *Opinion Mining*, also known as *Sentiment Analysis*, is the field accountable for processing raw text in order to extract the opinions contained in it, and later processing them to generate useful information.

The economic benefits of having an Opinion Mining tool could be undoubtedly very valuable. Such a tool would be able to automatically keep track of what clients feel toward a brand, enabling a company to know which actions to take to improve their consumer loyalty, or to know which features of a product to improve to increase its sales. With this information companies could save considerable amounts of money in expensive market research studies that would produce similar outcomes.

1.1 Context

1.1.1 OpinionZoom Project Description

This study is developed under the project INNOVA CORFO 13IDL2-23170: *"OpinionZoom - Plataforma de análisis de sentimientos e ironía a partir de información textual en redes sociales para la caracterización de la demanda de productos y servicios"* which literally translates to "OpinionZoom - Sentiment and irony analysis platform to characterize the demand of products and services from social network textual information."

The final goal of this project is to produce a tool that is capable of extracting sentiment and irony from unstructured textual data, and exploit them to characterize the demand of a given product or service. The goal of this study, however, is to create a functional prototype capable of detecting tweet polarity by applying a syntactic-based Opinion Mining approach in the sentence level.

1.1.2 Chile and the Social Networks

To understand the relevance of this work, it is first necessary to know the Chilean context in which it is situated. Chile has a yet small but growing Internet population. According to [1], [2] and [3] the Internet population grew from 59.8% of the total population in 2011 to 65.4% in 2012 and from there to 66% in 2013, accounting for a 6.2% growth between 2011 and 2013 (considering that the Chilean population did not vary greatly in this period of time). Compared to the 2% growth of Europe or North America, this 6.2% might seem high, however both these continents have a much higher penetration, 76% and 78.6% respectively [4]. This difference in Internet penetration could be explained by the fact that both the education quality and income in Chile are lower than in the aforementioned continents, hence the opportunity to buy a computer or learn how to use it are less likely to occur. This idea is further reinforced by the reason given in the study reported in [2] for not accessing Internet, which was "not knowing how to use a computer".

Furthermore, according to a more recent study [5], Chileans spend on average 17.6 hours

online per month, which is lower than the Latin American average of 21.7 hours and the global average of 22.8 hours a month. These indicators show that Internet usage in Chile has much space to grow both in terms of penetration and usage time.

Additionally, Internet population is composed mostly by teenagers and young adults: 34.3% between 15 and 24 years of age, and 26.4% between 25 and 34. These numbers are similar the averages in Latin America, but quite different from those in Europe and North America, where Internet usage is more evenly split among age ranges.

Concerning social network usage, 76% of the 66% total Internet population, uses social networking sites [3]. 96% of this 76% uses them for keeping in touch with their family and friends, 79% to share views about music and movies and less than 30% to share views about politics or religion. The most popular social networking website is Facebook with 4,925,000 unique visitors, followed by Taringa with 2,443,000 and LinkedIn with 2,393,000. Table 1.1 shows the 10 most visited social networking sites in Chile:

Site	**Unique Visitors (Thousands)**
Facebook	4925
Taringa	2443
LinkedIn	2393
Ask.fm	833
Twitter	790
Tumblr	662
Scribd	238
Deviantart	238
Badoo	217
Pinterest	144

Table 1.1: Top 10 Most Popular Social Networking Sites in Chile.
Source: [5].

Moreover, 30.1% of the time spent online is devoted to social networks, being this the most popular activity engaged by Chilean Internet users, followed by the "Services" category (e-mail, calendar, etc.) corresponding to 24.4% of the time and the "Entertainment" category corresponding to 17% [5]. This share of online time spent on social media is one of the highest in the world along with Italy(29%), Malaysia(39%) and Philippines (41%), while the lowest correspond to France(15%), South Korea(8%) and Japan(4%) [6].

Finally, 96% of the time devoted to visiting social networking sites by Chileans is monopolized by Facebook, followed by a 2.2% pertaining to Twitter. This shows that Facebook is not only the leader in terms of unique visitors but also in terms of the time people spend in it.

1.1.3 Chilean Retail Industry Presence in the Web

Retail presence on the Web has become very important with the passing of time. As it happened, in 2013 e-commerce sales surpassed US$70 billion in Latin America, 40 times more

than the US$1.6 billion observed in 2003 [7]. In Chile the year 2013 accounted for US$1.6 billion in online retail sales, 25% more than the previous year. This number was expected to exceed US$2 billion in 2014 and to grow between 20% and 30% in 2015. Additionally, the number of Chilean commerces with online presence grew from 1253 in 2011 to 2857 in 2013; a 128% growth in 2 years.

Table 1.2 shows the most popular e-commerce sites and the variation in unique visitors between 2013 and 2014:

Site	**Unique Visitors 2013**	**Unique Visitors 2014**	**% Variation**
MercadoLibre	1567	1270	**–19.0%**
Cencosud	1022	955	**–6.6%**
Falabella	1052	950	**–9.7%**
Sodimac	643	723	**+12.4%**
Amazon	730	651	**–10.8%**
Ripley	616	570	**–7.5%**
Buscape	568	408	**–28.2%**
eBay	370	406	**+9.7%**
Alibaba	193	390	**+102.1%**
Apple	433	312	**–27.9%**

Table 1.2: Top 10 Most Popular E-Commerce Sites in Chile (Thousands of Unique Visitors). Source: [5].

The first thing that comes to focus is that almost every site lost visitors from one year to the next, with Buscape, Apple and MercadoLibre being the most affected. On the other hand, Sodimac, a home improvement retailer, and eBay gained a fair amount of unique visitors, while the user base of Alibaba more than doubled. This could be explained partly because Chileans do not consider local e-commerces to be very good [6]. This fact could also shed light on the reason why 6 of the top 10 most popular e-commerce sites in Chile are international instead of local, including the first, MercadoLibre.

Otherwise, according to [8], the most mentioned retails in Twitter, without discriminating between positive or negative mentions, are Falabella, Ripley, La Polar, Paris and Johnson's[1]. It is also worth mentioning that the highest peak in mentions for Falabella was due to consumers complaining for the failed attempt to imitate the United States's "Cyber Monday", since the retailer's web page could not handle the higher-than-average amount of requests it received in said date. Another interesting fact revealed in the study is that a vast amount of peaks are due to users commenting on negative events. For instance, the highest peak for Jumbo (supermarket belonging to Cencosud) was observed in December 2013 and due to a strike that had prolonged for more than 45 days, while the highest one for La Polar was observed in the same month and was due to users commenting on news concerning misleading advertising from the retailer.

[1]Cencosud, the second most popular e-commerce shown in table 1.2, is indirectly mentioned in the study and falls in the category of "supermarket" instead of "retail" because it mostly focuses in selling groceries.

Further, in [7], Falabella appears to be the Chilean e-commerce with the highest amount of complaints addressed to the official authorities[2] both in 2013 and 2014. The main reasons for these complaints are:

- Non-compliance with the service condition agreements.
- Delays in product shipping times.
- Deficient quality of service.

Finally, the same report states that the most valued features of an e-commerce by the Chilean consumers are:

- Comfort in buying from home and not having to physically go to the retail store.
- Lower prices.
- Higher variety of products and brands.
- The ability to compare product prices and quality, brands, service-level, etc.

1.1.4 Opportunities

Up to this point, it is clear that there is much room for improvement. It is now known that Chilean consumers complain against local e-commerces mostly because of violations to the agreement terms, however the study presented in [7] doesn't specify exactly which terms are usually breached nor why the quality of service is deemed deficient. Still, the concerned e-commerce could ask for the data with which the report was made, but it would probably be disaggregate (each individual complaint), meaning the retail would have to spend a considerable amount of time and resources to analyze it manually. They could also refer to a study such as [8] but then the problem would be the opposite; the information presented in it is too coarse since it just reports the total amount of mentions in Twitter without any further details from them besides of the explanation for some peaks.

In section 1.1.2, it was shown that social networking sites in Chile are vastly popular, with Facebook in the lead and Twitter among the top 10. Retailers are aware of this which is why they usually possess online profiles. Taking advantage of the vast amounts of data generated in these sites would be greatly beneficial for the industry. In the previous section it was also made clear that consumers value the ability to compare product prices and quality, but Chilean e-commerces don't offer tools to make this comparison easier.

All of these issues reflect the need for a better system to obtain user-generated data, process them, and generate insightful information from them. With such a system retailers would be able to better understand what their customers want, making better decisions for

[2] *Servicio Nacional del Consumidor (SERNAC)* - Consumer's National Service

improving their business. Another example would be to use a similar system to offer better information to consumers about the product they are visiting. Moreover, with it, companies in charge of market research could offer a more fine-grained analysis of the social networks.

Admittedly, not seizing the opportunity to automatically analyze the data generated in social networking sites, in order to better understand the customers, would only deepen the problems e-commerces are facing today, such as the growing amount of complaints and the consumer leak towards international e-commerces.

1.2 Objectives

1.2.1 General Objective

The general objective of this thesis is to design, develop and implement an opinion mining platform, able to detect the sentiment from Twitter users towards a Chilean retail company and display the results in a user-friendly fashion, providing insights to support the industry's decision-making process.

1.2.2 Specific Objectives

To accomplish the general objective, several steps must be completed first:

1. Thoroughly investigate the state-of-the-art in *Opinion Mining*.
2. Design the logic structures and algorithms that will be involved in the whole information extraction process.
3. Implement the previously designed elements by applying the knowledge extracted from the state-of-the-art study.
4. Validate the algorithms by comparing the obtained results with known data and calculating performance metrics.
5. Visualize the validated results in a way that is easy for the user to understand and provides insightful information to aid in the retail company's mid-term decision-making process.

1.3 Research Hypothesis

There is a vast amount of user-generated data available on social networking sites such as Twitter and exploiting this data might yield information that could allow retailers to make better-informed decisions.

More specifically, by applying a syntactic-based opinion mining approach to a set of tweets and combining it with topic information, it should be possible to extract underlying information useful for most industries.

1.4 Methodology

This thesis was developed in the span of a year and a month, beginning on August 2014 and ending on September 2015. The initial period, spanning the first six months from August 2014 to January 2015, was devoted to deepening the knowledge on Opinion Mining by reading the available literature on the topic, in addition to creating the first prototype of the platform. During this period of time, the most relevant achievement was the creation of the earliest versions of the modules presented in Chapter 3 and Chapter 4. In order to do so, it was necessary to learn how to use Version Control Systems, a new Operating System, and several other tools including the ones presented in 4.1.

The second period, spanning from February the 9^{th} to March the 27^{th} was devoted to researching and writing the paper presented in [9]. Later, from March the 30^{th} to May the 8^{th} the whole time was assigned to writing the thesis, then, on May the 11^{th} the revisions for the paper were received, and the process of correcting and improving it begun and lasted until June the 2^{nd}, when it was resubmitted. Lastly, from June the 3^{rd} onward, the time was committed to finishing the thesis.

Additionally, after having read several books on Software Development, in particular the book by Steve McConnell [10], the whole code of the Opinion Platform was rewritten to adopt better practices, make it readable for future students to exploit it, easier to extend, and modular, which in turn allowed to create an Application Programming Interface for the rest of the research group to use.

Finally, some scripts were created to validate the current platform against the TASS corpus, presented in Section 5.1.2, and then the study concerning Falabella was carried out.

1.5 Expected Results

The expected results of this thesis are:

- A conceptual framework including the state of the art in the topics exploited for the creation of the thesis, namely Opinion Mining and Natural Language Processing.
- An annotated dataset containing tweets mentioning Falabella and their corresponding

polarity.

- The results of applying a methodology that was created for Spanish reviews to Spanish tweets, considering they are fundamentally different.
- A platform for extracting, processing and viewing tweets mentioning the retail company Falabella.

1.6 Contributions

A considerable part of the research carried out for creating the Conceptual Framework of this thesis was used for writing the paper *Opinion Mining and Information Fusion: A survey* [9], in collaboration with Juan D. Velásquez, to be published in the journal Information Fusion by Elsevier in January 2016. The paper is attached in Appendix A, and is also available online in `http://dx.doi.org/10.1016/j.inffus.2015.06.002`.

Additionally, a simple API was created to use the polarity classification algorithms presented in this thesis. At the time of writing, two other students have benefited from using it for their own theses.

1.7 Thesis Structure

The thesis is structured as follows. Chapter 2: Conceptual Framework, introduces and describes the conceptual elements that will play a relevant role in the development of this work. It covers the typical Opinion Mining pipeline, and includes a deeper analysis of the levels of analysis and different approaches to tackle it. Additionally it comprises the most relevant aspects of Natural Language Processing that are used in the following chapters, and it presents a brief analysis of the microblogging platform Twitter, how it is relevant for the industry, and how it has interacted with Opinion Mining lately.

Chapter 3: Design, describes the design of the Opinion Mining application created by using the knowledge presented in Chapter 2. In particular, it mentions the previous requirements for building the platform, describes its general architecture, the characteristics of the data to be extracted, and each module comprising the application. The main goal of this chapter is to provide the reader with the understanding of *what* was done without the need of understanding how it was done.

Chapter 4: Implementation, describes *how* the application was built in more technical terms. First, it mentions the third-party resources upon which the rest of the platform relies, second, it describes each module in a lower level of abstraction, and finally, it indicates how all of the modules interact with each other.

Chapter 5: Validation and Case Study, is divided in two major subsections. The first deals with describing how the polarity classification algorithm was validated and the results issued from it. The second presents the application of the platform to data related to the

retail company Falabella. More specifically, the analyzed dataset is described, and then an analysis of the polarity of several features characterizing the dataset is carried out.

Finally, Chapter 6: Conclusions, concludes the whole work by synthesizing the previous chapters, pointing out the limitations of the platform, its implications, and how it can be improved.

Chapter 2

Conceptual Framework

The aim of this chapter is to introduce and describe the conceptual elements that will play a relevant role in the remainder of this work. Section 2.1 defines Opinion Mining, explains the steps that compose the OM process, and exhibits some of the latest surveys that cover this field. Section 2.2, shows a deeper analysis of the OM core process by enumerating the different levels of analysis at which it is performed, and the existing approaches to tackle it. Section 2.3 introduces the field of Natural Language Processing (NLP), explains the main steps in a generic NLP process, and describes the role NLP plays in Opinion Mining. Finally, section 2.4 gives an overview of the popular microblogging platform Twitter, explains why this platform might be important for some companies, and reviews the latest Opinion Mining advancements applied to Twitter.

2.1 Opinion Mining

Merriam-Webster's Online Dictionary[1] defines an opinion as a belief, judgment or way of thinking about something. Opinions are formed by the experiences lived by those who hold them. A consumer may look for another's opinion before buying a product or deciding to watch a movie, to gain insights into the potential experiences they would have depending on the decisions they make. Moreover, businesses could benefit from knowing the opinions of their customers by discovering cues on what aspects of a certain service to improve, which features of a determined product are the most valued, or which are new potential business opportunities [11, 12]. In essence, a good Opinion Mining system could eliminate the need for polls and change the way traditional market research is done.

[1] *http://www.merriam-webster.com/dictionary/opinion*

2.1.1 Definitions

General Definition

Opinion Mining is the field charged with the task of extracting opinions from unstructured text by combining techniques from Natural Language Processing (NLP) and Computer Science.

Bing Liu [13] defines an opinion as a 5-tuple containing the target of the opinion (or *entity*), the attribute of the target at which the opinion is directed, the sentiment (or polarity) contained in the opinion which can be positive, negative or neutral, the opinion holder and the date when the opinion was emitted. Formally, an opinion is defined as a tuple:

$$(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$$

where e_i is the i-th opinion target, a_{ij} is the j-th attribute of e_i, h_k is the k-th opinion holder, t_l is the time when the opinion was emitted and s_{ijkl} is the polarity of the opinion towards the attribute a_{ij} of entity e_i by the opinion holder h_k at time t_l.

Note that the sentiment contained in an opinion was described as positive, negative or neutral, notwithstanding it could also be numerically represented. For instance -5 could denote a very negative opinion while 5 a very positive one. Also, in case the analysis did not require much level of detail, the attributes of an entity could be omitted and denoted by $GENERAL$ instead of a_{ij}.

Therefore the main objective of Opinion Mining is to find all the opinion tuples $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$ within a document, collection of documents (called *corpus*) or across many corpora. Other works define Opinion Mining as "the task of identifying positive and negative opinions, emotions and evaluations" [14], "the task of finding the opinions of authors about specific entities" [15], "tracking the mood of the public about a particular product or topic" [16], or simply "the task of polarity classification" [17]. These definitions present different scopes and levels of granularity, however all of them can be adapted to fit Liu's opinion model.

There are other approaches, like the one presented in [18], in which the authors attempt to classify emotional states such as "anger", "fear", "joy", or "interest" instead of just positive or negative. In this case, Liu's model could be enriched by adding another element to the opinion tuple model to represent this information.

Types of Opinion

Opinions can be classified into two main categories, *regular opinions* and *comparative opinions*. Additionally, both categories can be further subdivided into direct (or explicit) opinions, and indirect (implied, implicit) opinions. Bing Liu offers a classification similar to this one in [19] but here a simplified version is presented[2]

[2]In his work he makes a difference between explicit versus implicit opinions, and direct versus indirect opinions, however the distinction is not clearly marked. For this reason, here an explicit opinion is considered as equivalent to a direct opinion and an implicit opinion to an indirect one.

- **Regular Opinions:** Regular opinions are those that refer to a single entity or aspect of an entity. This definition corresponds to the one presented in the previous section (2.1.1: General Definition).
  - **Direct Regular Opinions:** Direct regular opinions, or simply direct opinions, are those that explicitly refer to an entity or one of its aspects. They are the most simple to handle and most of the research focuses on them (as does this thesis). An example of such opinion in Spanish and its translation to English is:

    (2.1) *Falabella tiene una pésima atención al cliente.*

    (2.2) *Falabella has terrible customer service.*

    In which the aspect "customer service" of entity "Falabella" is being directly judged with a negative polarity.
  - **Indirect Regular Opinions:** Indirect opinions are those that are expressed indirectly on an entity or one of its aspects by the observable effects they have on other elements. For example:

    (2.3) *Terminé el juego en dos horas.*

    (2.4) *I finished the game in two hours.*

    Indirectly states that the aspect "playtime" of the entity "game" was "short"[3] which could be considered as positive or negative depending on the context. However "playtime" was not directly mentioned and had to be inferred from the fact that the opinion emitter finished it in two hours.
- **Comparative Opinions:** A comparative opinion points to the degree of similarity or difference between two or more entities or aspects of those entities. Like regular opinions, these can be subdivided into direct comparative opinions and indirect comparative opinions.
  - **Direct Comparative Opinions:** A direct comparative opinion explicitly states the degree of difference or similarity between two or more elements. For example:

    (2.5) *El servicio al cliente de Falabella es mejor que el de Ripley.*

    (2.6) *Falabella's customer service is better than Ripley's.*

    The example clearly states that the aspect "customer service" of the retail store Falabella is better than that of Ripley. Furthermore, with some additional information it would be possible to set the polarity of both stores in an absolute scale. For instance knowing that Falabella's customer service is bad in an absolute scale and is better than Ripley's in a relative scale, then in follows that Ripley's customer service is even worse in the absolute scale.
  - **Indirect Comparative Opinions:** Indirect comparative opinions point to the degree of difference or similarity between two or more elements without explicitly mentioning the aspect that is being compared. An example of such an opinion is:

[3] In some specific contexts this could be deemed as a long time but for the sake of the example it will be considered as little time for beating a game.

(2.7) *Los vendedores de Falabella siempre te atienden con una sonrisa mientras que los de Ripley no.*

(2.8) *Falabella's associates always assist you with a smile on their faces whereas Ripley's don't.*

Here, the fact that Falabella's associates assist shoppers with a smile on their faces is linked to the "customer service" aspect must be inferred. Again, for humans this task is subconscious and requires little to no effort, but to program this into a computer is an entirely different matter.

Little research has been carried out for opinions other than Direct Regular Opinions. In this thesis all of the efforts will be put in processing this same kind of opinions, given the complexity of dealing with the others and the time limitations.

2.1.2 Opinion Mining Pipeline

The usual Opinion Mining process or pipeline usually consists of a series of defined steps [20–22]. These correspond to corpus or data acquisition, text preprocessing, Opinion Mining core process, aggregation and summarization of results, and visualization (see figure 2.1). In this section, a brief description of each of these steps is given.

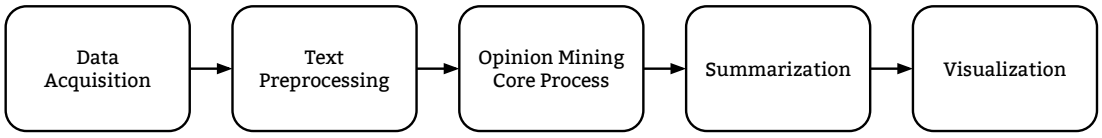


Figure 2.1: Opinion Mining Pipeline.

Data Acquisition

The first step of any Opinion Mining pipeline is called corpus or data acquisition and consists of obtaining the corpus that is going to be mined for opinions. Currently there are two approaches to achieving this task. The first is through a website's Application Programming Interface (API) being Twitter's[4] one of the most popular [22–25]. The second corresponds to the use of Web crawlers in order to scrape the data from the desired websites [26–28]. Olston and Najork portray a robust survey of Web crawling in [29].

Both approaches present some advantages and disadvantages so there is a trade-off between using either. In [30] the authors briefly compare them.

With the API-based approach the implementation is easy, the data gathered is ordered and unlikely to change its structure, however it presents some limitations depending on the provider. For instance search queries to the Twitter API are limited to 180 per 15-minute

[4] `http://dev.twitter.com/rest/public`, Accessed September 09, 2015

time window.[5] This approach is also subject to the availability of an API since not all websites provide one, and even if they do it might not present every needed functionality.

In contrast, crawler-based approaches are more difficult to implement, since the data obtained is noisier and its structure is prone to change, but have the advantage of being virtually unrestricted. Still, using these approaches requires to respect some good etiquette protocols such as the *robots exclusion standard*,[6] not issuing multiple overlapping requests to the same server, and spacing these requests to prevent putting too much strain on it [29]. Furthermore, Web crawlers can prioritize the extraction of subjective and topically-relevant content. In [31], the authors propose a focused crawler that collects opinion-rich content regarding a particular topic and in [32] this work is further developed by proposing a formal definition for sentiment-based Web crawling along with a framework to facilitate the discovery of subjective content.

Text Preprocessing

The second step in the OM pipeline is Text Preprocessing and is charged with common NLP tasks associated with lexical analysis [33]. In section 2.3, NLP is explained in finer detail but here some of the most common techniques for this particular step of the OM process are mentioned:

Tokenization: task for separating the full text string into a list of separate words. This is simple to perform in space-delimited languages such as English, Spanish or French, but becomes considerably more difficult in languages where words are not delimited by spaces like in Japanese, Chinese and Thai [34].

Stemming: heuristic process for deleting word affixes and leaving them in an invariant canonical form or "stem" [35]. For instance, *person*, *person's*, *personify* and *personification* become *person* when stemmed. The most popular English stemmer algorithm is Porter's stemmer [36].

Lemmatization: algorithmic process to bring a word into its non-inflected dictionary form. It is analogous to stemming but is achieved through a more rigorous set of steps that incorporate the morphological analysis of each word [37].

Stopword Removal: activity for removing words that are used for structuring language but do not greatly contribute to its content. Some of these words are *a*, *are*, *the*, *was* and *will*[7].

Sentence Segmentation: procedure for separating paragraphs into sentences [38]. This step presents its own challenges since periods are often used to mark the ending of a sentence but also to denote abbreviations and decimal numbers [39].

[5] `https://dev.twitter.com/rest/public/rate-limiting`, Accessed September 09, 2015

[6] `http://www.robotstxt.org/robotstxt.html`, Accessed September 09, 2015

[7] For a more complete list, visit: `http://snowball.tartarus.org/algorithms/english/stop.txt`, Accessed September 09, 2015

Part-of-Speech (POS) Tagging: is the step of labeling each word of a sentence with its part of speech, such as *adjective*, *noun*, *verb*, *adverb* and *preposition* [40–42], either to be used as input for further processing like dependency parsing [43] or to be used as features for a machine learning process [44].

Note that all of these steps are not always necessary and have to be selected to suit different Opinion Mining applications. For example, a machine-learning-based system that relies on a bag-of-words approach will probably use all of the mentioned methods in order to reduce dimensionality and noise [45], while an unsupervised approach might need some of the stopwords' parts of speech to build the dependency rules later used in the Opinion Mining core process [43], therefore omitting the stopword removal process. A more detailed analysis of supervised and unsupervised OM approaches is presented in section 2.2.2.

Moreover, there are other steps that depend heavily on the data source and acquisition method. In particular, data obtained through a Web crawler will have to be processed for removing HTML tags and non-textual information (images and ads) [12, 30, 46], and text extracted from Twitter will need special care for hashtags, mentions, retweets, poorly written text, emoticons, written laughs, and words with repeated characters [45, 47, 48].

Core Process

The third phase in the pipeline is the Opinion Mining core process. This step is the most complex of the pipeline and accordingly, it presents a vast amount of different approaches and levels of analysis which is why it will be covered in detail in section 2.2. Suffice it to say that the goal of this step is to extract the opinions of the preprocessed text, that it can be performed at the document, sentence, entity or aspect level, and that the approaches to perform it can be grouped in three categories, unsupervised lexical approaches, supervised machine-learning approaches and ontology-based approaches.

Aggregation and Summarization

The summarization step plays an important role in the Opinion Mining process since it allows to display the results of the core process in a more understandable way, nevertheless it still requires polishing and has not been the focus of the Opinion Mining community.

There are three dimensions in the summarization process. The first corresponds to the size of the source that is going to be summarized [49], the second corresponds to the type of summary that is going to be created [50, 51] and the third to the level of aggregation that will constitute the summary [19, 51].

Size of the Source

Single-Document Summarization: This kind of summarization corresponds to summarizing opinions at the document level, meaning that a summary would be produced for each input document [52].

Multiple-Document Summarization: Corresponds to the process that would produce a single summary for a collection or corpus of documents [50].

Summary Type

Extractive Summary: This type of summary is generated by *extracting* the segment of the document that is most representative of the overall opinion orientation [53].

Abstractive Summary: Is a summary created by *abstracting* the underlying information into a written statement. This approach has received less attention than the previous since it is considerably more complex and requires a deeper understanding of natural language [51]. A representative study tackling this issue is the paper by Ganesan et al. [54].

Fluent Summary: Is written in grammatical sentences that are coherent with one another at the syntactic and semantic level [50].

Disfluent Summary: Is not written according to grammatical rules and each of its composing segments does not necessarily relate to one another.

There are other types of summaries specified in [50] that encompass dimensions such as the specificity, genre, partiality, conventionality, audience, usage and expansiveness of the summary, but these are not widespread in the Opinion Mining community.

Level of Aggregation

Basic Sentiment Summarization: Corresponds to the most basic level of aggregation. The summary is generated by aggregating the results obtained directly from the OM core process depending on the level of analysis (document level, sentence level, entity level). This aggregation can be achieved by counting the positive versus negative opinions, or by obtaining the average sentiment polarity. This approach is the easiest to implement but provides results at a coarse level of granularity, which in turn does not greatly help in understanding the opinions as a whole.

Entity-Based Summarization: Represents a finer level of detail than the previous approach since it links the opinions to their target entities.

Aspect-Based Summarization: Builds the summary around the aspects found by the OM core process and the polarity towards them. One of the most significant studies on this matter is the work by Hu and Liu presented in [55] and further improved in [56].

Contrastive View Summarization: Is created by pairing positive and negative opinions on the same aspect or entity. For example a contrastive summary could group positive opinions on the service quality of a restaurant and oppose them to the negative ones [57].

Visualization of Results

The final optional step in the pipeline is the visualization process. Similarly to the summarization process described in the previous section, this step hasn't been the focus of the research performed by the OM community, hence there is no common ground in how to best perform the process. Some of the most relevant studies concerning this field are briefly introduced below.

In [58], the authors present *MoodViews*, a tool engineered for tracking the stream of mood-annotated text found in LiveJournal.[8] It is composed by three modules, *Moodgrapher*, built to plot the aggregate mood levels over time, *Moodteller*, to predict the mood levels by relying on NLP and Machine-Learning techniques, and *Moodsignals*, to detect words and phrases associated with specific moods.

Draper and Riesenfeld [59] exhibit an interactive tool for visualizing opinion poll results with a radial design. The authors argue that this design is optimal since it increases the accessibility of widgets and is simply delineated, meaning an element of the design is either inside the ring, on the ring or outside the ring, which reduces the number of states a user has to remember. They also test their visualization with 52 casual and 2 expert users and find that the interface is simple enough for both types of users.

Wu et al. [60] further develop the notion of a visualization tool with a radial design and present *OpinionSeer*, a system for visually analyzing a large corpus of hotel reviews. Their visualization is composed by an *opinion triangle* which is used to display the polarity of opinions as positive, negative or uncertain, and the *opinion rings* that allow to visualize the correlations between the opinions and other dimensions. The authors also carry out some case studies and find that their tool is useful for comparing opinions from different groups of users. Furthermore they state it could be applied to visualize opinions from virtually any field.

Finally, in [61] the authors present *SentiView*, a visualization tool for analyzing time-varying sentiment and the relationships between users given by the common orientation of their opinions. The biggest difference between this study and the previous ones is the way the authors display the relationship graph. The whole graph or *topic ellipse* represents one topic being commented on by users, each node represents an opinionated text from a user, its size represents the amount of words it contains and its position, the polarity of the opinion. The edges connect the different opinions emitted by the same user and their color represents the relationship between them. The authors finally test their tool with 300 users and find that most of them think the system is easy to use and useful.

2.1.3 Opinion Mining Surveys

This section presents several Opinion Mining surveys varying in scope, length and depth for the reader to broaden his knowledge beyond what is presented in this thesis.

[8] http://www.livejournal.com

The work by Pang and Lee [49] considers more than 300 publications and presents diverse applications and challenges, as well as the OM problem formulation and the different approaches for solving it. The authors also mention opinion summarization, study the economic implications of reviews and comment on a plethora of publicly available resources.

A more recent review was written by Bing Liu and covers more than 400 studies [19]. Here the author covers the OM subject more exhaustively by defining an opinion model and giving a stricter definition of Sentiment Analysis. He also addresses the different levels at which OM systems are implemented (document, sentence and aspect level), deals with sentiment lexicon generation, opinion summarization, comparative and sarcastic opinions, opinion spam detection, and the quality of reviews, among others.

In [17], Cambria et al., review the Opinion Mining task in general terms, describe its evolution, and discuss the direction the field is taking. In a similar fashion, Feldman [15] describes the task and places greater emphasis on its applications and some of the common issues faced by the research community, such as sarcasm and noisy texts.

More specific OM reviews include the work by Vinodhini and Chandrasekaran [16], in which they cover subjects such as commonly employed Sentiment Analysis data sources as well as different approaches like machine learning and unsupervised learning, or as they call it, "Semantic Orientation approach". They also explain some of the challenges faced in the field such as negation handling and mention some of the applications and tools available. They finish their work by presenting a table comparing different studies, the mining techniques used in them, their feature selection approaches, data sources utilized and performance metrics (accuracy, recall, and F-measure).

Khozyainov et al. [62] direct their study towards the difficulties often encountered in OM such as multidimensionality, indirect opinions, bad spelling and grammar, feature interinfluence in feature-based approaches, and the temporal dependency of opinions. Similarly, the study in [63] reviews the challenges encountered in developing sentiment analysis tools in the social media context, and covers additional concepts such as relevance, contextual information and volatility over time.

In [51] the authors survey the state of the art in opinion summarization in which they describe the background of Opinion Mining, define a conceptual framework for opinion summarization, and deepen their analysis in aspect-based and non-aspect-based opinion summarization. Finally they discuss how to evaluate summarization methods and mention some of the open challenges in this field.

Martínez-Cámara et al. [64], focus on the latest advancements in Sentiment Analysis as applied to Twitter data. They begin by giving an overview of this microblogging site, mentioning some of its sociological aspects as well as the importance of the word of mouth, and later discuss the research concerning polarity classification, temporal prediction of events and Opinion Mining in a political context. In a similar fashion, Marrese et al. [65] present an overview of Opinion Mining, describe some of the most popular sources for extracting opinionated data, discuss summarization and visualization techniques, and finally exhibit an example of a document-level Opinion Mining application for finding the most influential users

on Twitter.

Medagoda et al. [66] focus on recent advancements in Opinion Mining achieved in Hindi, Russian and Chinese. Guo et al. [30] define the concept of "Public Opinion Mining," compare different approaches used in each step of the OM pipeline and propose future directions for the field. In [20] the authors propose a faceted characterization of Opinion Mining composed of two main branches, namely *opinion structure* which deals with the relation between unstructured subjective text and structured conceptual elements, and *Opinion Mining tools and techniques* which are the means to achieve the OM task. They also tackle the problems of entity discovery and aspect identification, lexicon acquisition and sarcasm detection. Finally [67] covers some of the usual OM tasks and presents a table similar to the one presented in [16] but instead of using known metrics it just shows an arbitrary "performance" metric without clarifying whether if it represents accuracy, precision, recall, F-measure or some other measure.

2.2 Opinion Mining Core Process

This is the part of the process where opinions are actually extracted. The two previous steps of the Opinion Mining process, Data Acquisition and Text Preprocessing are also vital but do not address the core opinion extraction issue. Indeed, most analyses related to text mining, such as Information Retrieval and Topic Modeling, among others, share the first steps. Thus, what truly differentiates each field dwells within the core process. This step is the one that receives all of the OM research efforts and, as a consequence, the one with the most number of possible solutions and approaches.

In this section a taxonomy of current approaches for the Opinion Mining core process is presented. First, the possible levels of granularity at which the Opinion Mining process can be performed will be exhibited, and later, the types of techniques to execute the process will be presented.

2.2.1 Different Levels of Analysis

Since Opinion Mining began to rise in popularity, the sought-after level of analysis has passed through several stages. First it was performed at the document level where the objective was to find the general polarity of the whole document. Then, the interest shifted to the sentence level and finally to the entity and aspect levels. It is worth noting that the analyses that are more fine-grained can be aggregated to form the higher levels. For example an aspect-based Opinion Mining process could simply calculate the average sentiment in a given sentence to produce a sentence-level result.

Document-Level Analysis

Opinion mining at the document level attempts to classify an opinionated document into positive or negative. The applicability of this level is often limited and usually resides within the context of review analysis [19]. Formally, the objective in the document-level Opinion Mining task can be defined as a modified version of the one presented in section 2.1.1 and

corresponds to finding the tuples:

$$(-, GENERAL, s_{GENERAL}, -, -)$$

where the entity e, opinion holder h, and the time when the opinion was stated t are assumed known or ignored, and the attribute a_j of the entity e corresponds to $GENERAL$. This means that the analysis will only return the generalized polarity of the document. To give a few examples, in [46], Pang and Lee attempted to predict the polarity of movie reviews using three different machine learning techniques: Naïve Bayes, Maximum Entropy classification and Support Vector Machine (SVM). Similarly, in [68] the same authors tried to predict the rating of a movie given in a review, instead of just classifying the review into a positive or negative class.

Sentence-Level Analysis

Sentence-level Opinion Mining is analogous to the document-level analysis since a sentence can be considered as a short document. However, it presents the additional preprocessing step consisting of breaking the document into separate sentences, which in turn poses challenges similar to tokenization in languages not delimited by periods. In [69] Riloff and Wiebe used heuristics to automatically label previously unknown data and discover extraction patterns to extract subjective sentences. In [70] the authors achieved high recall and precision (80-90%) for detecting opinions in sentences by using a naïve Bayes classifier and including words, bigrams, trigrams, part-of-speech tags and polarity in the feature set.

Entity-Level and Aspect-Level Analysis

The entity and aspect levels represent the most granular level at which Opinion Mining is performed. Here, the task is not only to find the polarity of the opinion but also its target (entity, aspect or both), hence the 5-tuple definition described in section 2.1.1 fully applies. Both document-level and sentence-level analyses work well when the text being examined contains a single entity and aspect, but they falter when more are present [15]. Aspect-based Opinion Mining attempts to solve this problem by detecting every mentioned aspect in the text and associating them to an opinion.

The earliest work addressing this problem is [55] in which Hu and Liu detect product features (aspects) frequently commented on by customers, then identify the sentences containing opinions, assess their polarity and finally summarize the results. Likewise, in [71] the process to perform the aspect-based Opinion Mining task is to first identify product features, then identify the opinions regarding these features, later estimate their polarity and finally rank them based on their strength.

Marrese et al. [72] extend the opinion definition provided by Bing Liu by incorporating *entity expressions* and *aspect expressions* into the analysis. Later they follow the steps of aspect identification, sentiment prediction and summary generation and apply their methodology to the tourism domain by mining opinions from TripAdvisor reviews. They achieved high precision and recall (90%) in the sentiment polarity extraction task but were only able to extract 35% of the explicit *aspect expressions*. In [73], the authors further developed their

methodology and integrated it into a modular software that considers all of the previous steps with the addition of a visualization module.

2.2.2 Different Approaches

There are two well-established approaches to carry out the OM core process. One is the unsupervised lexicon-based approach, where the process relies on rules and heuristics obtained from linguistic knowledge [43], and the other is the supervised machine learning approach where algorithms learn underlying information from previously annotated data, allowing them to classify new, unlabeled data [46]. There is also a growing number of studies reporting the successful combination of both approaches [44, 74, 75]. Furthermore there is an emerging trend that uses ontologies to address the Opinion Mining problem. This is called concept-based Opinion Mining.

Unsupervised Lexicon-Based Approaches

These approaches attempt to determine the polarity of text by using a set of rules and heuristics obtained from language knowledge. The usual steps to carry them out are first, to mark each word and phrase with its corresponding sentiment polarity with the help of a lexicon, second, to incorporate the analysis of sentiment shifters and their scope (intensifiers and negation), and finally, to handle the adversative clauses (*but-clauses*) by understanding how they affect polarity and reflecting this in the final sentiment score [19]. Later steps could include opinion summarization and visualization.

The first study to tackle Opinion Mining in an unsupervised manner was [76], in which the author created an algorithm that first extracts bigrams abiding certain grammatical rules, then estimates their polarity using the Pointwise Mutual Information (PMI) and finally, computes the average polarity of every extracted bigram to estimate the overall polarity of a review. In [55], Hu and Liu created a list of opinion words using WordNet [77] to later predict the orientation of opinion sentences by determining the prevalent word orientation. Later, in [78], Taboada et al. incorporated the analysis of intensification words (*very*, *a little*, *quite*, *somewhat*) and negation words (*not*) to modify the sentiment polarity of the affected words. In [43], Vilares et al. further incorporated the analysis of syntactic dependencies to better assess the scope of both negation and intensification, and to deal with adversative clauses (given by the adversative conjunction: *but*).

Supervised Learning-Based Approaches

Also known as machine-learning-based approaches or statistical methods for sentiment classification, consist of algorithms that learn underlying patterns from example data, meaning data whose class or label is known for each instance, to later attempt to classify new unlabeled data [79]. Usually, the steps in a machine-learning-based approach consist of engineering the features to represent the object whose class is to be predicted, and then using its representation as input for the algorithm. Some features frequently used in Opinion Mining are: term frequency, POS tags, sentiment words and phrases, rules of opinion, sentiment shifters and syntactic dependency, among others [19, 44].

In [46] the authors were the first to implement such an approach. They compared the results of using the Naïve Bayes, Maximum Entropy classification and SVM approaches, and found that using unigrams as features (bag-of-words approach) yielded good results.

In [80], Pak and Paroubek relied on Twitter happy and sad emoticons to build a labeled training corpus. They later trained three classifier algorithms: Naïve Bayes Classifier, Conditional Random Fields (CRF) and SVM, and found that the first yielded the best results. In [81], Davidov, Tsur and Rappoport in addition to emoticons also used hashtags as labels to train a clustering algorithm, similar to k-Nearest Neighbors (kNN), to predict the class of unlabeled tweets.

In [82] the authors attempted to predict sentiment dynamics in the media by using 80 features extracted from tweets with two different machine-learning approaches, Dynamic Language Model (DynamicLM) [52] and a Constrained Symmetric Nonnegative Matrix Factorization (CSNMF) [83], achieving a 79% sentiment prediction accuracy with the latter, whereas only 60% with the former. This is caused mainly because DynamicLM performs better in long texts and tweets are limited to 140 characters.

Concept-Based Approaches

These approaches are relatively new and consist of using ontologies for supporting the OM task. An *ontology* is defined as a model that conceptualizes the knowledge of a given domain in a way that is understood by both humans and computers. Ontologies are usually presented as graphs where concepts are mapped to nodes linked by relationships. The study presented in [84] displays a good background study on ontologies, their applications and development. It also describes how the authors incorporated them into an Opinion Mining system to extract text segments containing concepts related to the movie domain to later classify them. In [85], Cambria et al. present a semantic resource for Opinion Mining based on common-sense reasoning and domain-specific ontologies, and describe the steps they took to build it. This resource is improved in [86], where it is enriched with affective information by fusing it with WordNet-Affect [87], another semantic resource, to add emotion labels such as *Anger*, *Disgust*, *Joy* and *Surprise*.

In [88], the author presents a new method to classify opinions by combining ontologies with lexical and syntactic knowledge. The work in [89] describes the steps in creating what the authors call a "Human Emotion Ontology" (HEO) which encompasses the domain of human emotions, and shows how this resource can be used to manage affective information related to data issued by online social interaction.

Discussion

One of the advantages of using unsupervised methods is in not having to rely on large amounts of data for training algorithms, nevertheless it is still necessary to obtain or create a sentiment lexicon. Unsupervised methods are also less domain-dependent than supervised methods. Indeed, classifiers trained in one domain have consistently shown worse performance in other domains [90, 91].

Finally it is worth noting that there are several other facets of Opinion Mining that are beyond the scope of this conceptual framework such as the lexicon creation problem, comparative opinions, sarcastic sentences, implicit features, cross-lingual adaptation, co-reference resolution, and topic modeling, among others. To get more information on these topics refer to the surveys [49] and [19].

2.3 Natural Language Processing

Natural Language Processing (NLP) is a field that implements techniques from Computer Science and Linguistics to study the understanding of human language by computers. In other words, the main concern of this field is to find the best methods to translate naturally spoken or written language, into machine-understandable data [92].

As the presence of computers gets increasingly ubiquitous by the widespread usage of smartphones and tablets, the need for faster and better communication between humans and machines becomes imperative. Up until recently the only way to communicate with a computer was, at first, by typing written instructions, and later, by interacting with graphical interfaces. Today, an increasing number of companies such as Google, Microsoft and Apple is adopting a new type of interface which is *spoken commands*. Indeed, each one of these businesses offers virtual assistants for smartphones and tablets;[9,10,11] which can provide valuable information such as weather forecasts, the user's agenda, unit conversions and tip calculations, and execute commands such as calling contacts, sending mails and posting social media. Natural Language Processing allows users to "communicate" with these assistants via spoken words, in a similar way that they would with another person.

Furthermore, NLP can be used to making sense of the massive amounts of written text currently being posted online by millions of users. Admittedly, such text is often written in a manner that cannot be "understood" by computers, hence it can only be consumed by other human users and no automated machine-driven analysis can be achieved until it is transformed into a computer-understandable format. This is the main reason why NLP is necessary for the task of Opinion Mining. In order to make sense of user generated text, and particularly, extract opinions from it, several NLP techniques must be used.

Usually, the NLP task is achieved through several steps of which the most relevant are, tokenization, lexical analysis, syntactic analysis and semantic analysis (refer to figure 2.2). Some of these were mentioned in section 2.1.2 as part of the preprocessing step of the Opinion Mining Pipeline, however, as it will be shown in this study, there are other techniques pertaining to NLP that are often necessary for later steps.

[9]Google - *Google Now*: `http://www.google.com/landing/now`, Accessed on April 01, 2015

[10]Microsoft - *Cortana*: `http://www.windowsphone.com/en-us/how-to/wp8/cortana/meet-cortana`, Accessed on April 01, 2015

[11]Apple - *Siri*: `https://www.apple.com/ios/siri/`, Accessed on April 01, 2015

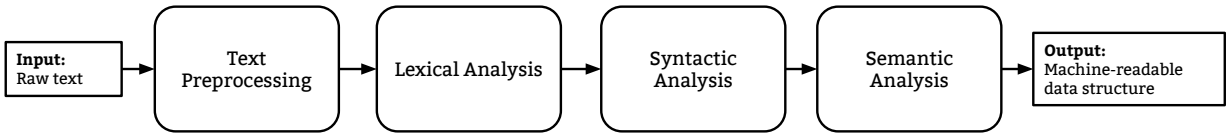


Figure 2.2: Natural Language Processing stages.

This section is structured according to the stages of analysis in Natural Language Processing, as depicted in [93]. First, the importance of text preprocessing and the challenges associated with it are explained, second, the analysis of the most basic unit of language, the word, is described, third, the analysis of sentences or syntactic analysis is discussed, fourth, the process of extracting meaning from sentences, or semantic analysis, is illustrated and finally the link between Opinion Mining and NLP is made.

2.3.1 Text Preprocessing in NLP

The definitions of text preprocessing in NLP and OM are conceptually different. The term "preprocessing" is vague and does not reflect what kind of preprocessing really takes place. Certainly, both definitions have some points in common such as tokenization, but the objectives of each are not the same. As it was previously shown in section 2.1.2, the preprocessing step in Opinion Mining deals with the tasks necessary to generate the input for the core process and, as it happens, these tasks are not exclusive to the preprocessing step of the NLP process. They include tokenization, sentence segmentation, stemming, and POS-tagging among others. In contrast, the preprocessing step in NLP deals with the tasks to generate input for the later stages of the NLP process. It could be said that text preprocessing in NLP is a subset of the text preprocessing step in OM, as shown in figure 2.3. Note that text preprocessing step of Opinion Mining includes the NLP text preprocessing step, and some of the subprocesses of lexical and syntactic analysis (depicted in dashed squares).

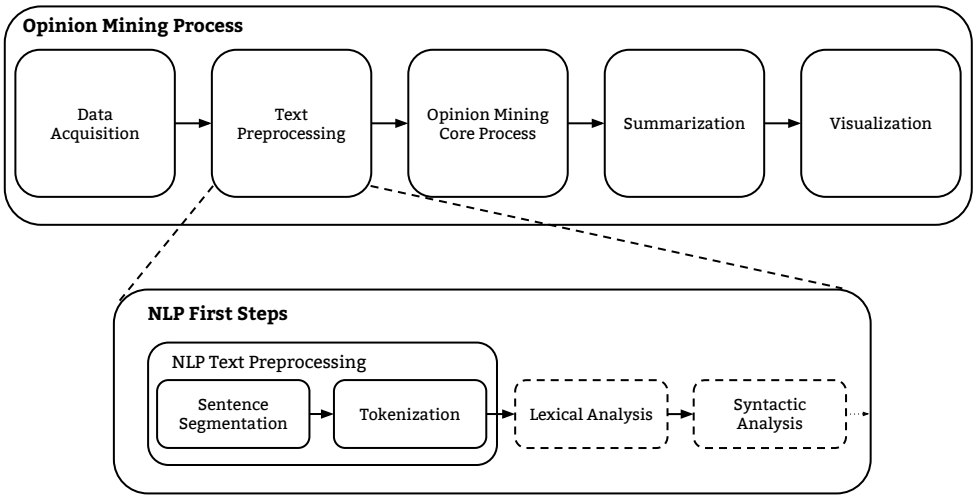


Figure 2.3: Natural Language Processing in the text preprocessing step of the Opinion Mining process.

The text preprocessing step in NLP is mainly composed of two subprocesses, sentence segmentation and tokenization (or word segmentation). Depending on the methods used for performing each subprocess, the order in which they are executed can vary. The application presented in this thesis first separates the sentences and later tokenizes the results. Furthermore, another issue previous to both of these steps is the problem of detecting the *encoding* of the documents being analyzed, however it will not be covered in this study. Roughly, the encoding of a document is the way the computer uses to translate raw text into bytes understandable by the computer [34].

Languages with few characters such as English can be easily represented with simple encodings such as ASCII which has 128 of them. In contrast, languages such as Spanish or French with special characters such as "é", "ç" and "ê" require an encoding capable of representing a larger character set. Even further, logographic languages like Japanese and Chinese require more than 2000 characters. UTF-8 is an encoding capable of representing most of the characters available in any language and is becoming the most adopted standard in the Web.[12,13]

The first step in a NLP system dealing with many documents is to ensure that every one of them is using the same encoding standard or at least devise a way to recognize and deal with each different encoding accordingly.

Sentence Segmentation

The next step in the process after having dealt with encoding issues is to break each paragraph into sentences. This is important because most NLP applications, as well as the one that will be presented in this study, consider the sentence as the unit of analysis. Indeed, as stated in [94], "a sentence expresses a proposition, an idea, or a thought, and says something about some real or imaginary world."

This task might seem simple at first glance because it is easy enough to detect periods in texts and define sentences according to them, however there are many situations in which this is not the case. In example (2.9):

(2.9) *Mr. Smith, today I was mistakenly charged with $299.99 for my rent instead of the advertised $249.99. I would like you to address this issue please.*

there are four periods, of which one represents an abbreviation, two are used for separating decimals in numbers and the last marks the ending of the sentence. As a result, the sentence segmentation problem in sentence-segmented languages is reduced to the dissambiguation of all the instances of punctuation characters that might signify the end of a sentence. The correct sentence segmentation of this example would be:

[12]Google Blog: `http://googleblog.blogspot.com/2010/01/unicode-nearing-50-of-web.html`, Accessed on April 02, 2015

[13]Web Technology Surveys: `http://w3techs.com/technologies/overview/character_encoding/all`, Accessed on April 02, 2015

(2.10) {*Mr. Smith, today I was mistakenly charged with $299.99 for my rent instead of the advertised $249.99.*}{*I would like you to address this issue please.*}

However, an algorithm that does not dissambiguate periods would segment it as follows:

(2.11) {*Mr.*}{*Smith, today I was mistakenly charged with $299.*}{*99 for my rent instead of the advertised $249.*}{*99.*}{*I would like you to address this issue please.*}

which is clearly incorrect, and does not correctly represent the information that the speaker was trying to communicate.

The approaches to solving this problem can be grouped in those that are based in rules and those that rely on trainable algorithms. The common ground for both is that the two exploit the context in which each punctuation mark is placed, in order to assess whether it represents the end of a sentence or not. Some of the context features commonly considered for sentence boundary detection are [34]:

- **Upper or lower case words:** If the corpora being studied consistently use upper-case letters for words beginning in a sentence, then case distinction provides an useful distinction for sentence boundary.
- **Part of speech:** Part-of-speech tags (or even an estimation of them) have shown to be good indicators of sentence boundary.
- **Word length:** The length of the word preceding a period might also shed light on the sentence boundary.
- **Affixes:** Prefixes and suffixes of the words surrounding a period have been used to aid in the boundary detection problem.
- **Abbreviation classes:** Grouping types of abbreviations into classes such as titles, corporate designations, and internet idioms for determining whether they might be likely to occur in a sentence boundary has also proven to improve boundary detection. To further illustrate this point take the following statement as an example:

  (2.12) *Smith & Co. stock price skyrocketed yesterday after the announcement of the acquisition of Doe Ltd.*

  In this case, the period at the end of the statement represents both a sentence boundary and an abbreviation indicator. This type of scenario shows the need for a list of abbreviations enriched at least with their class and its likelihood to appear at a sentence boundary.

- **Internal punctuation:** Punctuation that occurs within a token such as "$299.99" also provides information for boundary detection. For instance a rule could be implemented to avoid segmenting a sentence with a period that occurs between two digits.[14]

Even though there are features that are common to both rule-based and learning-based approaches for sentence segmentation in NLP, there are also different aspects for each. Below both are presented in finer detail and some of their advantages and disadvantages are discussed.

Rule-based approaches: Correspond to those solutions that rely on hand-crafted rules for a specific corpus in a specific language following specific grammatical rules. These methods are usually quick to write and quite effective. Most of them use simple grammars combined with word, exception and abbreviation lists. The advantage of these approaches is mainly the fact that simple rules for a single case-study are usually fast to define and have good overall performance. The downside is that they have to be written again for corpora that present a different writing style or another language.

Learning-based approaches: Rely on trainable machine-learning algorithms that learn underlying patterns from features extracted from the corpora in order to determine the sentence boundaries. Alternatively to rule-based approaches, these solutions require a considerably greater initial effort to create the algorithms but present the advantage that once created, only the training data must be changed in order to apply them to different corpora.

Word Segmentation or Tokenization

The other step considered in the NLP preprocessing step is word segmentation, more commonly known as tokenization. This step consists mainly on defining the components of each sentence or tokens. In space-delimited languages such as English and Spanish, tokens are usually considered as those substrings that are found between whitespaces or between a whitespace and a punctuation mark. More simply, in such languages every word should be considered as a token. Additionally, punctuation marks are also considered as tokens. For instance, a good tokenizer would segment example (2.12) as shown below:

(2.13) *[Smith] [&] [Co] [.] [stock] [price] [skyrocketed] [yesterday] [after] [the] [announcement] [of] [the] [acquisition] [of] [Doe] [Ltd] [.]*

[14] A powerful tool to accomplish these kind of string matching rules is called *regular expressions* (or *regex*). The following regular expression, implemented in *python*, would find tokens that abide the defined rule and therefore a period located in a decimal number would be considered as such and not as sentence delimiter: `\d{1,}\.\d{1,}`

This regex matches any digit of length 1 or more, followed by a period and further followed by another digit of length 1 or more. [95] presents an introduction to regex and is an overall good resource for learning the basics.

In example (2.13) punctuation marks are treated as separate tokens, which is a common practice, however, there are specific cases when they should be left as part of the token. An example of such case was presented in the previous section concerning the token "$299.99," in which the period is contained within it. Furthermore the tokenizer could expand abbreviations to generate the result:

(2.14) *[Smith] [&] [Company] [stock] [price] [skyrocketed] [yesterday] [after] [the] [announcement] [of] [the] [acquisition] [of] [Doe] [Limited] [.]*

or differentiate punctuation marks that indicate an abbreviation from those that mark a sentence boundary as in:

(2.15) *[Smith] [&] [Co.] [stock] [price] [skyrocketed] [yesterday] [after] [the] [announcement] [of] [the] [acquisition] [of] [Doe] [Ltd.] [.]*

Another aspect to bear in mind while attempting to create a tokenizer is the fact that many abbreviations might have more than one possible expansion like *St.* for instance, that could signify *Street*, *State* or *Saint*, which would require the tokenizer to consider contextual information in order to dissambiguate it.

All of these examples attempt to illustrate that tokenization is not as simple as separating words according to whitespaces, and more often than not, several considerations must be incorporated into the analysis for creating a robust tokenizer. Additionally, there are many ways to treat punctuation in the tokenization step, as depicted in examples (2.13), (2.14) and (2.15), without one necessarily outperforming the others. Admittedly, a tokenizer could use any of the approaches presented in the aforementioned examples, but later steps of the process would have to deal with the tokens while considering their definition. For example, supposing a later step consists of a part-of-speech tagging process, it would have to "know" that the token *[Co.]* is an abbreviation and attempt to expand it or to do another kind of analysis to tag it.

In most Opinion Mining studies, both sentence and word segmentation are often overlooked or even considered as obvious for the reader while in reality they pose a challenge for corpora that present exotic features, such as those created from Web-user-generated text. Moreover, languages that don't have a clear delimitation of words or sentences, such as Japanese or Thai respectively, pose even greater challenges for researchers that intend to apply NLP techniques in their analyses.

More information on the NLP preprocessing step such as how to deal with unsegmented languages can be found in [34].

2.3.2 Lexical Analysis

The next step in the NLP process after having segmented both sentences and words is the *Lexical Analysis*. The word *lexical* was originated from the Greek word $\lambda\epsilon\xi\iota\kappa\acute{o}\varsigma$ (*lexikos*)

which means "of words". Indeed, lexical analysis deals with the study of words as the building blocks of any natural language text, without considering the context in which they are placed. More specifically, this step of the NLP process attempts to further break down words into the fundamental atomic meaning-bearing units of language, called *morphemes*. A morpheme is defined as the minimal bearing unit in a language [96], and can be categorized into two classes, *stems* and *affixes*. A stem is the main morpheme constituting the word, meaning it is the one that supplies the most information and the main meaning of it, whereas affixes supply additional information. Furthermore, affixes are subdivided into four categories, *prefixes*, which precede the stem, *suffixes*, which follow it, *infixes*, which are placed inside the stem and *circumfixes*, which are placed both before and after the stem. An example to illustrate this would be the word *books* which is composed of the stem *book*, conveying the meaning of the word, and the suffix *-s*, indicating the plural form.

The main task of lexical analysis is parsing each word to decompose it into its stem and numerous affixes, which is called *morphological parsing*. The usual way to perform this task is by using a *Finite-State Automaton* (FSA) [33, 92, 96], however the theory behind them is beyond the scope of this thesis. Furthermore, there are three basic elements required to build a morphological parser: a list of stems, affixes and information about them (lexicon), a model depicting the ordering of morphemes (morphotactics), and orthographic rules defining how each stem changes when an affix is appended to it. For example to process the word *playing* the parser would have to know that *-ing* corresponds to a suffix and *play* to a stem by referring to the lexicon. Moreover, it should be aware that the suffix *-ing* placed after the stem signifies the gerund o present participle according to the morphotactics. In contrast, to process the word *ingenious* the parser would know, according to the morphotactics, that *-ing* is a suffix and has no meaning placed before the stem, so it should consider *ingenious* as a whole.

Stemming and Lemmatization

In Opinion Mining, and Information Retrieval in general, lexical analysis is used mainly to reduce a word into its stem or lemma in order to reduce complexity and dimensionality [97]. As stated in section 2.1.2, *Stemming* corresponds to the heuristic process for deleting word affixes, whereas *Lemmatization* is the algorithmic process to bring a word into its lemma or non-inflected dictionary form through morphological analysis. Usually only one of both is used since the two achieve the same goal of complexity reduction. Here, it is worth stating the difference between a *heuristic* and an *algorithm*. On the one hand a heuristic is a technique that helps in the search for answers, it is based in manually defined rules and its results are often unpredictable, however it is easier and faster to implement than an algorithm. On the other hand an algorithm is a set of well-defined instructions for carrying out a particular task, with predictable results [98].

The most popular stemming heuristic is the one created by Porter [36] for the English language. More heuristics have been created to support a considerable amount of European languages such as French, Spanish, Italian and German, among others.[15]

[15] `http://snowball.tartarus.org/`, Accessed on April 08, 2015

An example of the Porter Stemmer and Lancaster Stemmer, implemented in the *python* package *nltk* [39] and applied to a Spanish text and its translation to English is presented below:

(2.16) **Original Text in Spanish:** *Falabella tiene una pésima atención al cliente. Me gustaría conversar con el gerente a cargo para solucionar mi problema.*

(2.17) **Porter Stemmer Spanish:** *Falabella tien una pésima atención al client. Me gustaría conversar con el gerent a cargo para solucionar mi problema.*

(2.18) **Lancaster Stemmer Spanish:** *falabell tien un pésima atención al cli. me gustarí convers con el ger a cargo par solucion mi problem.*

(2.19) **Original Text in English:** *Falabella has terrible customer service. I would like to speak to the manager in charge to solve my problem.*

(2.20) **Porter Stemmer English:** *Falabella ha terribl custom servic. I would like to speak to the manag in charg to solv my problem.*

(2.21) **Lancaster Stemmer English:** *falabell has terr custom serv. i would lik to speak to the man in charg to solv my problem.*

Furthermore, the results of the lemmatization algorithm implemented in TreeTagger [99] are shown below:

(2.22) **TreeTagger Lemmatizer Spanish:** *Falabella tener un pésimo atención al cliente. yo gustar conversar con el gerente a cargo para solucionar mío problema.*

(2.23) **TreeTagger Lemmatizer English:** *Falabella have terrible customer service. I would like to speak to the manager in charge to solve my problem.*

By observing examples (2.16) through (2.23) it is possible to visualize the difference of the results obtained by stemming as opposed to lemmatizing. The stemming heuristic process chops suffixes and, in the case of the Lancaster Stemmer, it normalizes text by changing capital letters to their uncapitalized form. Conversely, the words outputted by the lemmatization algorithm are orthographically correct and represent the base dictionary form instead of the stem.

Stopwords Removal

Another step aimed towards the goal of reducing complexity and dimensionality is *stopwords removal*. *Stopwords* are words that are considered not to convey significant meaning, and this process simply deletes them. Examples of these words are:

(2.24) **Some Stopwords in Spanish:** *de, la, el, en, y, muy, pero, con, sin, ni, antes, ...*

(2.25) **Some Stopwords in English:** *i, me, my, in, it, am, are, an, if, but, very, ...*

The usual way to carry out this task is to compare each token to a list of stopwords and delete every match. In order to compile this list, the usual procedure in an Information Retrieval context is to sort every term that appears in the corpora by their frequency and put those that are the most frequent into the stopwords list [97].

All of the previously mentioned tasks, stemming, lemmatization and stopwords removal, are aimed towards reducing the complexity of human-written text to simplify the input for the later steps of the NLP process. Implementing these steps is assuming that computers are not capable of handling this level of complexity. However as computers get "smarter" and computerized systems increase their capabilities, these preprocessing steps become less required than they were before. Indeed, as Manning et al. [97] state: "The general trend in [Information Retrieval] systems has been from standard use of quite large stop lists (200–300 terms) to very small stop lists (7–12 terms) to no stop list whatsoever." In the Opinion Mining system presented in this thesis, neither the stopword removal process nor stemming and lemmatization were implemented, because stopwords and words in their inflected forms were needed to create the dependency trees (see "Dependency Grammars" on section 2.3.3).

2.3.3 Syntactic Analysis

The next step after having broken down words into their constituents, is to decompose the sentence into its components. The word *syntactic* comes from the Greek word *σύνταξις* (*suntaxis*) meaning an "arrangement" or "coordination" of elements.[16] In the case of linguistics, it refers to the arrangement and coordination of words within a sentence. Syntactic analysis is mainly concerned with grammatical rules that hold a sentence together.

There are many different words that play the same role in different contexts. For example in the sentence:

(2.26) *The dog is brown.*

dog is a noun that plays the role of the subject in the sentence, *is* is the verb, and *brown* the adjective that qualifies the subject. Any other noun could play the same role as *dog* while maintaining syntactic correctness, as in:

(2.27) *The cat is brown.*

(2.28) *The pen is brown.*

(2.29) *The moon is brown.*

where *cat*, *pen* and *moon* are the nouns that play the role of the subject in their respective sentences. Furthermore, every one of these sentences is syntactically correct, while not necessarily semantically accurate, since planet Earth's moon is not brown for the common observer, however in order to know that, another layer of complexity must be added to the analysis (see next section on semantic analysis 2.3.4).

[16] `http://www.etymonline.com/index.php?term=syntax`, Accessed on April 09, 2015

In order to construct the rules that define the correct syntax of a language, it is necessary to group words into equivalence classes called *parts of speech* (POS). The most common parts of speech, and those the reader might be most familiar with, are *nouns*, *verbs*, *adjectives*, *prepositions*, *adverbs*, *conjunctions* and *articles*. There are many compilations of the different ways these parts of speech can manifest themselves, called *tagsets*. A tagset is simply a lexicon that binds a tag with a part of speech. For example the TreeTagger tagset[17] is composed by 58 tags, some of which are "VV" which represents a verb in its base form, "NNS" which represents a noun in its plural form, and more specific ones, "RBR" representing a comparative adverb and "WP$" a possessive wh-pronoun. There are other tagsets such as the Penn Treebank,[18] the Brown Corpus tagset,[19] and the C7 tagset[20] for English; and the Eagles tagset[21] for other European languages.

Part-of-Speech Tagging

The process of labeling each word with its corresponding POS tag is called *part-of-speech tagging* (POS tagging) [96]. This process is not as simple as looking for a word's POS in a dictionary since it heavily depends on context. In the following sentences:

(2.30) *I will deal with that later.*

(2.31) *The deal was closed successfully.*

The same word *deal* has two different parts of speech. In example (2.30) *deal* is employed as a verb, whereas in example (2.31) it is used as a noun. Hence POS tagging is charged with dissambiguating these types of cases. Similar to the different types of approaches for the Opinion Mining Core Process (section 2.2.2), and for the sentence segmentation problem (section 2.3.1), three types of POS-taggers can be created to solve the POS-tagging problem, rule-based taggers, stochastic taggers (also known as learning-based or data-driven taggers) and transformation-based taggers.

Rule-Based Taggers: Rule-based tagger architecture is often composed of two stages. The first stage assigns candidate tags to words based on a dictionary, while the second stage utilizes handmade disambiguation rules to decide which of the candidate tags is the most likely to represent the truth [100].

Stochastic Taggers: These taggers consider frequency-based information extracted from the training corpora to derive underlying rules and learn how to tag unknown data. The concept of picking the most likely tag for each word in this approach is analogous to the one presented for the rule-based approach, however while in the latter the final decision is made based on handmade rules, in this approach the decision is

[17], Accessed on April 09, 2015

[18]`https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`, Accessed on April 09, 2015

[19]`http://www.scs.leeds.ac.uk/amalgam/tagsets/brown.html`, Accessed on April 09, 2015

[20]`http://ucrel.lancs.ac.uk/claws7tags.html`, - Accessed on April 09, 2015

[21]`http://nlp.lsi.upc.edu/freeling/doc/tagsets/tagset-es.html`, Accessed on April 09, 2015

made based on probabilities. The most popular stochastic methods for POS tagging are those based on *Hidden Markov Models* (HMM) [101], *Maximum Entropy* (ME) Models [102] and others that have not been used as widely as the previous ones such as SVM, Neural Networks (NN), Decision Trees, Genetic Algorithms, and Fuzzy Set Theory [42].

Transformation-Based Taggers: These taggers are based on *transformation-based learning* (TBL) defined by Eric Brill [103]. TBL combines characteristics of both rule-based tagging and stochastic tagging. Like the former, TBL relies on rules to specify which tag should be assigned to each word and, like the latter, TBL is a machine-learning technique in which rules are derived from underlying patterns in known data. The main difference from rule-based tagging is that TBL automatically infers the rules instead of a human having to craft them manually, whereas the main difference with stochastic tagging is that instead of outputting probabilities, TBL outputs rules to correctly tag words.

Roughly, transformation-based taggers operate as follows, first, tags are assigned to words randomly, based on a lexicon, or even by relying on another POS tagger, second, after every word is assigned a candidate tag, the learning phase begins. At its initial state, this phase relies on a set of predetermined rule templates which are applied sequentially to the corpus. In the first iteration, the algorithm selects the rule that reduces errors the most and adds it to the set of learnt rules. In subsequent iterations the algorithm repeats this process until none of the remaining rules reduces the error more than a predefined threshold [40]. The final output is a set of rules to "patch" incorrectly tagged words following the pattern:

(2.32) *Change tag* `a` *to tag* `b` *if condition* `z` *is satisfied.*

for example:

(2.33) *Change tag* `VB` *to* `NN` *if one of the previous two tags is* `DT`

The rule in example (2.33) can be translated to "Any word tagged as a verb in its base form should be tagged as a noun, if one of the two words preceding it is a determiner." If a tagger incorrectly tagged example (2.31) as follows:[22]

(2.34) *The deal was closed successfully.*
`DT` `VB` `VBD` `VBN` `RB`

it could be corrected by applying rule (2.33) resulting in:

(2.35) *The deal was closed successfully.*
`DT` `NN` `VBD` `VBN` `RB`

Note how the `VB` tag under the word *deal* was corrected to `NN` since it was preceded by the word *the*, tagged as a determiner `DT`.

[22]The tags correspond to the Penn Treebank tagset.

Some of the advantages of using TBL taggers are that they are flexible in terms of the features that can be incorporated into the model, since rule templates can be easily edited, they are less prone to overfitting and their output is easier to interpret than those of the stochastic methods [42]. Furthermore, there is a trade-off between using purely rule-based methods and using stochastic or TBL methods, which is the effort implied in crafting the rules or creating the training corpora. However, as it was previously mentioned, learning-based methods are often easier to apply to other domains than the rule-base methods.

The part-of-speech tagging process is just an intermediate step in understanding the structure of a sentence, since it only represents words as atomic units and does not reflect the relationship between them. This is the reason why another step must be implemented before transforming a sentence into a data-structure that can be further processed by a computer.

Grammars and Parse Trees

Words are the most basic unit of meaning in language, however to convey meaning through a sentence they must be grouped with other words. A *phrase* or *constituent* is a group of words that act as a unit [96]. A *noun phrase* for instance, is defined as a phrase containing at least one noun [92]. Some examples of noun phrases are:

(2.36) *the man*
DT NN

(2.37) *the book*
DT NN

(2.38) *a dog*
DT NN

Note that these examples are all noun phrases, even if they are composed by different words.

The various rules that define the ways words (or symbols) of a given language can be combined to form phrases, are grouped in what is called a *context-free grammar* (CFG). A simple CFG rule able to define the three previous examples could be the following one:

(2.39) NP → `DT NN`

Example (2.39) signifies that any succession of words which is composed by a determiner (`DT`) followed by a noun (`NN`) will be defined as (or is *derived* from) a noun phrase (NP). Furthermore, it should be noted that CFGs are represented by two types of elements, *terminal* symbols and *nonterminal* symbols. In the case of natural language grammars, terminal symbols correspond to the words that would be found in a lexicon and nonterminal symbols represent clusters or generalizations. Context-free rules, such as the one presented in example (2.39), are defined by a single nonterminal symbol (NP) to the left of the arrow (→) and an ordered list of one or more terminal and nonterminal symbols (`DT NN`) to the right of the arrow.

A context-free rule is formally called a *derivation*, since it conceptually represents the fact that the items to the right of the arrow can be derived from those on the left. This definition sheds light on why terminal and nonterminal symbols are called like that. A terminal symbol cannot be further derived whereas a nonterminal one, obviously, can.

In addition, derivations can be represented as *parse trees*. For example, rule (2.39) could be represented as shown in figure 2.4:

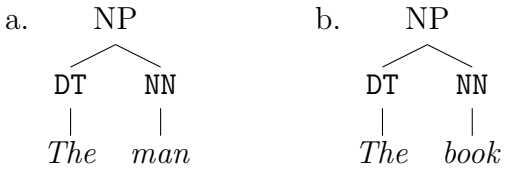


Figure 2.4: Parse trees representing two noun phrases.

However, to build a sentence that transmits an idea, another context-free rule must be first defined. For the sake of simplicity, a way to define a *verb phrase* (VP) is as a structure composed by a verb, either in its base form or past tense, followed by a noun phrase (in the English language there are many more ways to define a verb phrase but here two are shown in order to exemplify in a simple way).

(2.40) VP → `VV` NP

(2.41) VP → `VVD` NP

By deriving the "NP" component of rule (2.41), according to (2.39), the following result rule is obtained,

(2.42) VP → `VVD DT NN`

Meaning that the following example can be considered as a verb phrase,

(2.43) *took the book*
`VVD DT NN`

but still example (2.43) only represents a phrase, not a sentence, meaning it doesn't convey any meaning on its own. The simplest way to define a sentence is through the following rule,

(2.44) S → NP VP

meaning that a sentence must be at least composed of a noun phrase followed by a verb phrase. By deriving a sentence, according to rules (2.39) and (2.42), the following rule is obtained,

(2.45) $\mathrm{S} \rightarrow \texttt{DT NN VVD DT NN}$

hence, a grammatically correct sentence derived from rule (2.45) could be,

(2.46) *The man took the book*
DT NN VVD DT NN

and its representation, along with its translation to Spanish, as a parse tree is presented in figure 2.5.

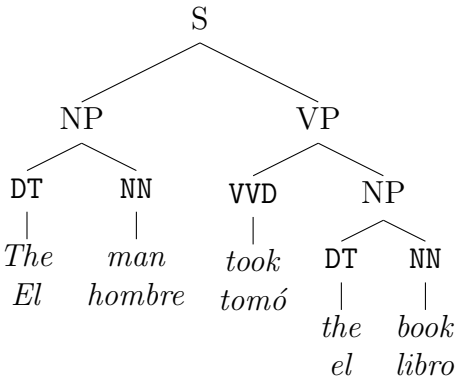


Figure 2.5: Parse tree of a sentence in English and Spanish.
Source: This was the first representation of a parse tree, presented in [104].

It is worth noting, as exemplified by figure 2.5, that most, if not all, of the concepts explained up until now, are perfectly applicable to similar languages such as Spanish.

Finally, what has been presented until now only corresponds to the representation of a context-free grammar. Another type of grammar that is of particular interest to this work is called *dependency grammar* and will be explained in short. Furthermore, in [105] there are examples of other types of grammar, like *head grammars*, *lexicalized grammars* and *type-theoretical grammars* but they are beyond the scope of this work.

Dependency Grammars

An alternative to representing language through context-free grammars, is to do so by using dependency grammars. In these type of grammars, the syntactic structure of a sentence is given purely by the binary relationships between words. This type of grammar does not group words in higher-order structures such as phrases, but instead it shows how any given word relates to another word in the same sentence.

Some of the most common dependency relations are, for example, the *nominal subject* of a sentence which represents the noun that executes the action, *direct object* which represents the object upon which the action is being executed, and *adjectival modifier* which is the relationship describing an adjective modifying a noun. A full description on dependency relationships can be found on [106].

A single dependency relationship can be represented as a triplet (rel_i, w_j, w_k) where rel_i represents the dependency relationship between the *head word* w_j and the *modifier* w_k [44]. For example, in the phrase *the green tree*, the dependency relationship between the words *green* and *tree* could be represented as $(amod, tree, green)$ meaning that *green* is an adjectival modifier of *tree*.

Just like with parts of speech, dependency relations are usually represented by labels or tags for which there is no global standard. As it was stated before, there are many tagsets for assigning tags to parts of speech, and the same occurs to dependency relationships. In this thesis, both the Stanford typed dependencies [106] and the AnCora-ES typed dependencies [107] will be used for illustration purposes. Furthermore, the latter will also be used for the rest of the development since the corpus used for training the dependency parser uses them.[23] For more information on the AnCora-ES types dependencies refer to Appendices B, C, and D.

Moreover, a dependency structure for any given sentence is represented by a directed acyclic graph, where nodes are words and edges are the dependency relationships [105]. Nivre et al. [108] define the conditions these graphs must fulfill in order to be *well-formed*:

Single Root Node: They possess a single root node.

Connectedness: They are *weakly connected*, which means that there is a path between every pair of nodes.

Single Head: Every node posses at most one head or parent node.

Aciclicity: If the triplet (rel_i, w_a, w_b) exists, then no other triplet (rel_j, w_b, w_a) $\forall j$ may exist. This means that if word w_a is modified by word w_b by the dependency relationship rel_i, then it is not possible for word w_a to modify word w_b by any dependency relationship rel_j.

Projectivity: if there is a relationship between words w_a and w_b then there is a path of arcs that connects w_a with the words between w_a and w_b.

[23] The dependency parser and its training corpus are briefly described in 4.1.4

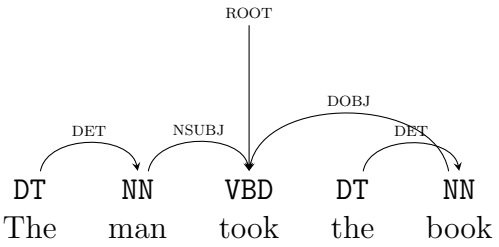


Figure 2.6: Dependency structure for an English sentence with Stanford-typed dependencies.

Figure 2.6 shows the dependency structure of the example sentence (2.46). It is possible to see that the word *The* is the determiner (DET) of *man* which is the nominal subject (NSUBJ) of the sentence, represented by the root node *took*. Furthermore, the word *book* is the direct object (DOBJ) of the verb and is determined by the second word *the* in the sentence. It is also worth mentioning that each node points to its parent, so the first node *The* points to its parent *man* which subsequently points to its parent *took*.

Dependency graphs can also be represented as trees, given their previously mentioned properties. Figure 2.7 represents the same information as figure 2.6, where each node corresponds to a word with its subscript being the type of relationship it holds with its parent. It is important to know both kinds of representation since the two are used in literature.

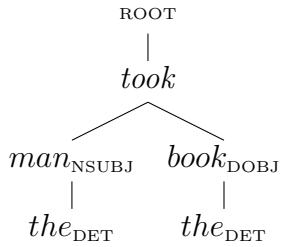


Figure 2.7: Dependency structure for an English sentence represented as a tree.

To further illustrate syntactic dependencies, figures 2.8 and 2.9 show the dependencies of a slightly more complex English sentence containing adjectives, represented by the Penn Treebank POS tag JJ.

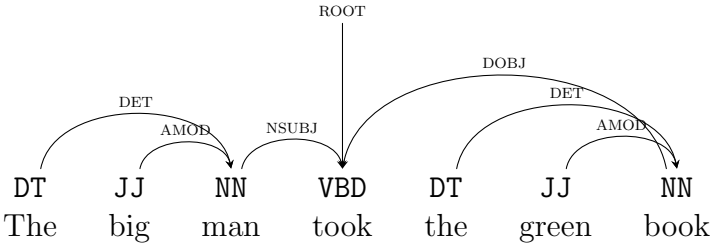


Figure 2.8: Dependency structure for an English sentence with adjectives.

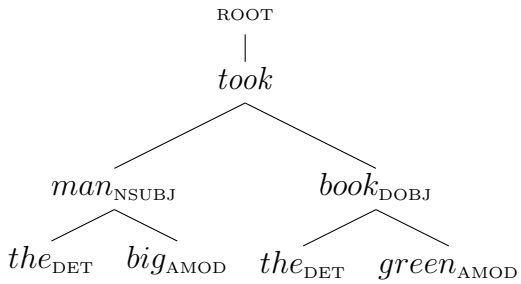


Figure 2.9: Dependency structure for an English sentence with adjectives represented as a tree.

With the dependency structure it is possible to know how words are related, and particularly, which are the targets of adjectives and adverbs. In the case of figures 2.8 and 2.9, it is easy to see that the word *man* is modified by the adjective *big* and the word *book* by the adjective *green*. This kind of analysis will later allow to create rules for specific syntactic constructs, such as negation and intensification, in order to quantify their impact in the polarity of an opinion. For more information on these rules, refer to Section 3.6.2 and Section 4.4.2.

Finally, all of these concepts are also applicable to Spanish with some slight variations such as the dependency notation. Figure 2.10 shows the syntactic structure of sentence (2.46) translated to Spanish and using AnCora-ES typed dependencies [107].

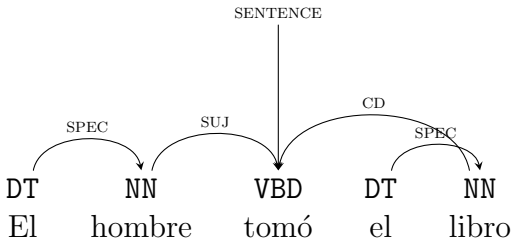


Figure 2.10: Dependency structure for a Spanish sentence with AnCora-ES typed dependencies.

2.3.4 Semantic Analysis

This kind of analysis is not used in the study developed in this thesis, but for the sake of consistency it will be briefly presented.

The last step in the NLP process is *Semantic Analysis*, charged with the task of assigning meaning to the constructs obtained in the stage of syntactic analysis. The word semantic comes from the Greek word *σημαντικός* (*semantikós*) which means *significant*.[24] The reason for this name is that semantic analysis studies what linguistic symbols or signs *signify* in the real world.

To be able to map linguistic utterances to an appropriate meaning, a *meaning representation* is required for linking linguistic inputs to non-linguistic knowledge [92, 96, 109]. These

[24] `http://www.etymonline.com/index.php?term=semantic`, Accessed on April 16, 2015

representations consist of structure composed by a set of symbols which, when arranged in some order, represent objects of the real world and the relationships between them. One of the most widely accepted meaning representation is called *First-Order Predicate Calculus* (FOPC). FOPC is composed by various atomic elements that are combined to represent a state of the world or truth. These elements are *terms*, denoting objects or entities, *logical connectives* ($\wedge$, $\vee$, $\Rightarrow$, ...), representing the relationships between terms and *quantifiers* ($\exists$, $\forall$, ...).

The example presented in [109],

(2.47) *Some politicians are mortal.*

represents a linguistic input that a human can easily understand but in order to make its meaning clear to a machine it must be represented through a meaning representation such as FOPC. The following example depicts such representation:

(2.48) $\exists x$ *(politician(x)* $\wedge$ *mortal(x))*

Where $\exists x$ means that "x exists" or that "there is at least an x", *politician(x)* means that variable x *is a* politician and *mortal(x)* that x *is* mortal. Furthermore (*politician(x)* $\wedge$ *mortal(x)*) means that both conditions are fulfilled by x, denoted by the symbol $\wedge$. In other words, example (2.48) can be reformulated as:

(2.49) *An x exists such that x is a politician and x is mortal.*

Another example of a statement, along with its logical notation and rephrasing, is presented in [96] and replicated below:

(2.50) *All vegetarian restaurants serve vegetarian food.*

(2.51) $\forall x$ *VegetarianRestaurant(x)* $\Rightarrow$ *Serves(x, VegetarianFood)*

(2.52) *For all x such as x is a VegetarianRestaurant, x serves VegetarianFood*

Where $\forall x$ means "every x" or "for all x", the symbol $\Rightarrow$ represents implication and *Serves(x, VegetarianFood)* is a two-argument function that denotes that variable x serves the object *VegetarianFood*.

In summary, the main task of semantic analysis is mapping the output produced by the syntactic analysis, meaning the sentence representation according to a certain grammar, to its semantic representation in FOPC or any meaning representation. To learn more on semantic analysis refer to [92], [96] and [109].

2.3.5 Natural Language Processing in Opinion Mining

This final section is intended to summarize what has been said about Natural Language Processing and link the field to its applications in Opinion Mining. Figure 2.3 depicts some stages of NLP as the text preprocessing step of the Opinion Mining. Indeed, since every Opinion Mining application must have a text preprocessing step, it can be said, quite accurately, that every one of them uses Natural Language Processing to some extent.

The NLP techniques that are used in the early steps of the OM process are common to all approaches and include tokenization, sentence segmenting (for analyses more fine-grained than document-level opinion mining), and stemming or lemmatization. In other words, most of the NLP text preprocessing techniques, and some of the lexical analysis, are used in every OM application. However, the closer to the OM core process and the more NLP techniques diverge. For example, a purely aspect-based Opinion Mining system relying on a machine-learning approach might only need to parse the sentence to extract its structure for obtaining linguistic features that could later be used to find named entities [110], without the need for dependency relationships, whereas other approaches may exploit dependencies for other uses, such as the one presented in this thesis (See section 3.6.2), and in [43].

Concerning lexical analysis, Opinion Mining only uses the already-created tools for stemming and lemmatizing, and is not concerned with a finer level of detail for lexical features such as the ones given by morphological analysis. Furthermore, the application of syntactic analysis varies from application to application. Accordingly, most OM studies use POS tagging as a means for obtaining every token's POS and using it either as a feature for a machine learning approach or for syntactic parsing. Moreover, OM applications do not frequently use syntactic parsing, but use a simpler process called *chunking*. Chunking can be defined as the step previous to parsing, responsible for the task of segmenting sentences into phrases or *chunks* [111]. The advantage of using chunks, as opposed to complete parsed sentences, is that the chunked structure is simpler and easier to work with [112]. Finally, semantic analysis, as presented in [92], [96], and [109], has yet to gain popularity in the Opinion Mining research field. It is only recently that approaches such as concept-based Opinion Mining have begun incorporating knowledge representations of the real world such as ontologies (see subsection Concept-Based Approaches of section 2.2.2).

Additionally, the border between the later steps of NLP used in the context of an OM application is blurry. For example one could consider the extraction of linguistic features as part of the machine learning process pertaining to the OM core task, or as a part of the syntactic analysis stage of the NLP process. In the end it should be considered as an overlap of both.

For the most part, all of the NLP techniques used in the text preprocessing step of the OM process have already been implemented and are free to use by researchers. For instance, the *NLTK* Python package [39] provides numerous tools for tokenization, sentence segmentation, and many others; the TreeTagger [99] provides functionality for POS tagging in several languages; the Stanford Parser [113] allows the user to parse sentences in English, French, German, Chinese and Arabic; and MaltParser [114] offers the tools necessary to perform dependency parsing, provided the user possess a tagged corpus in the target language to

train it.

Finally, it is worth mentioning that NLP has two facets: language understanding and language generation. In this section only the former was considered. To learn more on language generation and Natural Language Processing in general, refer to [93].

2.4 Twitter

In this section the microblogging platform Twitter is presented. First, a brief overview will be given, second, the relationship between Brands and Twitter will be analyzed, and third, the latest Opinion Mining techniques applied to Twitter will be described.

2.4.1 Overview

Origins

Twitter is a microblogging platform launched on July 13, 2006, which has since then, seen an unprecedented user growth. Indeed, its traffic increased from 200 million tweets a day in June 2011[25] to 500 million in June 2013;[26] and is today the 8th most popular website in the world?[27] Furthermore, on February 2013 the site accounted for 200 million active users.[28]

Description

Microblogging is defined as the activity of making short frequent posts to a microblog,[29] which in time is a blog with restrictions on the amount of characters each user can use in each post.[30] Accordingly, Twitter allows each user to post short messages containing no more than 140 characters, called *tweets*, as frequently as they desire. Moreover, the content of each tweet varies depending on each user and range from personal information to news information [64].

There are three different types of tweets: *status updates*, which correspond to messages the users post in their profile, *retweets*, which are tweets "forwarded" to the followers of the user that retweeted, and *replies*, which are answers to tweets mentioning the replying user. Furthermore, there are different symbols that represent different elements inside each tweet. Some of these elements are, the tag "RT" at the beginning of the message which denotes that the message corresponds to a retweet, the symbol @ used when a user *mentions* another user, and the *hashtag* symbol # used to indicate that the tweet is relevant to a certain topic. Hashtags that are being used frequently are called a *trending topics* (TT). Finally, tweets

[25] `https://blog.twitter.com/2011/200-million-tweets-day`, Accessed on April 17, 2015

[26] `https://blog.twitter.com/2013/new-tweets-per-second-record-and-how`, Accessed on April 17, 2015

[27] `http://www.alexa.com/siteinfo/twitter.com`, Accessed on April 17, 2015

[28] `https://blog.twitter.com/2013/new-compete-study-primary-mobile-users-on-twitter`, Accessed on April 17, 2015

[29] `http://www.merriam-webster.com/dictionary/microblogging`, Accessed on April 17, 2015

[30] `http://www.merriam-webster.com/dictionary/blog`, Accessed on April 17, 2015

may also contain URLs linking to external content. Below, two examples depicting tweets in English and Spanish are presented:

(2.53) *Will Obama's Immigration Actions Remain on Hold? Appeals Court to Decide. http://dailysign.al/1OIVYoc via @SiegelScribe @DailySignal*

(2.54) *Estoy participando por un viaje a las Cataratas con Falabella y LAN. ¡Sumate! #ViajeDeTusSueños via @captia http://bit.ly/1ERzmPA*

Example (2.53) is a tweet that references an article written by user @SiegelScribe who works at the Daily Signal (@DailySignal). Furthermore, example (2.54) represents a tweet related to a contest for winning a trip to the Iguazu Falls, referenced by the hashtag *#ViajeDeTusSueños*, and was probably generated automatically by an application.

With all these features, Twitter has proven to be a good medium to disseminate all kinds of information such as daily life activities, news, opinions, seek knowledge and expertise [115], and even serve as a tool for communicating in times of large-scale emergencies (See hashtag #TerremotoChile for tweets related to the earthquake that took place on February the 27$^{\text{th}}$ 2010 in Chile)

User Characterization

Each Twitter user may *follow* other users, which means that their updates will appear in the user's *timeline*. The default setting for any new Twitter account is to accept every follow request and to allow tweets to be seen publicly, however every user has the option change these settings in order to decide whether to accept a follow request or not, and to restrict tweets to be seen only by those users that follow him.

The study by Krishnamurthy et al. [26] divides Twitter users in three categories. First, the *broadcasters* are users that have a much larger number of followers than the number of users they themselves follow. Usually these users correspond to online radios, newspapers or newswire in general. Second, *acquaintances* are users that follow a number of users similar to the number of followers they have, and their relationship is often reciprocal. Third, accounts that have few or no followers, and follow a high number of users, are usually associated with *spamming* activities.

In addition, the same study found that users with many followers also tweeted frequently, confirming their status as broadcasters. In fact, they found that users with more than 250 followers tweeted more frequently than those users that follow 250 accounts or more, meaning that the follower count of an user is a good indicator of his activity status.

Moreover, a study published by the Pew Research Center [116], reveals that, as of September 2014, 23% of adults that are internet users and live in the continental territory of the United States, use Twitter. Furthermore 21% of female and 24% of male respondents stated they use the microblogging platform. Finally, 37% of internet users between 18 and 29 years of age, 25% between 30 and 49, 12% between 50 and 64, and 10% of 65 or more use Twitter.

2.4.2 Companies and Twitter

Twitter's success is not only based on its massive amount of active users or the huge flow of data it produces, but also by the increasing interest presented both by the business and political worlds [64]. This interest is mainly driven by the fact that Twitter has given birth to a new type of electronic word-of-mouth marketing [117]. The goal of this subsection is first, to define what word-of mouth is, second, transmit the importance it has in a business context, and third, reflect how Twitter is playing a fundamental role in driving electronic word of mouth.

Electronic Word of Mouth

Word of Mouth (WoM[31]), is defined as "oral, person to person communication between a receiver and a communicator whom the receiver perceives as non-commercial, concerning a brand, a product or a service" [118]. Moreover, WoM communication is based upon social networking and trust, since communicators and receivers often rely on family members, friends or acquaintances, however, research also indicates that receivers tend to trust disinterested opinions from people outside their inner circle, such as online reviews [117].

Electronic Word of Mouth (eWoM) is defined as "any statement based on positive, neutral or negative experiences made by potential, actual or former consumers about a product, service, brand or company, which is made available to a multitude of people and institutions via the Internet" [118].

Consequences of Electronic Word of Mouth

It is widely accepted that WoM is a powerful tool for driving customer behavior and influencing purchase decisions [64], and further, WoM marketing has been found to be more effective than conventional advertising media [118]. Furthermore, even if eWoM is less personal than traditional face-to-face WoM, it is considerably more powerful since it is immediate, has significant reach and is publicly accessible by others [119].

The study by Dellarocas [120], states that eWoM is affecting a wide range of activities within organizations. Some examples of these activities are first, *brand building and customer acquisition*, since online feedback mechanisms can help in acquiring and retaining customers as an addition to regular advertising but can also quickly spread negative feedback and harm brand equity, and second, *product development and quality control*, because these feedback mechanisms can help an organization better understand consumer reactions to its products or services. The study further notes that eWoM is different to traditional WoM in that it presents an unprecedented scale given by the bidirectional communication channel facilitated by the Internet, it allows companies to control and monitor eWoM through automated feedback mediators (such as the application presented in this thesis), and it poses new challenges that are characteristic to online interaction such as the lack of context and the volatile nature of online identities.

[31]Not to be confused with Web Opinion Mining (WOM).

Up until now, the benefits of eWoM are pretty clear since by correctly using them, a firm can influence purchasing behavior and obtain insightful information to improve its business, however, eWoM is a double-edged sword. If left unattended, electronic word of mouth can grow to be a liability and deteriorate brand image. Park and Lee [121], demonstrated that the effect of negative eWoM is greater than the effect of positive eWoM, which is why businesses must not only drive potential consumers toward consumption and influence what is being said about them, but also must avoid and contain cases of negative customer experiences. Apart from this, consumers not only know that companies exploit eWoM-related phenomena, but *expect* them to do so and to have online presence in a variety of platforms. This, combined with the growing amount of available online channels for consumers to express themselves, pose significant marketing challenges to every company interested in having online influence [118].

Twitter as a Channel for Electronic Word of Mouth

One of the many channels to transmit eWoM are microblogging platforms and, specifically, Twitter. Being the 8th most popular website in the world, preceded only by search engines such as Google and Yahoo, and sites like Facebook, Youtube, Wikipedia and Amazon, there is no point in discussing that Twitter is indeed the most popular microblogging site in the world. It could be argued that some activities allowed by Facebook should be considered as microblogging but it is clear that this is not the main and only purpose of the platform, as opposed to Twitter, hence this subsection will focus in eWoM transmitted purely through it.

From the previous subsection, it is straightforward that electronic word of mouth, in its many forms, plays an essential role for most businesses. This section will attempt to shed light on what should be expected while using Twitter as the platform to drive eWoM.

The study by Dellarocas [120], mentioned earlier, dates from 2003, three years before the advent of Twitter, but already touched subjects that are relevant even now. Most of it is focused on what the author calls *online feedback mechanisms*, which in the time of writing was considered to be feedback left by customers in E-commerce sites such as eBay, however the concepts described by the author and the insights he found are perfectly applicable to Twitter, provided it is considered as an online feedback mechanism.

Studies have shown that roughly 40% of what is posted on Twitter is "pointless babble" or content that does not convey anything meaningful, followed by 37% being conversational content, 9% content with pass-along value, 6% corresponding to self-promotion, 4% to spam and 4% to news [122]. These values should not be considered as definitive but just as an approximation since the study was based only on 2000 tweets, however they confirm the common belief that the entirety of the data found in Twitter is not exploitable whole. At any rate, even if there is a considerable amount of unusable data in Twitter, there is also an important quantity of exploitable data. Furthermore, studies such as the one by Park and Lee [121] previously mentioned, and the one by Campbell et al. [123], have demonstrated that online WoM-related activities have a measurable impact on a firm's business.

Some characteristics pertaining to Twitter that directly impacts eWoM communication are that users are able to share brand-affecting opinions without location or reach limitations,

meaning that the user can create a post wherever he wishes and virtually anyone can access them, all of this in an unprecedented scale. Furthermore, according to [117], tweets are:

- *Asynchronous*: They can be accessed independently of the time.
- *Noninvasive*: A user can choose which users to receive updates from.
- *Indexable*: They are searchable through Web search engines and services like Topsy.[32]
- *Immediate*: As soon as a user posts a tweet, it can be accessed by anyone user immediately.
- *Ubiquitous* : Every tweet is accessible anywhere in the world, by any follower of the user that posted it (or by anyone in case tweets are public).

Besides, given the time-independent nature of microblogging, a tweet can be posted online very near or during the purchase decision [124], which could deeply affect the success of advertisers, businesses and products. This further reinforces the fact that firms must devote at least some effort in understanding consumer behavior through online channels and researching how to legitimately exploit eWoM to both their and the consumer's benefit.

A recent study by Twitter found that shoppers rely on the platform for information and advice.[33] In addition, the study found that Twitter users have bigger budgets and buy more often than non-users, also, they are 160% more likely to stay up to date on brand news and promotions and 120% more likely to search for deals. On top of that, they found that users tweet at every stage of purchase (Awareness, Interest/Consideration, Evaluation, Purchase intent, Conversion, Post-purchase chatter and Advocacy/loyalty), for every retail category (Big box retail, Consumer electronics, Apparel, Home improvement, Grocery/Pharmacy), and what's more, every retail category presented different tweet distributions for each stage. For example 51% of tweets concerning apparel shopping correspond to awareness followed by 15% corresponding to conversion, whereas 49% of those concerning consumer electronics correspond to post-purchase chatter and 43% to awareness. Finally, the study found what is the sales driver in each retail category, specifically, it found that top sales driver for big box retail is customer service, and the one for consumer electronics is advertisements.

Apart from this, Jansen et al. [117] further elaborated on the findings presented in the study by Esch et al. [125], to show how microblogging influences *brand image* and *brand awareness*. They propose the model summarized in figure 2.11. The model considers that current purchases are directly affected by brand image and indirectly by brand awareness. Further, both components are the most influenced by eWoM microblogging, which requires firms, and particularly, brand managers to take an active role in the microblogging context. This way they can better manage brand satisfaction, brand trust and brand attachment, and ultimately drive the consumers' behavioral outcomes.

[32] `http://topsy.com/`, Accessed on April 21, 2015

[33] `https://blog.twitter.com/2015/new-shopper-behavior-research-twitter-s-role-in-the-retail-path-to-purchase`, Accessed on April 21, 2015

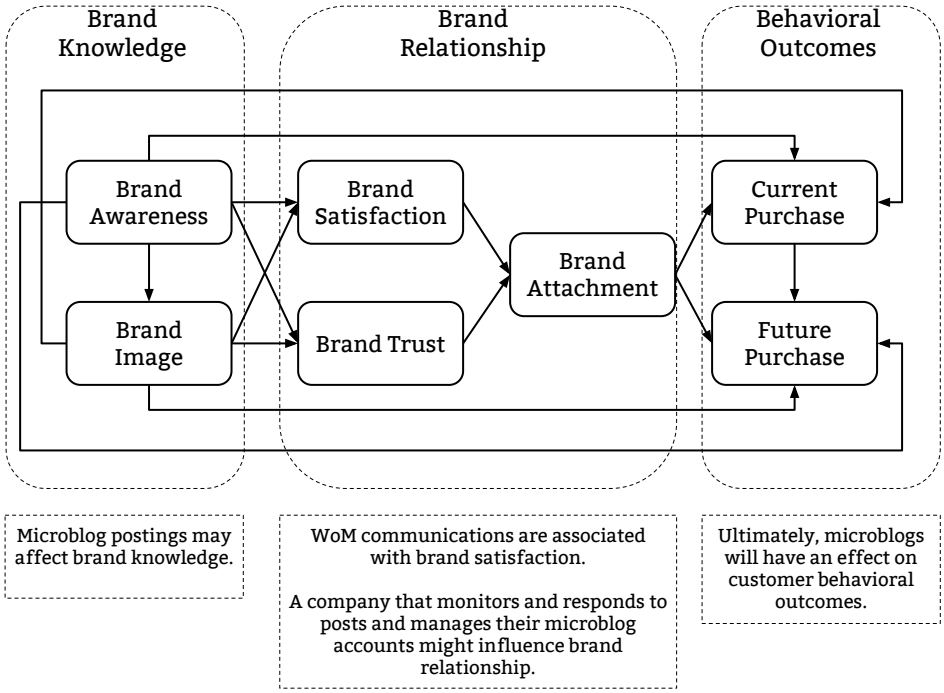


Figure 2.11: How microblogging affects branding components.
Source: [117]

Additionally, Jansen et al. report that 19% of tweets, pertaining to a random sample of 14,200 tweets, mentions some brand or product, which implies that microblogging could be a rich area for companies interested in brand and customer relationship management. Besides, they further divided this 19% in 4 categories, those tweets containing an opinion, those related to information seeking and information providing, and those where the brand is just commented and is not the primary focus of the tweet. They found that the majority of tweets (48.5%) corresponds to *comments*, 22.3% contained opinions, 18.1% were related to information providing and 11.1% to information seeking. This shows that even if most tweets do not mention any brand, there is still a considerable amount that does. Moreover, for those that do, more than half contain either opinions or information-seeking related content. This further supports the idea that microblogs could be exploited by firms to practice brand management and to serve as a channel of communication to provide feedback and disseminate information concerning products or services. However, given the growing amount of users and content, eventually the ability to manage these social networks will become humanly impossible, if not already so, making automated systems ever so necessary.

To summarize, Twitter has proven to be a new channel for transmitting electronic word of mouth. It is clear that most data in Twitter are not exploitable, since they lack meaningful content or, in other words, are pointless babble but, despite this, there is still a considerable amount of information that could and should be exploited. The volume of user-generated content the platform generates and its growth, are phenomena being observed for the first time in human history and, for what has been briefly presented in this section, firms that do not put them to good use will indisputably fall behind those that do.

2.4.3 Opinion Mining in Twitter

Now that both Opinion Mining and Twitter have been explained, this section attempts to describe the role the former plays in extracting useful information from the latter. First, some characteristics inherent to microblogs will be presented and then the latest techniques and findings in the Twitter context will be discussed.

Twitter Characteristics

The main difference between posts of a common blog and those of a microblog is the limitation for the amount of characters it may contain. This and other characteristics make the analysis of Twitter data more complex, as opposed to the analysis of data coming from review sites for instance, which raises the need to pay particular attention to them while attempting to exploit them by means of Opinion Mining techniques. Below, some of these characteristics are presented according to [64] and [126]:

- **Sparsity:** Twitter posts or tweets are limited to 140 characters which makes users refer to different concepts with a wide range of different expressions, difficulting the polarity classification and topic detection tasks.
- **Jargon:** Given the character limitation, the use of abbreviations, jargon and idioms is very common. For example, in Spanish the expressions *por qué* (why), and *porque* (because), are usually replaced by *xq* or *pq*. Similarly, in English expressions like "in my opinion" or "for your information" are replaced by "imo" and "fyi" respectively. Furthermore, there are other words in English that are contracted depending on their sound. For instance, words that contain a sound similar to "eight" are often transformed to a mix of letters and numbers that produce the same sound than the original word when read, but have no lexical meaning. Some instances of these words are "waiting," "great," and "skate" which are often replaced by "w8ing," "gr8" and "sk8" respectively.
- **Poor grammar and orthography:** Users usually do not care for the lexical and syntactical correctness of their tweets as long as they are understandable by other humans. However, the process of understanding exotic orthographic and grammatical constructs is done subconsciously, and scientists are not very familiar with the way the human brain does it. Indeed, to understand this, and how the brain processes natural language in general, would imply a considerable scientific breakthrough. In short, the low quality of language usually found in tweets makes the OM process more difficult.
- **Lack of context:** Each tweet, as an atomic unit, has little or no context. This raises the need to find ways to contextualize each tweet in order to facilitate the Opinion Mining process. The simplest context indicators are hashtags, which denote topics specified by the author (see section 2.4.1), however, to analyze them means to incur in more overhead. Another way to contextualize tweets could be to automatically detect their topics by means of topic models such as the Latent Dirichlet Allocation (LDA) [127] which implies an even greater effort.
- **Multilingualism:** Tweets are not restricted to English, in fact, Twitter has a considerable reach in Spanish-speaking countries such as Chile (see section 1.1.2). Processing

more than one language poses many challenges and further increases the Natural Language Processing task's complexity (see section 2.3). Studies analyzing Twitter data are usually limited only to one language.

- **Miscellaneous Twitter Features:** Finally, Twitter contains specific features that have to be considered in any Opinion Mining process. These are the ones mentioned in section 2.4.1: hashtags (#), mentions (@), retweets (RT), trending topics (TT) and URLs. In addition to these features, other elements that are frequently used are called *emoticons*. Emoticons are simply a written representation of a facial expression and, just like facial expressions, they are intended to transmit a certain emotion. The most common emoticons are those that represent happiness or sadness. Table 2.1 exemplifies some emoticons and their most common associated emotion.

Emotion	Emoticons
Happiness	:), :D, =), =D, :-), :-D, c:, C:, ...
Sadness	:(, :-(, =(, :c, :C, ...
Surprise	:o, :O, o_O, O_O, ...
Bewilderment	D:, ...
Anger	>:(, >=(, >:c, ...
Disappointment	:|, :/, ...
Love	<3, ...

Table 2.1: Some Emoticons and Their Most Common Associated Emotion

It is important to note that table 2.1 presents the *most common* associated emotion for each emoticon since a user might use them for a different purpose than simply conveying their "literal" emotion. The following tweet for example uses a happiness emoticon sarcastically.

(2.55) *I've been w8in 2 hours for my refund... I hate you Walmart :)*

Latest Opinion Mining Techniques and Findings in the Twitter Context

As it was shown in section 2.2, the two first Opinion Mining studies were published in 2002 and correspond to [46] and [76]. In 2005, Read [128] demonstrated that Opinion Mining techniques can generate domain dependency, topic dependency and temporal dependency, meaning that a classifier trained in the movie domain for example, would not perform well in another domain. The author also proposed that emoticons have the potential of being independent from these kind of dependencies, hence providing useful information as features for classification.

After Twitter became available, one of the first studies addressing Opinion Mining applied to Twitter data was published in 2009 by Go et al. [129]. In their study, the authors attempted to classify tweets as positive, negative or neural. Furthermore, they argued that in order to train supervised classifier, a labeled training dataset was required, and given the abundance of data and topics present in Twitter, labeling tweets manually would have required a gargantuan effort. Consequently, they used a set of tweets containing emoticons,

which they defined as "noisy labels." The authors, based on Read's results [128], postulated that the polarity of a tweet was represented by the emoticons contained within it. Therefore a tweet containing a happy emoticon (:)) was considered as being positive, and a tweet containing a sad emoticon (:( ) as being negative. With this labeled corpus they trained a SVM, a Naïve Bayes classifier, and a Maximum Entropy classifier. Some of the conclusions they drew were that the use of POS tags does not provide significant information for a classification based on a bag-of-features approach, and that the simple use of unigrams and bigrams, as a means to represent tweets, provides good results.

A later study by Pak and Paroubek [80] extended the methodology presented by Go et al. by introducing a new type of training data as an addition to tweets with noisy labels. Besides from positive tweets, represented by positive emoticons, and negative tweets, represented by negative ones, they used tweets posted by 44 accounts corresponding to newspapers and magazines such as "The New York Times," and "The Washington Post," among others. With these data, the authors trained a SVM, a CRF classifier and a Naïve Bayes classifier, to classify tweets as positive, neutral or negative, and found that the Naïve Bayes classifier performed the best. They also found that the best features for classification are n-grams and POS tags.

A parallel study carried out by Davidov, Tsur and Rappoport [81], utilized both emoticons and hashtags as features for classification using a KNN algorithm. However, in contrast to the previously mentioned studies, here the authors considered each emoticon and hashtag as a different class, instead of first assigning them to the positive, negative, or neutral classes, and then attempted to classify each tweet into one of them. Additionally, the authors used word-based, n-gram-based, pattern-based and punctuation-based features and found that each one of these types contributes to their sentiment classification framework.

Another parallel study conducted by Barbosa and Feng [130], also exploited emoticons as features for polarity classification, in addition to meta-features (POS tags, prior word subjectivity and polarity), and tweet syntax features (retweet, hashtags, replies, links, punctuation marks, emoticons and upper-case letters, among others). Furthermore, their classification process was separated in two steps, first, they classified each tweet as being objective or subjective, and second, they classified subjective tweets as being positive or negative. They found that the five features that best helped to predict the subjectivity of a tweet were the positive prior polarity of each word, the presence of links, prior word subjectivity, upper-case letters, and presence of verbs. Additionally, they concluded that the best features for polarity classification were the negative and positive prior polarities of each word, presence of verbs, good emoticons, and upper-case letters.

The study by Jiang et al. [24], is based on the work made by Barbosa and Feng [130], but instead of only using tweet syntax features and meta-feaures, they also used target-dependent features. The target of an opinion is defined as the aspect or entity the opinion refers to (see section 2.1.1 to remember the formal definition of an opinion according to Bing Liu), and target-dependent features correspond to those elements of a tweet that contribute to finding this target. In the case of this paper, target-dependent features were a limited number of syntactic constructs defined by a set of manually crafted syntactic rules. Furthermore, they defined opinion-target candidates (they called them extended targets), simply as every noun

phrase found in the tweet. For example, one of such features is defined as follows: A transitive verb with a target candidate as object. In case such rule was fulfilled in a tweet, the feature was considered as *true* (1), else the feature was *false* (0). Evidently, to know if a given word corresponded to the object of a given verb, the authors had to rely on the syntactic parse tree of each tweet (see section 2.3.3 for a reminder on syntactic constructs and parse trees). They finally concluded that incorporating syntactic features into the learning-based analysis yielded better results than target-independent approaches that do not.

Thelwall, Buckley and Paltoglou [131], studied whether popular events, which are commented on in Twitter, were associated with strong sentiment strengths. In order to classify each tweet they used the SentiStrength algorithm, created by them and presented in [132]. This algorithm is capable of handling abbreviations, jargon and most of the previously mentioned Twitter characteristics. Additionally, it returns two outputs for each tweet, a negative and a positive score on a scale from 1 (*no sentiment*) to 5 (*very strong sentiment*). The authors exemplified this classification scheme with the tweet "Luv u miss u," which was classified as having a moderately positive sentiment with a strength of 3, and a slightly negative sentiment with a strength of 2. Moreover, they did not give much details on how their algorithm was built, but only said that it combined a sentiment lexicon with linguistic rules for spelling correction, negations, intensifiers, emoticons and other factors. In the end, they concluded that there was evidence proving that popular events were associated with increases of negative sentiment strength, and that some peaks of interest concerning certain events had strong positive sentiment.

All of the studies previously mentioned in this section have used machine-learning-based methods for polarity classification. The study by Zhang et al. [133] presents a new method that combines both lexicon-based and machine-learning-based methods for Opinion Mining in Twitter. In it, the authors stated that usually lexicon-based approaches have high accuracy but low recall,[34] which translates into opinionated tweets being incorrectly classified as neutral. The reason for this issue is that sentiment-bearing words, in the microblogging context, are very dynamic, making static opinion lexicons obsolete. The authors gave the following example to illustrate this point:

(2.56) *I bought iPad yesterday, just lovvee it :-)*

In this tweet, there is no word that would appear in a lexicon, hence the lexicon-based classifier would categorize it as neutral. Indeed a human might infer that the word "lovvee" means the same as the word "love," but there is no way, a priori, for a computer to know this. To face this issue, the authors proposed to automatically identify tweets that are likely to be opinionated by checking if they contain sentiment-indicators, which are words that do not appear in the opinion lexicon ("lovvee"), but frequently appear in an opinionated context. The basic concept behind their approach was that if a word frequently appears in a positive or negative context, it has a higher probability of being an opinion-indicator. To assess whether an unknown word is an opinion-indicator or not, they applied the χ^2 test to compare observed word frequencies with expected word frequencies. Any word that turned

[34] For more information on recall and evaluation metrics in general, refer to 5.1.1.

to be highly dependent of an opinionated word-set (high χ^2 value) was then considered as an opinion-indicator.

Moreover, the authors incorporated rules to handle intensification, negation, but-clauses and comparative opinions, and they implemented a simple heuristic to perform coreference resolution. Later, they used the output produced by this lexicon-based methodology as training data for a SVM algorithm, with unigrams, emoticons and hashtags as binary features (denoting the presence or absence of them). Finally, after discussing the results of their experiments, they concluded that this hybrid approach performed considerably better than pure learning-based or lexicon-based based approaches.

Another study to propose a hybrid approach for Opinion Mining applied to Twitter data is the one by Vilares et al. [47], where the authors combined lexical, syntactic and semantic features, obtained through an unsupervised Natural Language Processing approach, first presented in [134], and used them as input for a specific implementation of SVM, called Sequential Minimal Optimization (SMO). Some of these features corresponded to frequencies of POS tags and dependency relations, and the binary occurrence of sentiment-bearing tokens (words and emoticons, among others). The authors demonstrated that their approach performed better than pure learning-based approaches, concluded that the morphosyntactic structure of tweets is useful to classify their sentiment, and proposed, as future work, to refine their preprocessing module, to adapt the dependency parsing algorithm to better deal with microblog data and modify some aspects of their features.

To summarize, the earliest studies concerning Opinion Mining in Twitter used mainly machine-learning approaches with simple lexical features. As the field advanced, researchers refined these features to incorporate more information on natural language, such as morphosyntactic information (POS tags) and syntactic information (parse trees, dependency relations). Today most approaches are aiming towards hybrid methodologies that combine unsupervised algorithms to generate the features to be later used in learning-based systems, as exemplified by [133] and [47].

Finally, the survey by Martínez-Cámara et al. [64] provides a more exhaustive review of the latest Opinion Mining studies applied to Twitter in the year 2012. To learn more on specific topics such as temporal prediction of events, political opinion mining, author influence and sarcasm on Twitter, refer to their study.

Chapter 3

Design

In this chapter, the design of the Opinion Mining application will be described by applying the knowledge presented in Chapter 2: Conceptual Framework. The aim is to provide the reader with an integral understanding of *what* was done in the development of the application without giving much detail on *how* it was done. In other words, this chapter will present the application and the logical structure behind it, without specifying details of the implementation such as the algorithms used, the development tools, database types or development environment. These details, and more issues concerning the implementation, will be presented in Chapter 4: Implementation.

This chapter is structured in a manner that is similar to the application structure, in hopes that it will be easier for the reader to understand. Section 3.1 describes the software requirements of the application, section 3.2 presents the application's general architecture and the following sections explain each component of it: section 3.4 exhibits the Data Extraction Module, section 3.5 describes the Preprocessing Module, section 3.6, explains the Polarity Classification Module and finally, section 3.7 depicts the Visualization Module.

3.1 Software Requirements

The first step in building any piece of software is defining the problem that is going to be solved [135]. Here, the problem is highly related to the research hypothesis presented in 1.3, in fact, the hypothesis is contained within the problem: *There are vast amounts of user-generated data in Twitter that could be potentially useful for the retail industry but are difficult to exploit.* Accordingly, this application must attempt to solve this problem, or in other words, eliminate or at least reduce the difficulty of exploiting user-generated Twitter data.

After defining the problem, which represents the foundation upon which to build the software, the requirements must be laid out. According Steve McConnell [135], requirements describe in detail what a software system has to do, and represent the first step toward solving the previously defined problem.

This section exhibits the requirements for the Opinion Mining platform that is going to be developed or, in essence, what are the specific tasks that the application must be able to do while bearing in mind that the ultimate goal is to facilitate the exploitation of Twitter user-generated data. Particularly, given time and resource constraints, the kind of data to be considered will correspond exclusively to opinions. Consequently the application must fulfill the following requirements.

- **Tweet Extraction:** This process should extract tweets containing user-defined terms and save them into a specific database. These are the data that will be later used in the Opinion Mining step.
- **Opinion Mining:** To facilitate the understanding of each tweet, the information contained within them must be summarized in a specific way. For this application, the preferred method for achieving so will be to assign each tweet a polarity score that will represent its positivity, negativity or neutrality. Additionally, the system must be able to process input that is written in Spanish. The idea behind these requirements is to provide the Spanish-speaking user with aggregated information on a given entity, represented by the previously-mentioned user-defined terms. A feature that could be possibly implemented in the future is to detect relevant terms or topics related to an entity and obtain their associated polarity.
- **Data Visualization:** The only way to make the generated data useful for the end-user is to display them in a fashion that is easy to understand and interpret. Therefore, the application will have to fulfill this requirement.

The requirements were purposefully stated in a general way since they will be developed only by one person, and the final product is expected to be just a prototype. Furthermore, the idea behind the development of this application is to exercise the principle stated by McConnell [135] that "just as the more you work with the project, the better you understand it," which implies that, as the project progresses, the requirements will probably vary, requiring them to be flexible to some extent. In a formal software project, however, while still having to be flexible, requirements should be considerably more specific. Now that the general requirements are clear, the architecture for the application can be defined.

3.2 General Architecture

In this section the application's general architecture will be presented. This means that only the most relevant modules and the interactions between them will be presented. In other words, only the minimal components required to understand how the application works will be explained. For a more thorough explanation refer to Chapter 4: Implementation.

The application will be built according to the Opinion Mining pipeline presented in 2.1.2, meaning that it will have a data extraction module for extracting tweets, a data preprocessing module for transforming the raw data into a data structure that will be easier for a computer to process, a polarity classification module charged with the task of assigning a polarity to

each extracted and preprocessed tweet, and finally, a visualization module that will display the processed data to the end-user. Additionally, a data layer will be built to abstractly expose database functionality to the previously mentioned modules. With this, all the Create, Read, Update, and Delete (CRUD) operations, along with database connections will be centralized and easier to manage. Finally, an API that offers an abstraction layer to the most basic polarity classification functionalities will be created.

Both the data layer and API will be explained in Chapter 4 since they are not essential to understanding how does the system work and could be considered as extra functionality. Figure 3.1 displays the system's general architecture.

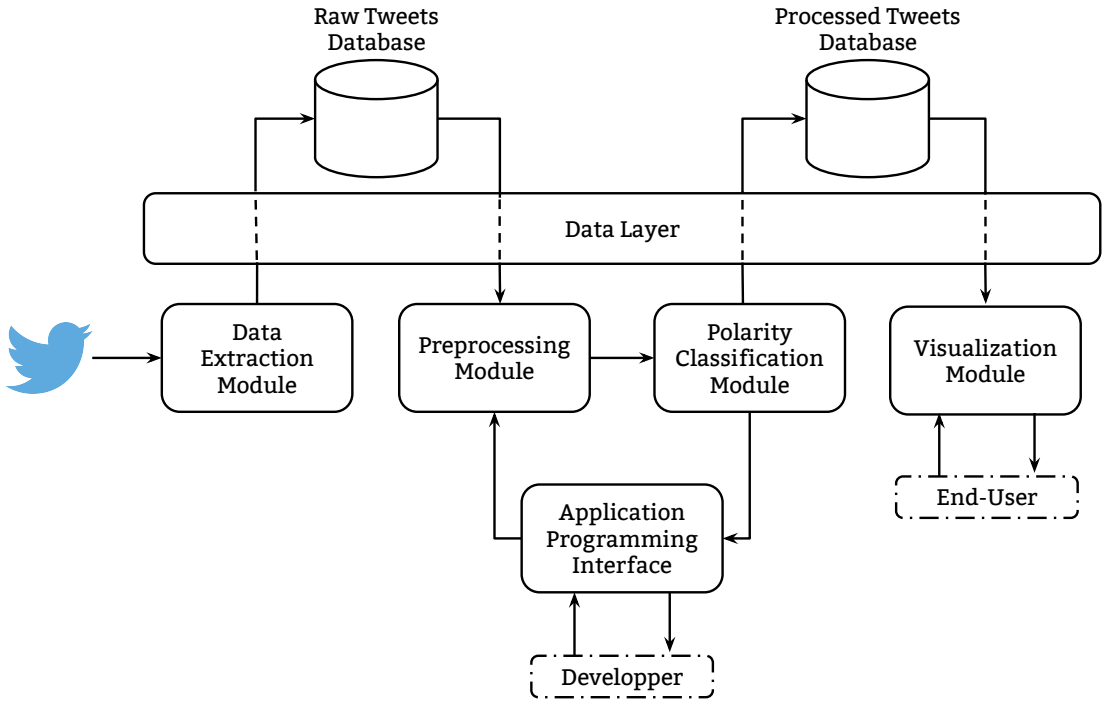


Figure 3.1: System General Architecture.

By observing figure 3.1 it is possible to infer the relationships between modules and users. The architecture considers two types of users: developers that want to build new applications on top of the classification functionalities,[1] and end-users that only care about visualizing the processed data, however, the end-product of this thesis is oriented towards end-users from the retail industry, so further explanations will be focused on them.

3.3 Data Characteristics

Since the case study that will be presented in Section 5.2 concerns the particular retail firm Falabella, only tweets containing the keyword `falabella` will be downloaded. For non-developers, a tweet has a simple enough definition and some special properties, however,

[1]At first the construction of the API was not considered but it arose as a new requirement from the rest of the OpinionZoom project team.

from a data-structure standpoint, a tweet is fairly complex as it has many fields that are not visible for the common user[2]. Some of these fields include the coordinates where the tweet was tweeted, a list of the hashtags, URLs and user-mentions contained in it, and the amount of times it has been retweeted and favorited, among others. Since this study is focused on applying Opinion Mining techniques to the text contained within tweets, text will be the most relevant field. Along with it, the Unix timestamp, screen name of the tweet's author and the tweet id will be extracted. If any other field were to be needed it could be extracted by using the tweet's id along with the API provided by Twitter.

Below, an example of a tweet extracted by the Data Extraction Module is presented:

(3.1)
```
{
    "timestamp": 1412365158,
    "text" : "Increible lo ineficiente de falabella. Me traen una base de una cama
        y SIN PATAS y ahora tengo que esperar 2semanas elcambio@FalabellaAyuda",
    "screen_name" : "barbarabarriosc",
    "status_id" : 518123143287021570
}
```

In it, the user with screen name *barbararriosc* is complaining about Falabella's inefficiency for delivering a bed base with no legs, which meant she would have two wait for 2 weeks before getting a replacement. Below, a more detailed explanation of each field is provided:

- `timestamp`: Unix timestamp which corresponds to the amount of seconds elapsed since January the 1st 1970 (UTC)[3]. The presented timestamp in example (3.1) is 1412365158 seconds which corresponds to 2014-10-03 at 19:39:18 UTC.
- `text`: The tweet itself.
- `screen_name`: Username of the tweet's author.
- `status_id`: Unique identifier of the tweet.

These fields provide all the necessary information to be used later in the process. Additionally, since the screen name might be changed at any time by the user, the only relevant field for querying the Twitter API to obtain every remaining field is the status id. Likewise, the status id is the only required field for reconstructing the URL where the tweet is located. The latter corresponds to `http://twitter.com/barbarabarriosc/status/518123143287021570`, however the screen id segment can be replaced by any string and it will be automatically corrected by twitter, so `http://twitter.com/any_random_name/status/518123143287021570` also works.

[2]Refer to the following URL to see all the components of a tweet:
`https://dev.twitter.com/overview/api/tweets`, Accessed on May 07, 2015

[3]`http://www.unixtimestamp.com/index.php`, Accessed on June 22, 2015

3.4 Data Extraction Module

The Data Extraction Module (DEM) is located at the beginning of the process and is charged with two responsibilities, first, it must extract the data from Twitter and second, it must save the data into a database.

The DEM was designed with a simple goal in mind: to be able to extract and save every tweet in Spanish mentioning the keyword "falabella." According to the Data Acquisition subsection of section 2.1.2, there are two ways to achieve this task. The first is to use Twitter's API to obtain the tweets and the second is to create a Web Crawler to do so. The chosen option was the latter because Python's wrapper for the API didn't allow the developer to filter the extracted tweets according to their language, so, at the time, it seemed best to create a simple crawler to achieve the task while filtering out non-Spanish tweets.

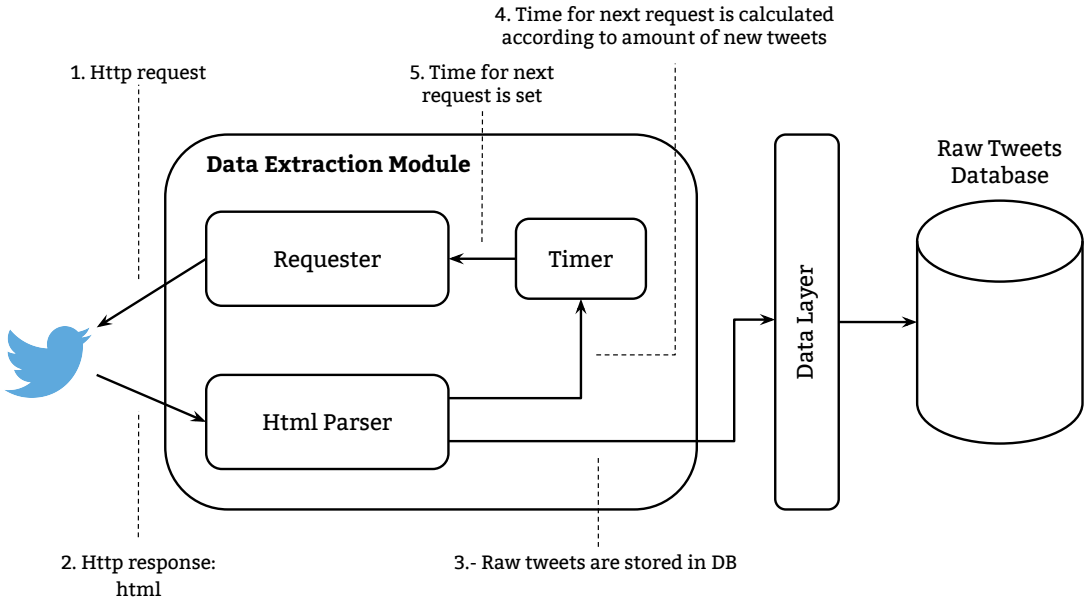


Figure 3.2: Data Extraction Module Architecture.

The DEM is composed of three sub-modules, as shown in Figure 3.2: a requester, a parser and a timer. The requester's main task is to handle the http requests to Twitter's search page (`https://twitter.com/search-home`). Twitter's response, in form of plain html text, is then handled by the parser, whose task is to extract the fields mentioned in section 3.3 and to form the data structure comprising the raw tweet. Next, the tweet is saved in the database by means of the Data Layer, and data related to the amount of new tweets extracted is passed to the timer, which finally decides when to make another request and passes this information to the requester. The timer's decision is based solely on the amount of new tweets extracted in the last request; lower frequency implies more time between requests and vice versa.

Finally it is worth mentioning that the html response coming from Twitter contains at most the 20 latest tweets. The previously presented architecture works because the frequency of new incoming tweets is less than 20 per cycle. However, if this were not the case it would

be necessary to redesign this module to streamline the tweet extraction process. A possible way to do this would be to create a system capable of spawning several parallel processes, for example.

Details on the DEM implementation, including the algorithms involved in each sub-module, scheduling and the development environment can be found in Section 4.2.

3.5 Preprocessing Module

The preprocessing Module (PM) is next in the process, and is charged with the task of transforming each tweet's text string (the `text` field in the representation introduced in section 3.3), into a data structure that is exploitable by the remainder of the process. Most of the steps carried out by this module correspond to the first steps of Natural Language Processing, as depicted in Figure 2.3. Those that are not, correspond to ad hoc tasks implemented specifically for handling elements that are characteristic of Twitter, as presented in section 2.4.3. For understanding the underlying theory upon which this module is built, see section 2.3.1.

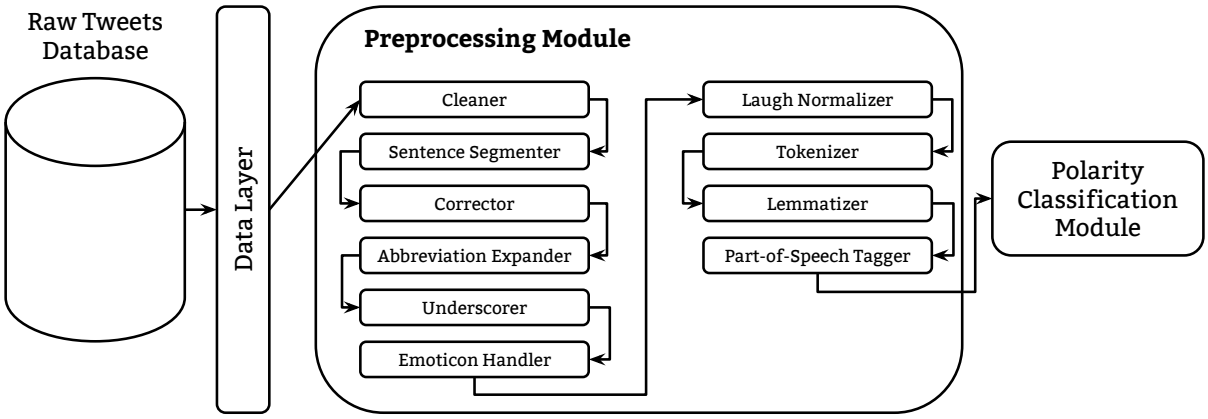


Figure 3.3: Preprocessing Module Architecture.

Figure 3.3 illustrates the elements composing the PM. Both the sentence segmenting and tokenizing steps can be immediately associated with the text preprocessing step of the NLP pipeline: lemmatizing to the lexical analysis step, and POS tagging to the syntactic analysis step. The remaining sub-modules are ad hoc implementations for handling Twitter text. The input for this module corresponds to a string of text, and the output to a list of POS-tagged sentences. Below, a more detailed explanation for each sub-module is presented.

Cleaner: Removes unused elements from tweets such as URLs, encoding symbols ("&#39," "&quot"), and optionally, hashtags and mentions.

Sentence Segmenter: Divides a string of text into its composing sentences. This allows to treat the sentence as a unit of analysis instead of the whole paragraph.

Corrector: Handles every miscellaneous correction tasks. These include, adding spaces around punctuation marks and removing unnecessary spaces. Some functionalities yet to be implemented include the normalization of "written screams" (goooooood ⇒ good) and the correction of spelling errors.

Abbreviation Expander: Expands common abbreviations into their expanded form.

Underscorer: Underscores composite expressions such as composite intensifiers (*lo más* ⇒ *lo_más* which translates to "the most"), noun phrases (Noam Chomsky ⇒ Noam_Chomsky) and other common constructs (*sin embargo* ⇒ *sin_embargo* which translates to "however").

Emoticon Handler: Handles the emoticon related tasks.

Laugh Normalizer: Normalizes different types of "written laughs" into a standard form.

Tokenizer: Separates each sentence in its composing tokens: words, punctuation, numbers and dates.

Lemmatizer: Transforms each word into its non-inflected dictionary form.

Part-of-Speech Tagger: Once every other task is completed, this module attempts to tag every word of the sentence with its Part of Speech.

Below, the application of every step of the preprocessing phase is illustrated. The raw tweet that will be preprocessed corresponds to:

(3.2) *Esta cuenta esta de luto por la muerte de Oscar de la Renta.
Pero uds ni saben quien es, pq se visten en Falabella
Sigan*

Which literally translates to "This account is mourning the death of Oscar de la Renta. But you don't even know who he is because you dress in Falabella Continue" After the tweet is cleaned it looks like:

(3.3) *Esta cuenta esta de luto por la muerte de Oscar de la Renta. Pero uds ni saben quien es, pq se visten en Falabella Sigan*

Later, the sentence segmenter divides the tweet into its two composing sentences.

(3.4) *[Esta cuenta esta de luto por la muerte de Oscar de la Renta.][Pero uds ni saben quien es, pq se visten en Falabella Sigan]*

The corrector then adds one space around each punctuation mark and deletes unnecessary spaces.

(3.5) *[Esta cuenta esta de luto por la muerte de Oscar de la Renta .][Pero uds ni saben quien es , pq se visten en Falabella Sigan]*

Next, every abbreviation is expanded.

(3.6) *[Esta cuenta esta de luto por la muerte de Oscar de la Renta .][Pero ustedes ni saben quien es , porque se visten en Falabella Sigan]*

After, composite expressions are underscored,

(3.7) *[Esta cuenta esta de luto por la muerte de Oscar_de_la_Renta .][Pero ustedes ni saben quien es , porque se visten en Falabella Sigan]*

Since there are not written laughs nor emoticons, the tokenizer can proceed to segment each sentence.

(3.8) *[(Esta)(cuenta)(esta)(de)(luto)(por)(la)(muerte)(de)(Oscar_de_la_Renta)(.)][(Pero)(ustedes)(ni)(saben)(quien)(es)(,)(porque)(se)(visten)(en)(Falabella)(Sigan)]*

Later, the lemmatizer brings each word to its non-inflected dictionary form.

(3.9) *[(este)(cuenta)(estar)(de)(luto)(por)(la)(muerte)(de)(Oscar_de_la_Renta)(.)][(pero)(ustedes)(ni)(saber)(quien)(ser)(,)(porque)(se)(vestir)(en)(Falabella)(seguir)]*

Finally, each token is tagged with its part of speech.

(3.10) *este cuenta estar de luto por la muerte de Oscar_de_la_Renta .*
d n v d n s d n s n f

(3.11) *pero ustedes ni saber quien ser , porque se vestir en Falabella seguir*
c p c v p v f c p v s n v

The current implementation needs simplified tags to be used as input for the next step in the process, which is why they might not be familiar to the reader. For more details on this refer to Section 4.3.

3.6 Polarity Classification Module

The Polarity Classification Module (PCM), represents the core of the whole application, and coincides with the Opinion Mining Core Process described in section 2.2. This module was designed to perform Opinion Mining at the sentence level, with an unsupervised Lexicon-Based Approach. More specifically, it uses a dependency parser for obtaining the grammatical

function of each word, and how they relate to each other, based on the work by Vilares et al. [43]. Theoretical background for this approach can be found in subsection Dependency Grammars of section 2.3.3.

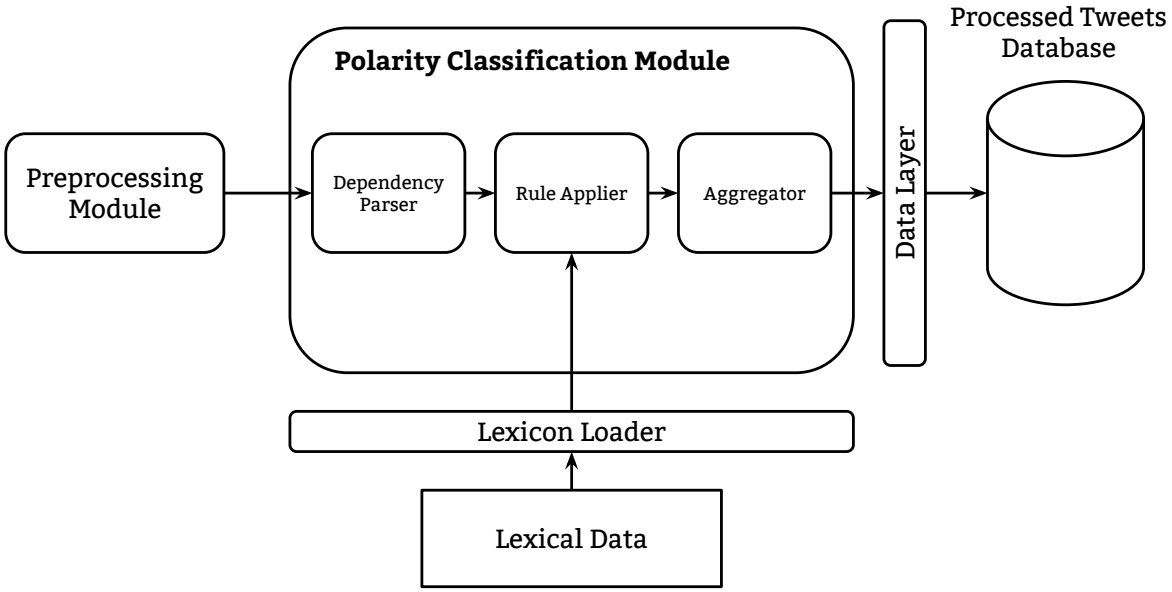


Figure 3.4: Polarity Classification Module Architecture.

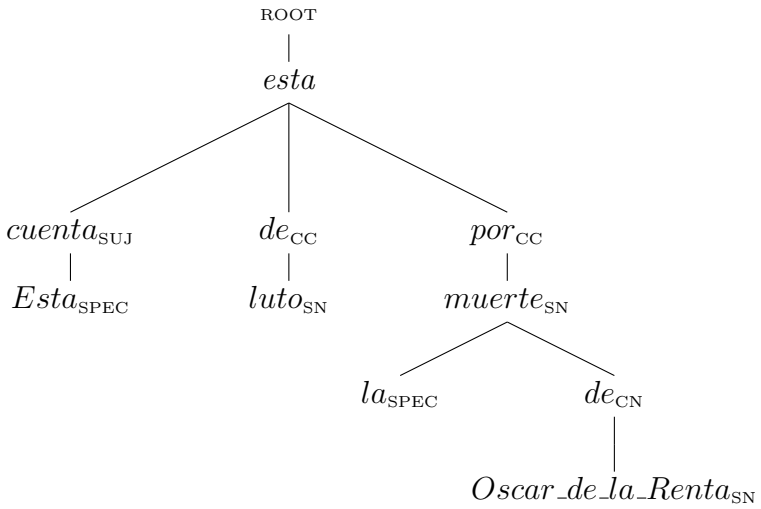
Figure 3.4 presents the architectural design of the Polarity Classification Module (PCM). Even though this is the most important module of the whole process, its complexity is not reflected on its architecture. This module's complexity will be more apparent when its implementation is presented in Section 4.4.

The PCM works as follows. First, the dependency parser receives the POS-tagged sentences of the tweet being analyzed, and returns the dependencies between each word composing it. Second, a set of rules concerning intensification, negation and adversative clauses is applied, while considering external lexical data for calculating each word's polarity and, later, the tweet's overall polarity. Finally, the tweet is rebuilt, the previously calculated polarity is associated to it, and everything is saved to the Processed Tweets Database. Below, each sub-module is briefly described.

3.6.1 Dependency Parser

This sub-module is charged with defining the dependencies between words in any given POS-tagged sentence. Taking the example sentences (3.10) and (3.11) given as output by the preprocessing module as input would produce the result presented in Figure 3.5.

a.



b.

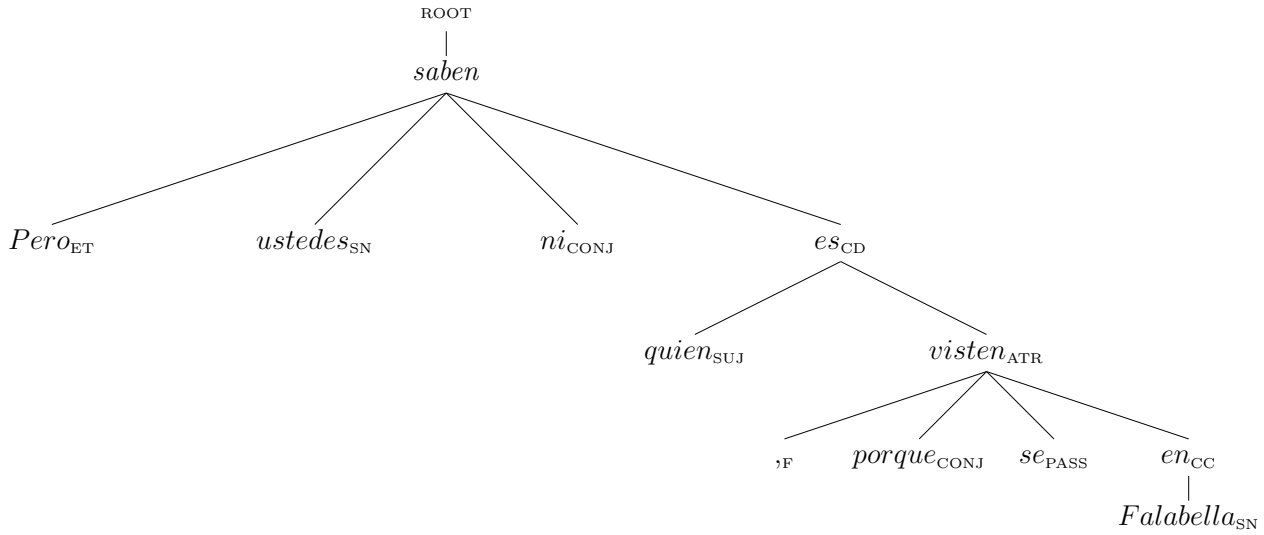


Figure 3.5: Dependency trees representing the sentences of a tweet.

Evidently, each sentence produces a different dependency tree. The first sentence, presented in Figure 3.5.a, is well formed and produces a predictable output. The principal meaning of it is given by the verb "*esta*" (is), which is executed by the subject "*cuenta*" (account). Every noun and noun phrase is labeled as such ("*luto*" (mourning), "*muerte*" (death), "*Oscar_de_la_Renta*") indicated by the tag SN, and the relationship between words is correctly defined.

The second sentence represented in Figure 3.5.b, on the other hand, is not correctly tagged. The root verb "*saben*" (know) should be executed by the subject "*ustedes*" (plural "you") but this word is instead tagged as a noun phrase SN. Furthermore, the word "*es*" (is) is incorrectly tagged as a direct object (verbs cannot be direct objects). A possible cause for this is that the language structure used in tweets is considerably different from the one used in the corpus for training the parser. Regrettably, Twitter corpora tagged at the dependency level do not exist yet, so it is not possible to validate how well the parser performs for this kind of data nor to train the parser with it. At any rate, the quality of the parser will be

reflected in the validation results of the overall polarity classification quality, presented in Section 5.1.

3.6.2 Rule Applier

This sub-module is the core of the Opinion Mining engine. Its main task is to take the dependency tree generated by the dependency parser and the domain-independent polarity of each word given by the Lexicon created by Maite Taboada et al. [78] – described in Section 4.4.1–, and apply heuristic rules concerning intensification, negations and subordinate clauses for propagating the polarity from the leaf nodes of the dependency tree to the root. The rules used in this work were the same that Vilares et al. applied in [43], and their goal is to provide a more accurate representation of the speaker's intended meaning.

Intensification

The first kind of rules correspond to *intensification*. An intensifier, simply put, is a word that can *amplify* or *attenuate* the meaning of another word [78]. The most intuitive examples of an amplifying and attenuating intensifier are *muy* (very), and *un poco* (a little), respectively. In order to understand the role of an intensifier in a specific sentence, it is first necessary to obtain the scope of intensification, that is, what is being intensified (or attenuated). Then, depending on the degree of intensification and its scope it is necessary to calculate the new shifted polarity. Hence the intensification rule is defined as follows:

Intensification Rule: *If an adverb is labeled as being a non-head determiner (SPEC, ESPEC), an adverbial phrase (sadv) or an adjunct (CC) then the adverb is considered as an intensifier and its head is defined as the scope of the intensification.*

(3.12) *Me siento completamente estafado por Falabella.*

(3.13) *I feel completely cheated by Falabella.*

Example (3.12), which directly translates into (3.13), can be represented as the following dependency tree:

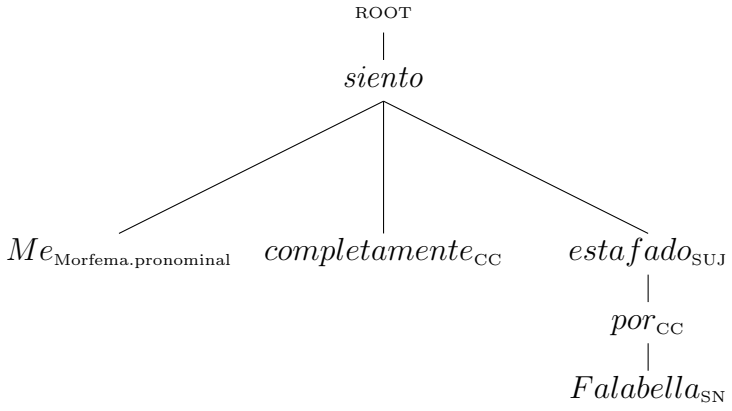


Figure 3.6: Dependency tree of a sentence with intensification.

The word *completamente* (completely) is an adverb with an adjunct relationship (CC) to its parent node which means it is considered as an intensifier by the Intensification Rule. This implies that the final polarity of the node represented by the word *siento* will be modified by *completamente*. Since the word *estafado* bears a negative polarity, the intensification will make the whole sentence even more negative after it is transmitted to the head node.

So, now that the scope of intensification is defined, it is necessary to specify how the intensification will be applied. After the polarity of the word *estafado*, corresponding to -4[4], is transmitted to its head *siento*, it is then intesified by 25%, value associated to *completamente*. The final polarity of the sentence will be $-4 * (1 + 25\%) = -5$. The propagation process is represented in Figure 3.7.

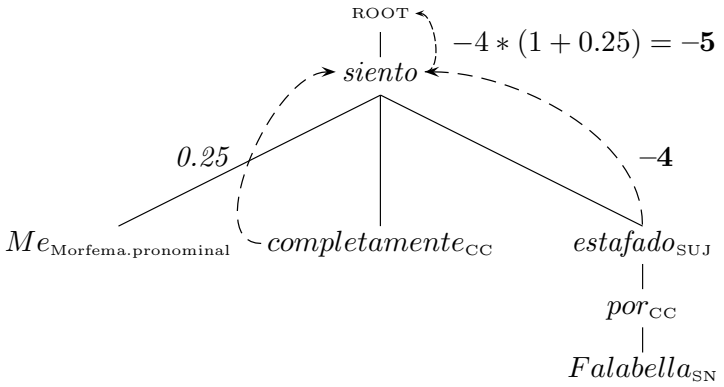


Figure 3.7: Polarity propagation of an intensified sentence.

Negation

The second kind of rules correspond to *negation*. A negation word is simply a word that negates its scope. In terms of the OM application, this means that the polarity of the scope of negation is shifted by a given amount. Just like in the study by Vilares et al. [43], the only words that will constitute a negation will be *no* (not), *nunca* (never) and *sin* (without), even though there are many others (*nadie* (no one) and *ninguno* (none), among others). Similar to the intensification rules, to apply negation it is necessary to identify the scope of negation first, and then apply the negation to the detected scope. Vilares et al. state that the scope of negation of the word *sin* is always its child node, hence there is no need to define a more elaborate rule for this word. *No* and *nunca* however, require a more complex set of rules, described below:

Negation Rules: *When a token is parent to a word* no *or* nunca *labeled as NEG or MOD, one of the following heuristics is applied:*

> **Subjective Parent Rule:** If the parent token has an associated polarity, then the scope of negation is considered to be this single token. This rule is represented in Figure 3.8.

[4]This word was not initially in the lexicon created by Taboada et al. and was manually added.

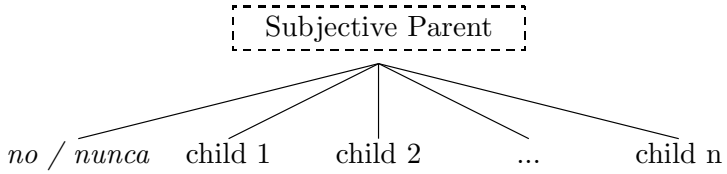


Figure 3.8: Subjective parent rule representation.
Source: [43]

Subject Complement – Direct Object Rule: If a sibling (node at the same level) of the negation word is tagged as an attribute (ATR) or a direct object (CD), then this sibling corresponds to the scope of negation. This rule is represented in Figure 3.9.

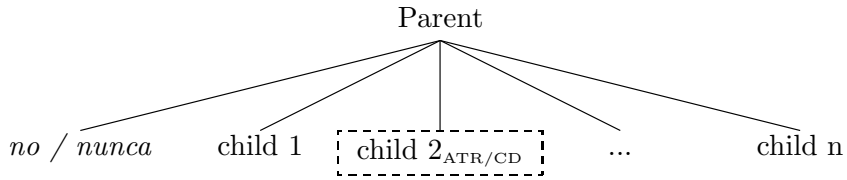


Figure 3.9: Subject Complement – Direct Object Rule representation.
Source: [43]

Adjunct Rule: If the negation word has one or more siblings tagged as Adjuncts (CC), the first one of this occurrences is considered as the scope of negation. This rule is represented in Figure 3.10

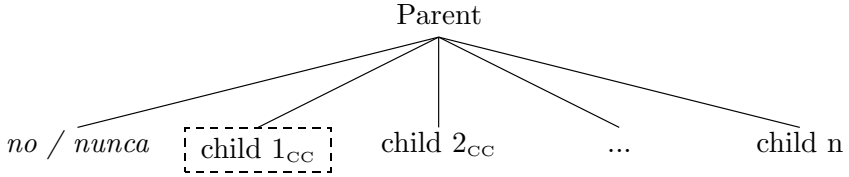


Figure 3.10: Adjunct Rule representation.
Source: [43]

Default Rule: If none of the previous rules matches then the scope of negation is considered to be all of the negation word's siblings, as represented in Figure 3.11.

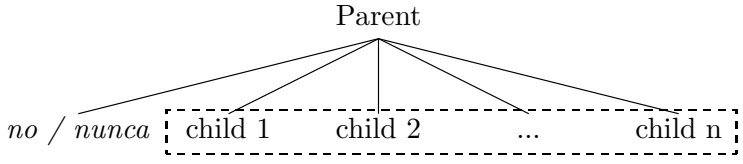


Figure 3.11: Default Rule representation.
Source: [43]

Once the scope is determined, the polarity of the scope is shifted by 4 if the negation word is *no* or *nunca* and by 3.5 if it is *sin*. Vilares et al. [43], justify the former choice by

citing Taboada et al. [78], and the latter by saying that the adverb *sin* is less potent since it has a local scope. They further support their claim by proving it improves their polarity classification accuracy. Finally, the polarity shift depends on the polarity of the scope: if the scope has negative polarity, then the shift occurs in the positive "direction" (the negation value is *added* to the current polarity). If, on the other hand, the scope has positive polarity, the shift occurs in the negative "direction" (the negation value is subtracted from the current polarity).

Figure 3.12 displays an example of applying a negation rule to the sentence *"No me gusta Falabella"* (I don't like Falabella). The corresponding rule for this specific example is the Subjective Parent Rule, because the head of the negation word *no*, corresponding to *gusta*, has a positive polarity of 2. Since its polarity is positive the value of the negation term is subtracted from it, resulting in an overall polarity of –2.

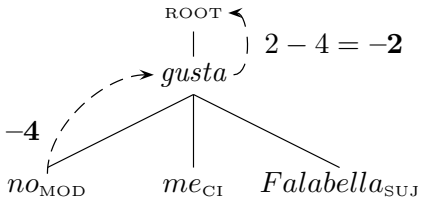


Figure 3.12: Example of the application of the Subjective Parent rule.

Adversative Clauses

The final set of rules are related to *adversative clauses*. An adversative clause expresses meaning that opposes the main clause. Vilares et al. [43], state that these constructs have an effect similar to the intensification phenomenon in the sense that the polarity of a given clause can be amplified, attenuated or excluded from the overall sentence polarity. Furthermore, the authors recognize two types of adversative conjunctions, *restrictive conjunctions* and *exclusive conjunctions*. These are described below:

Restrictive Conjunctions: They increase the polarity strength of the subordinate clause while attenuating that of the main clause. In this thesis, just like in Vilares's work, the only indicators of this type of conjunction will be considered as the words *pero* (but) and *mientras* (while). Additionally, the weight of the main clause will be set as 0.75, while that of the subordinate clause will be set as 1.4. The authors argue that these values yielded the best results while testing their system with the SFU Spanish Review Corpus [136].

Exclusive Conjunctions: These type of conjunction ignore the sentiment of the main clause and maintain that of the subordinate clause. The words considered in this study for indicating this type of conjunction are *sino* and *sino que*, which approximately translate to "but rather" and "but also" (as in "not only ... but also ...") respectively. As it might be inferred from what was said before, the weight of the main clause associated with this type of conjunction is 0 while that of the subordinate clause is 1.

To apply both of these rules, an artificial node is created in the dependency tree, located as parent of the adversative conjunction word. This way, the subtree at the left of the node can be considered as the main clause, while the one at the right as the subordinate clause. This facilitates the polarity calculation while considering the previously mentioned weights. Figure 3.13 exemplifies the dependency tree for the sentence: *"En Falabella atienden lento pero atienden bien."* (Service in Falabella is slow but good) containing a restrictive conjunction, and how the sentiment propagates from the leaves to the root. The node represented as two brackets corresponds to the artificial node.

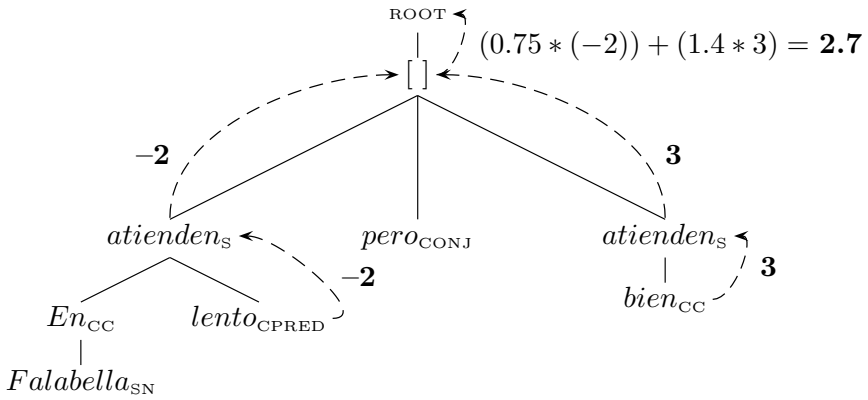


Figure 3.13: Example of a sentence with a restrictive conjunction.

3.6.3 Aggregator

The final sub-module of the Polarity Classification Module corresponds to the *Aggregator*. This sub-module is charged with the simple task of reforming the tweets that were separated into their composing sentences, and further into their composing tokens. The Aggregator takes the output returned by the Rule Applier and appends it to the reformed tweet. This way the final output, composed by the full tweet and its corresponding polarity, is returned and saved into the "Processed Tweets Database," which can be accessed later by the Visualization Module.

The main advantage of encapsulating this simple functionality in its own module is that, later, it would be possible to add new rules for guiding the aggregation process. For example, if the level of analysis were to be refined to consider aspects, then the Aggregator could be modified to adapt to this new context. Likewise, if it were deemed best to assign a greater weight to the polarity of the sentences near the end of the tweet, or any text being analyzed, the Aggregator would be the sub-module charged with doing so.

3.7 Visualization Module

3.7.1 Architecture

The final module of the Opinion Mining platform corresponds to the *Visualization Module*. This module is responsible of displaying the polarity data obtained in previous steps in a

user-friendly way. Figure 3.14 portrays its architecture.

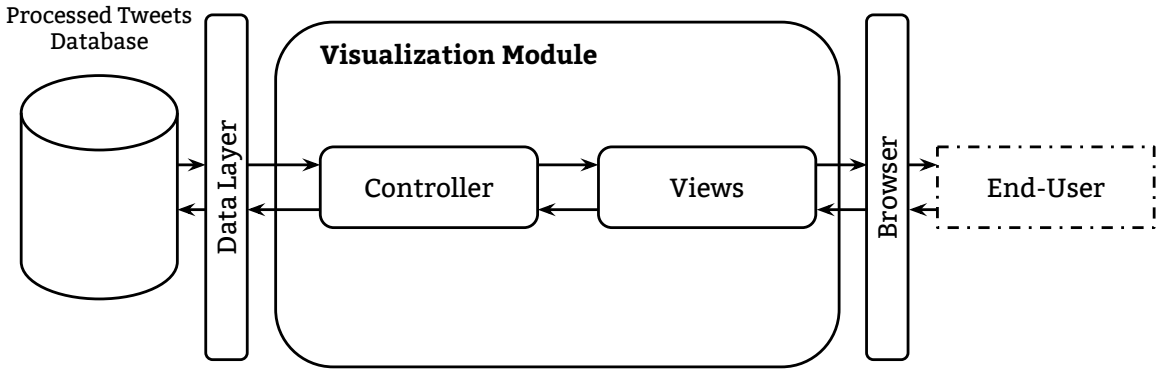


Figure 3.14: Visualization Module Architecture.

This module is composed by two sub-modules, the *Controller* and the *Views*. End-User interaction is limited to the browser, meaning that every request the user wishes to perform, and every response the system returns, is made through it. The information displayed by the browser, as well as the requests made by it are directly handled by the Views sub-module. This module works as a bidirectional formatter, meaning it "translates" the user's requests into something the controller can understand in order for it to request data further down the line. At the same time, it formats the system's responses into something the browser can understand and display to the end-user. Finally, the controller is in charge of receiving the formatted requests from the Views sub-module, processing them, deciding what to do next, and performing additional data requests to the Processed Tweets Database through the Data Layer (also known as the Model in a Model View Controller architecture).

A typical data flow could be exemplified as follows. A user wants to see the latest positive tweets, so he navigates to the corresponding menu and clicks the option that will take him there. The click request will be captured by the browser, sent to the Views which will translate the click into the specific data request for the Controller. Later, the Controller will perform the necessary operations for communicating with the database, requesting the relevant data, formatting it for the Views and sending them back to the latter. Then, the Views will decide how to better display the data (graphs, tables, pie-charts, etc.), and do it through the browser. Finally, the user will be able to see the data he requested earlier.

3.7.2 Web Platform Prototype

Below, some screenshots of the Web platform prototype are presented.

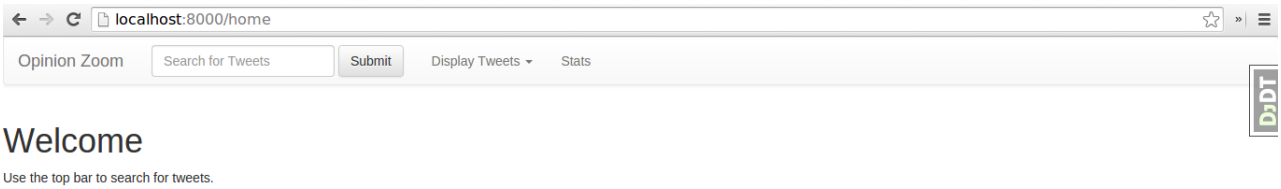


Figure 3.15: Website Landing Page.

Figure 3.15 shows the Landing Page of the created platform. From there the user can choose to search for keywords contained within the tweets, to display tweets that are very positive, positive, neutral, negative or very negative, or to show some statistics concerning the polarity of available tweets. Evidently, the amount of functionality and aesthetic detail are limited due to time restrictions, indeed, a whole thesis could be devoted to finding out how to display the data in the best possible way.

Figure 3.16 displays the view of the tweets tagged as "Very Positive" while figure 3.17 displays those are tagged as "Very Negative."[5] Search results are divided in two sections, a graph that displays the evolution of the polarity of tweets containing this keyword and the list of these tweets. Figure 3.18 displays the graph search results for the keyword "servicio" (service), while Figure 3.19 displays the list results. Again, it is clear that for production-level software, the graph visualization should have more functionality such as an aggregation selector for selecting whether to display data aggregated daily, weekly, monthly or yearly, and the option to select how much data to display, among others.

[5]Some tweets contain strong language.

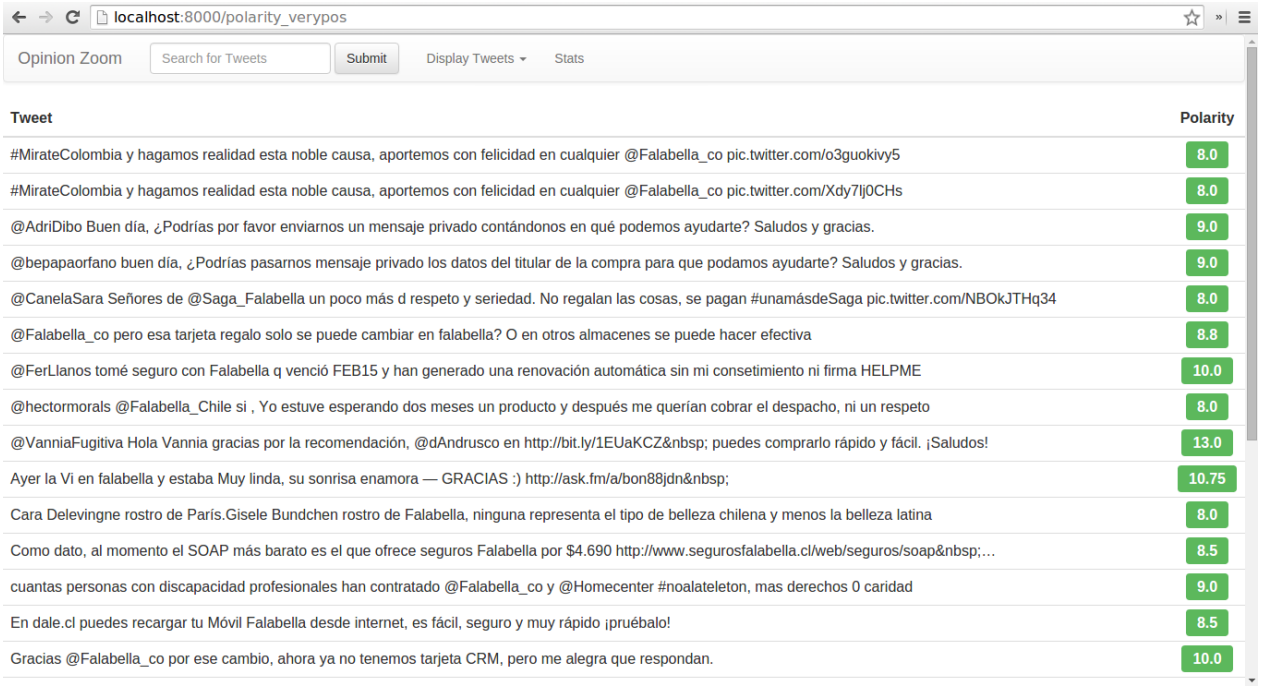


Figure 3.16: View of the tweets tagged as "Very Positive."

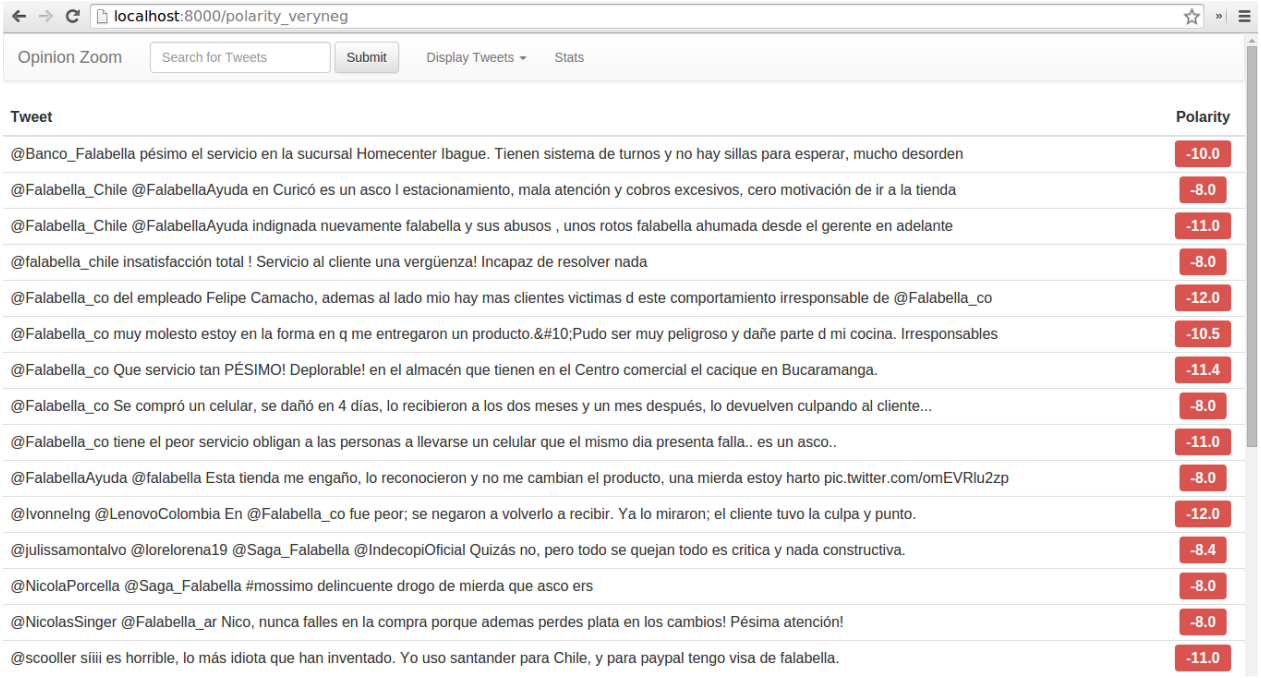


Figure 3.17: View of the tweets tagged as "Very Negative."

Figure 3.18: View of the graph search result of the keyword "servicio."

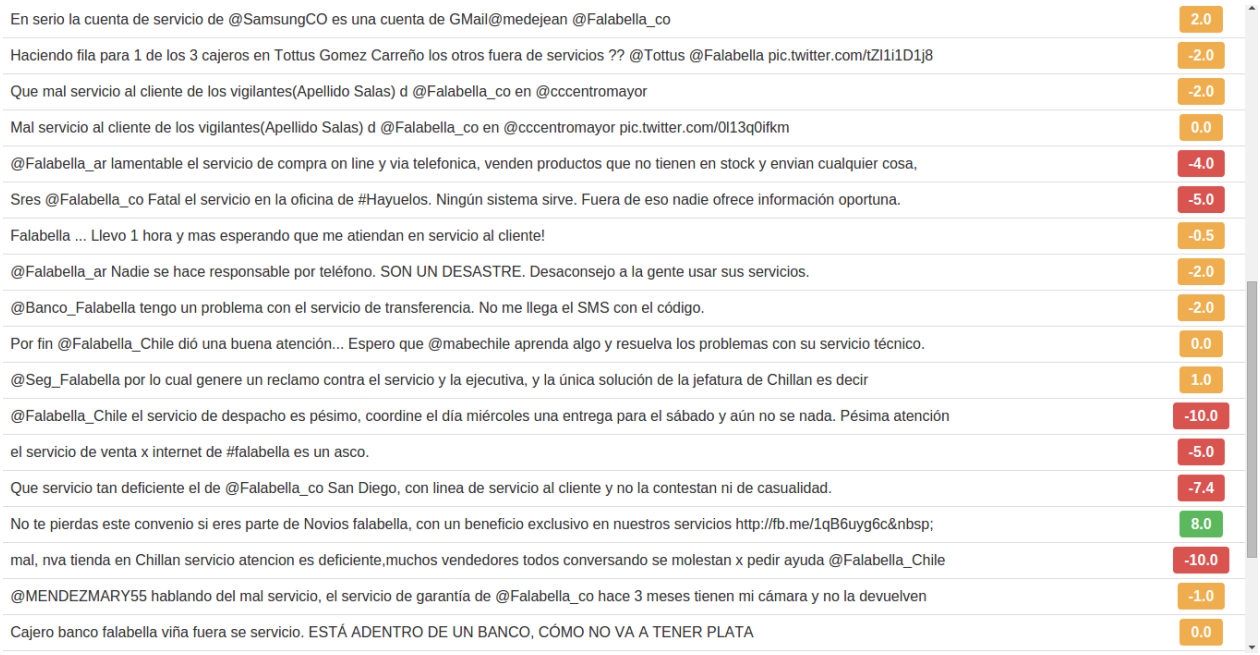


Figure 3.19: View of the list search result of the keyword "servicio."

Finally, Figures 3.20 and 3.21 show the statistics view. In the prototype version of the platform, only two visualizations are available: the polarity distribution (the amount of tweets tagged with a polarity score ranging from -15 to 15) as a bar chart, and a polarity label distribution (percentage of tweets tagged as very positive, positive, neutral, negative or very negative) as a pie chart.

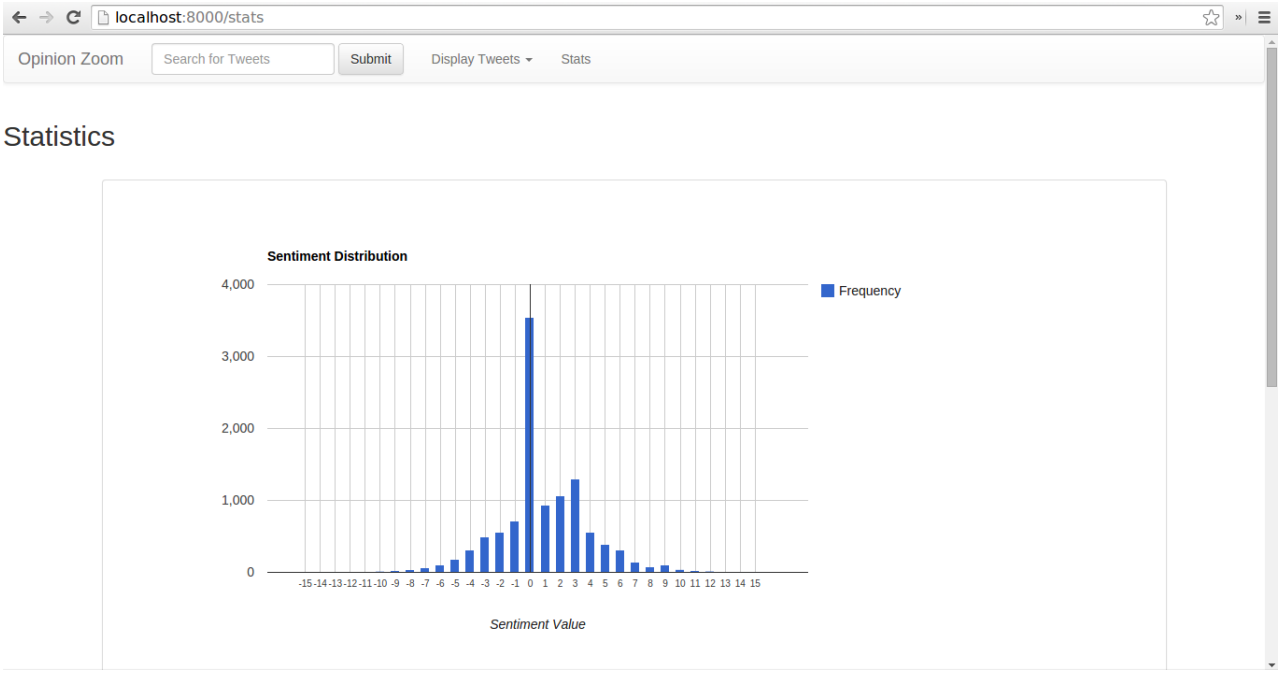


Figure 3.20: Statistics View: Polarity Distribution.

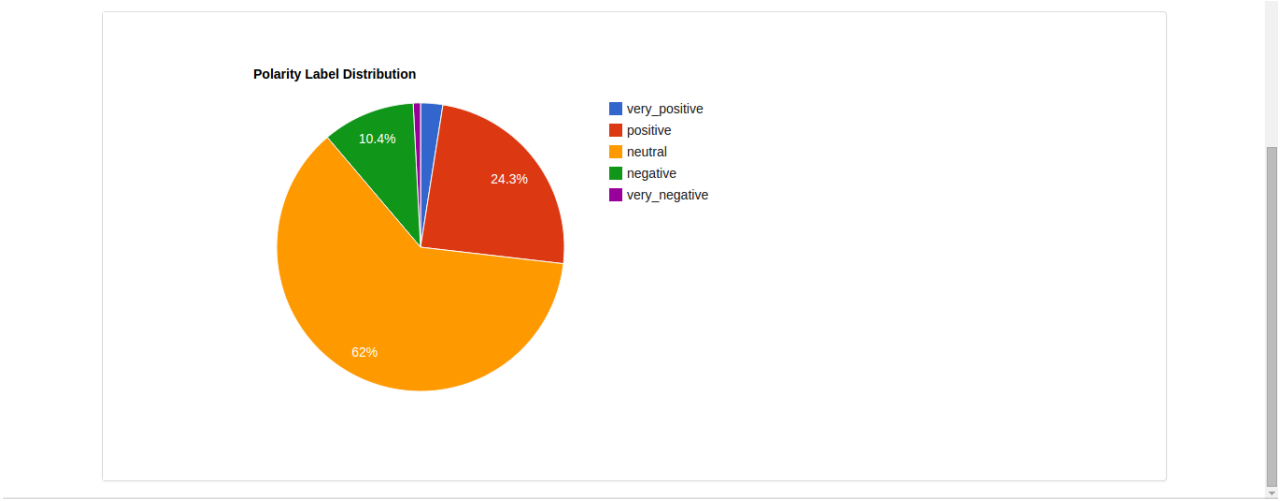


Figure 3.21: Statistics View: Polarity Label Distribution.

Since this section is only concerned with visualization, the results obtained will not be discussed here. For a detailed analysis of the results and insights obtained with this tool please refer to Section 5.2. The visualization displayed in this section represents a glimpse of what could be achieved with the software developed in this thesis. Given the modular way the software is designed, the visualization layer is totally interchangeable and could be easily replaced by another module or even by a proprietary visualization and reporting tool such as Tableau.[6]

[6] `http://www.tableau.com/` Visited June 8, 2015.

Chapter 4

Implementation

This chapter contains information about the OM platform on a lower level, and is intended for the more technical reader that wishes to understand how the application was coded. While the previous chapter (Chapter 3: Design) described *what* the platform does, this chapter will offer deeper insights on *how* it does it.

It is worth mentioning that some of the structures mentioned in the previous chapter were not implemented as depicted but with some minor changes. The main reason for this is that the design represents the logical structure of the software which can not always be implemented optimally.

This chapter will be structured as follows: First, third-party resources will be described, second, the data extraction module implementation will be presented, followed by the description of the Preprocessing, Polarity Classification, and Visualization modules' implementation. Finally, the general implementation architecture will be depicted.

4.1 Third-Party Resources

4.1.1 Environment

Operating System

The development of the Opinion Mining platform began on a Windows 7, 64-bit Operating System, but it was made clear early in the development process that said system would not be the optimal for several reasons. Two of the most compelling ones were that most of the resources needed for developing the platform were available only for Unix-based systems and that these systems offer a significant amount of tools that are not available on Microsoft systems. As a result, most of the development was carried out on a Ubuntu 14.04, 64-bit Operating System.

Programming Language

The chosen programming language for the development of the Opinion Mining Platform was Python 2.7.[1] Python is an interpreted scripting language that was first released in 1991. Eric Raymond [137], defines a scripting language as a programming language designed to "glue" together other applications and tools. He points out however, that Python is one of the major scripting languages that has outgrown this definition and is now a "standalone general-purpose programming language of considerable power." Additionally, Python includes both built-in and third-party high quality libraries for a great variety of application domains, some of which are Web and Internet development, scientific and numeric computing, education, graphical user interfaces (GUIs), and general software development.[2]

Some of the features that characterize Python are that it encourages clean and readable code, it is scalable to large projects with many cooperating developers, allows coding in object-oriented style but does not enforce it, it is highly portable between Unixes and other operating systems, and has an active community constantly maintaining and improving it. Its downside is that it is inefficient and its execution is slow compared to other compiled and scripting languages, but since it is friendlier than other languages, it offers fast prototyping speed which gives it an advantage for creating applications that are not speed-critical or highly complex. Further, there is a great number of applications whose speeds are limited by external factors, such as network waits or Input/Output operations (like the one developed in this thesis), making the language run speed the least important bottleneck. Moreover, Python offers integration with both Java and C, which enables developers to write code in any of these languages in order to improve the speed of critical modules [137].

Given that the platform developed in this thesis did not need to have production-level quality, and that there were important time constraints for its development, Python seemed to be the most appropriated language for developing it. In the end, its fast prototyping speed and ease of use allowed the creation of the platform within the time limits.

4.1.2 MongoDB

MongoDB[3] is a non-relational document-oriented database. The main difference between MongoDB and a traditional relational database such as MySQL or PostgreSQL is that it stores its data as documents instead of rows and columns. Each document uses a JavaScript Object Notation (JSON[4]) structure. This structure maps well to objects in object-oriented programming, and specifically, to Python dictionaries. Additionally, in document databases, schemas, as traditionally defined, do not exist; instead each document can be composed of different fields, which offers great flexibility for modeling unstructured and dynamic data [138].

The decision for using this database in the implementation of the OM platform was mainly

[1] `https://www.python.org` Accessed on July 22, 2015

[2] For a complete list of Python libraries visit the Python Package Index `https://pypi.python.org/pypi`, Accessed on July 22, 2015

[3] `https://www.mongodb.org/` Accessed on August 03, 2015

[4] `http://json.org/` Accessed on August 03, 2015

supported by its transparency with Python's dictionaries, the flexibility it offers – indeed, not having to create or modify schemas and tables every time the structure of the data changed, proved to be very time-saving –, and the fact that the challenge of learning how to use a new trending technology seemed very appealing.

4.1.3 TreeTagger

The chosen tool for performing the Part-of-Speech Tagging process was the TreeTagger[5]. The main reason for this decision was that this tagger had already been used successfully by other team members of the lab where the author of this thesis carried out his research, hence saving him the time of researching which would be the better option. In contrast, Vilares et al. [43], the authors that inspired the work developed in this thesis, used a transformation-based tagger, called the Brill tagger (see the Part-of-Speech Tagging subsection of section 2.3.3).

The TreeTagger is a probabilistic (stochastic) tagger that works by modeling the probability of a tagged sequence of words [99]. More specifically, it is a Markov Model tagger which relies on a decision tree for estimating contextual parameters. The probability of a tagged sentence is defined as follows:

$$p(w_1 w_2 \dots w_n, t_1 t_2 \dots t_n) \tag{4.1}$$

Where $w_1 \dots w_n$ are the words and $t_1 \dots t_n$ their corresponding tags. According to the Bayes' Theorem, expression (4.1) is equivalent to:

$$p(w_n | w_1 \dots w_{n-1}, t_1 \dots t_{n-1} t_n)\ p(t_n | w_1 \dots w_{n-1}, t_1 \dots t_{n-1})\ p(w_1 \dots w_{n-1}, t_1 \dots t_{n-1}) \tag{4.2}$$

Additionally, assuming that the probability of any given word is only dependent on its tag and that the probability of a POS tag depends only on the previous k POS tags, expression (4.2) can be further simplified into:

$$p(w_n | t_n)\ p(t_n | t_{n-k} \dots t_{n-1})\ p(w_1 \dots w_{n-1}, t_1 \dots t_{n-1}) \tag{4.3}$$

Then, applying the Bayes' Theorem recursively results in the expression:

$$p(w_1 w_2 \dots w_n, t_1 t_2 \dots t_n) = \prod_{i=1}^{n} p(w_i | t_i)\ p(t_i | t_{i-k} \dots t_{i-1}) \tag{4.4}$$

[5] `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`, Accessed on July 22, 2015

Which is the definition of a Markov Model of k-th order. For the sake of simplicity and since most POS taggers, including the TreeTagger, rely mostly on first-order and second-order Markov models (bigrams and trigrams respectively) [139], the following explanation will only consider a second-order Markov model ($k = 2$). This restriction corresponds to the assumption that the probability of any given POS tag depends only on the two previous tags. With this assumption, expressions (4.1) and (4.3) can be combined into:

$$p(w_1 w_2 \dots w_n, t_1 t_2 \dots t_n) = p(w_n|t_n)\ p(t_n|t_{n-2} t_{n-1})\ p(w_1 \dots w_{n-1}, t_1 \dots t_{n-1}) \tag{4.5}$$

Which is equivalent to:

$$p(w_1 w_2 \dots w_n, t_1 t_2 \dots t_n) = \prod_{i=1}^{n} p(w_i|t_i)\ p(t_i|t_{i-2} t_{i-1}) \tag{4.6}$$

According to Helmut Schmid, creator of TreeTagger [99], most N-gram taggers use the following formula, based on the *Maximum Likelihood Estimation* (MLE) principle, for estimating the transition probabilities $p(t_i|t_{i-2} t_{i-1})$:

$$p(t_i|t_{i-2} t_{i-1}) = \frac{F(t_{i-2} t_{i-1} t_i)}{F(t_{i-2} t_{i-1})} \tag{4.7}$$

Where $F(t_{i-2} t_{i-1} t_i)$ is the frequency of the trigram $t_{i-2} t_{i-1} t_i$ and $F(t_{i-2} t_{i-1})$ that of the bigram $t_{i-2} t_{i-1}$. He states however that this method pose problems since most frequencies are small, therefore making the probability estimation less reliable. He further adds that this method does not account for "ungrammaticalities" whereas a robust tagger should.

Instead of using formula (4.7), the TreeTagger uses binary decision trees for estimating transition probabilities. The probability of any given trigram is estimated by following the path down the tree until a leaf node is reached. Details on how the tree is built and refined can be found in [99]. Additionally, to see how the software was improved after its initial release refer to [139].

4.1.4 MaltParser

The chosen tool for finding the dependency relationships between the words of a sentence was MaltParser,[6] because it was already tested by Vilares et al. [43], yielding satisfactory results. Full documentation on its theoretical foundations can be found in [108], whereas a simple introduction is provided in the subsection Dependency Grammars of Section 2.3.3.

MaltParser is a Java implementation of a system for data-driven dependency parsing. For this thesis' purposes, it suffices to know that the program had to be trained with a correctly

[6] `http://www.maltparser.org/`, Accessed on August 03, 2015

labeled corpus, in the CoNLL (Conference of Natural Language Learning) format. In this format, a sentence is represented by a series of rows corresponding to its words and some features associated with them, represented by tab-separated columns. Refer to Appendices E and F to see a sentence written in the CoNLL format with Ancora Dependencies and a table presenting the meaning of each CoNLL column.

Just like in the study by Vilares et al. [43], the chosen corpus for training MaltParser was AnCora. Information on the methodology for building the corpus, and its annotation schemes can be found in [107]. In a few words, AnCora is a corpus composed of 500.000 words in Catalan and 500.000 words in Spanish, taken from different press sources (Spanish EFE news agency and "*El Peródico*" newspaper), and from a Spanish balanced corpus (Lexesp). The Spanish corpus possesses morphosyntactic, chunk and syntax information, along with thematic roles and noun senses.

Finally, for building a dependency graph for a given sentence or group of sentences MaltParser first receives the input, written in the CoNLL format with the ID, FORM, LEMMA, CPOSTAG and POSTAG columns filled. Its output is the same file containing the input sentences along with the HEAD column, corresponding to the ID of the word that is the head of each token, and the DEPREL column, corresponding to the Ancora-typed dependency relationship of each node to its parent.

4.1.5 Natural Language Toolkit (NLTK)

The Python Natural Language Toolkit is a Python module for performing various NLP-related tasks. It was created in 2001 as a part of a computational linguistics course in the Department of Computer and Information Science at the University of Pennsylvania. Its original creators are Steven Bird, Edward Klein and Edward Loper; today NLTK counts with more than a dozen contributors. The primary resource for learning how to use this module is the book "Natural Language Processing with Python," written by its creators [39].

This module was mainly used for dealing with Maltparser's output; the CoNLL file containing the dependencies returned by the parser is processed and its data is transformed into a Dependency Graph which is used for applying the rules presented in Section 3.6.2. The module, however, has several other sub-modules for almost everything related to Natural Language Processing, including tokenization, stemming, POS-tagging, chunking, parsing, semantic interpretation and applications, among others. Further, NLTK is constantly being updated and improved; it even has an active development branch developing sentiment analysis features (`https://github.com/nltk/nltk/tree/sentiment` Accessed August 03, 2015).

4.1.6 Django

Django[7] is an open source Python Web framework that comes with almost every functionality needed for Web development out of the box (user authentication, content administration, and a template language, among others).

[7] `https://www.djangoproject.com/` Accessed August 03, 2015

The reason for choosing this framework was that it was created for Python, therefore integrating it with the rest of the OM application was not difficult. One might ask why another Python Web framework like Flask[8] was not used, and the reason for that is simply that Django seemed more complete and potentially useful for larger applications. If the Web framework were to be chosen again today, probably Meteor[9] would be the best choice.

4.2 Data Extraction

The data extraction process began on October the 3rd, 2014 and ended on March the 4th 2015. In this period, a total of 56.773 tweets containing the keyword `falabella` either in the text field or the screen name, were gathered. It is not possible to ensure this amount corresponds to the real amount of tweets generated in this period, because several times during the extraction process the server where the script was running ended it, therefore causing the loss of all the tweets from termination to resumption.

This process was the only one executed on a different environment, because it had to run constantly and with as few interruptions as possible. Accordingly, a t2.micro instance of the Elastic Compute Cloud (EC2) service of the Amazon Web Services was used for this purpose.[10] Later, when the data was needed for further processing, a local copy of them was created. Obviously in a production context this process would have to be automated by means of an Extract-Transform-Load (ETL) process or similar tool.

Data: (k, u) such that k is a keyword and u the url elements to build the search URL.
Result: None. A database is populated in each loop.

```
1  build URL from k and u;
2  initialize H = {0,0,0,0,0} ;                    /* Insertion history */
3  while True do
4  |   fetch html from URL;
5  |   find tweets in html;
6  |   i = 0;
7  |   foreach tweet in tweets do
8  |   |   if tweet not in database then
9  |   |   |   store tweet in database;
10 |   |   |   i = i + 1;
11 |   |   end
12 |   end
13 |   remove first element of H;
14 |   insert i as last element of H;
15 |   sleep for 100/(average(H)+1) seconds;
16 end
```

Algorithm 4.1: Data Extraction Algorithm.

[8] `http://flask.pocoo.org/` Accessed August 03, 2015
[9] `https://www.meteor.com/` Accessed August 03, 2015
[10] `http://aws.amazon.com/ec2`, Accessed on July 20, 2015

The procedure for extracting data from Twitter is described in Algorithm 4.1. In line 1 the search URL is built from the provided keyword ("falabella"), and some URL elements from the Twitter search page. The final url to be queried corresponds to `https://twitter.com/search?f=realtime&q=falabella%20lang%3Aes&src=hash`. After the URL is built, a queue for storing the last 5 insertion amounts (how many new tweets were found in the last query) is initialized in line 2. After initialization, the *while* loop begins without any terminating condition. In line 4, the html of the previously mentioned URL is fetched and then, in line 5, it is parsed. Despite the fact that a plethora of third-party tools for parsing html already exist,[11,12] the chosen implementation for extracting tweets from the html was a Python *regular expression* (regex), presented below:

```
<small class="time">.*?<a\shref.*?/(.*?)/status/(.*?)".*?data-time="(.*?)".*?
<p\sclass="js-tweet-text\stweet-text"\slang="es".*?"0">(.*?)</p>
```

What this regex does is to append every pattern between parenthesis (capturing group) to a list, each time the whole expression matches. Since the URL mentioned before only displays the latest 20 tweets, the whole expression would have 20 matches in every loop. However, as it was mentioned earlier in the Data Acquisition subsection of section 2.1.2, it is very likely that the webpage's structure changed since this script was last run, meaning the regex would have to be rewritten in order to account for the new structure.

The most relevant pattern of the presented expression corresponds to "`.*?`". What this pattern does is literally to match anything. In a regex expression the dot symbol (`.`) corresponds to *any character*, the asterisk (`*`) is a quantifier that signifies "match the previous expression 0 or more times," and the question mark (`?`) is another quantifier signifying "match the previous expression 0 or 1 time," therefore `.*?` means "any character 0 or more times, 0 or 1 time."[13] As a result, the previous regex would extract the user's screen name with the first capturing group, the tweet id with the second, the Unix timestamp with the third, and the status (tweet text) with the fourth.

Continuing with the algorithm's explanation, after the tweets are extracted, the counter for new tweets is initialized in line 6. From line 7 to 12 every extracted tweet is compared with the latest tweets stored in the MongoDB database. If the tweet is not present, it is stored and the counter is increased by one, if the tweet is present, nothing is done. Later, in line 13, the first element of the insertion history data structure is removed (dequeue operation), while the new insertion counter is inserted (enqueue operation). Finally, the script sleeps for an amount of time inversely proportional to the amount of new tweets extracted. It is worth noting that the parameters for calculating the sleep time were determined empirically and by no means represent the optimal wait time between requests.

[11]Beautiful Soup (`http://www.crummy.com/software/BeautifulSoup/`, Accessed on July 21, 2015)

[12]Scrapy (`http://scrapy.org/` Accessed on July 21, 2015)

[13]Visit `https://www.debuggex.com/cheatsheet/regex/python` for a quick reference on regular expressions (Accessed on July 21, 2015)

4.3 Preprocessing

The Preprocessing Module of the platform presented in Section 3.5 was finally implemented as a package containing the following classes: Cleaner, Corrector, Sentence Segmenter, Tokenizer and Tagger. Additionally, the functionality of these classes was made available through a package API. In this section these classes will be presented.

4.3.1 Cleaner

The Cleaner class is the one charged with cleaning tweets after they have been extracted by the Data Extraction module. Most of the Cleaner's tasks involve replacing or deleting unwanted content from each tweet, such as URLs, quotes, incorrect punctuation (at the beginning of the tweet for instance), and dashes, among others. For this analysis emoticons were also removed.

Removal Tasks

Most of these tasks were defined empirically by observing tweets and the appearance of repeated unwanted text. Python has a built-in function for replacing a text matched with a regular expression by any other regular expression, `re.sub()`. In the case of the removal tasks, any text that matches any of the regular expressions presented in Table 4.1 is replaced by an empty string (deleted).

Note that some expressions begin with a "backslash u" (\u). These represent unicode characters and those regular expressions that contain them must be declared differently, which is why they are presented in different lines.

Text Type	**Regex**

URL	`(http|https)://.*?(\s|;|$)`
	`(http|https)://.*?(\u2026)`
	`(http|https)://.*? `

Picture URL	`pic\.twitter\.com/.*?(\s|$)`
Undetermined Symbols	`&#.*?;` `&#39;`

Quotes	`&quot`
	`\u201c`
	`\u201d`
	`" (?#double quote character)`
	`' (?#apostrophe)`

Dash & Underscore	`-` `_`
Colon	`\:(\s|$)`
Happy Emoticons	`[:=x;](\)|D|3|P|p|\*)+(\s?|$)`
Sad Emoticons	`[:=x;](\(\)+(\s?|$)`

Table 4.1: Regular Expressions used for deleting unwanted text.

Replacement Tasks

In some occasions, instead of deleting the text it is replaced by another substring. Such is the case for ampersands, thousands separators, written laughs and spaces. Table 4.2 presents the regular expressions for matching these types of text and the text by which they are replaced.

Text Type	Regex	Replacement
Ampersand (&)	`&` `&+`	`&` `&`
Thousands Separator	`(?P<before>\d+)\.(?P<after>\d+)`	`\g<before>\g<after>`
Written Laughs	`(\s|^)(a*j*jaj*a*j*)+([^a-zA-Z]|$)`	`jaja`

Spacing	`\s*\W+$ (?#spacing before punctuation mark)`
	`\s{2,} (?#multiple spaces (2 or more))`
	`@\s (?#space after @ symbol)`
		Empty String
		Single Space
		`@`

Table 4.2: Regular Expressions used for replacing text.

Both the "Thousands Separators" and "Written Laughs" regular expressions can be improved; the former only matches those numbers that contain one separator,[14] whereas the latter only detects laughs that are compactly written (with no spaces) and composed only by the letters "j" and "a". The first could be improved to match numbers that contain an arbitrary amount of separators and the second to detect more kinds of laughs written in Spanish, such as "ja ja ja," "jejejeje," "jsakjskajskjak" and all their variants.

The tasks of removing and replacing are executed sequentially for each tweet so the pseudo-code implementation will not be presented.

4.3.2 Corrector

After the tweets are cleaned the Corrector is executed. This class has three main tasks: to expand abbreviations, to underscore composite expressions, and to underscore composite intensifiers. Both composite expressions and composite intensifiers are simply a combination of words that are always used together to depict the same meaning independently of context. One of the most frequent composite expressions is "Sin embargo" (however), whereas one of the most frequent composite intensifiers is "lo más" ("the most").

The procedure for doing this is checking whether any of the abbreviations, composite expressions, or composite intensifiers located in a lexicon are present in the analyzed tweet. In case they are, they are replaced by their extended or underscored version depending on the case. Both composite expression and composite intensifier lexicons were provided by Maite Taboada [78], while the abbreviation lexicon was created by making a simple word-frequency analysis of 50.000 tweets, and extracting high-frequency, non-stopword words of length 5 or less (refer to Appendix G to see some of these abbreviations). The simple algorithm for expanding abbreviations, which is analogous to the one for underscoring composite expressions and intensifiers, is presented in Algorithm 4.2.

[14]Not alike English, thousand separators are represented by dots instead of commas in Spanish.

```
   Data: text: The text to be processed
   Result: text′: The text with every found abbreviation expanded
1  lexicon ← load abbreviations lexicon;
2  foreach abbreviation in lexicon do
3  |  if abbreviation in text then
4  |  |  text′ ← replace abbreviation by expanded_abbreviation
5  |  end
6  end
7  return text′
```

Algorithm 4.2: Abbreviation Expansion.

In line 1 of the algorithm, the lexicon containing both the abbreviations and their corresponding expanded version is loaded. The lexicon variable is a list of lists where each list contains the two previously mentioned elements. Then for each tweet, the whole lexicon is traversed in search for abbreviations. Clearly, this is not the optimal algorithm, because for most texts only a few, if any, abbreviations will be found. The optimal would be to look for an abbreviation only after a candidate abbreviation has been found in the text. In the following lines of code each found abbreviation is replaced by its expanded form and finally the text containing only expanded abbreviations is returned.

4.3.3 Sentence Segmenter

After being corrected, the sentences contained in each tweet are segmented by the Sentence Segmenter. This means that if a tweet contains a punctuation mark (“.”, “!”, “?”), the sentence string is transformed into a list that contains both the string before and after the mark. If there is more than one mark, the tweet is separated accordingly so each different sentence can be processed independently. Sadly, there was not enough time to implement the feature to recognize abbreviations denoted by a dot, so such abbreviations will be incorrectly considered as the end of a sentence. The observation of the tweets, however, has empirically shown that these type of abbreviations are not often, if ever, used. The process for segmenting a tweet is presented in Algorithm 4.3.

This algorithm is complicated and poorly built at best, since it was created experimentally. The reason for keeping it in the final implementation was that it worked for the most common scenarios, and instead of improving it, time was spent in developing the platform further.

4.3.4 Tokenizer

Once the tweet is segmented, each sentence is tokenized. The Tokenizer class was built as a simple wrapper around the “twokenize.py” script created by Brendan O’Connor (`https://github.com/brendano/tweetmotif/blob/master/twokenize.py`). This script is part of a larger application called TweetMotif [140]. Briefly, TweetMotif is useful for detecting rumors, uncovering scams, summarizing sentiment and tracking political protests in real time. Its tokenization step is based on regular expressions and is custom-tailored for Twitter messages, meaning it can deal with hashtags, replies, abbreviations, strings of punctuation,

```
   Data: tweet: The tweet to be processed
   Result: sentences: A list with every sentence contained in the tweet
 1 sentences ← initialize empty list;
 2 punctuation_marks ← initialize a list containing punctuation marks;
 3 current_sentence ← initialize empty string;
 4 character_location ← 0;
 5 foreach character in tweet do
 6     if character not in punctuation_marks then
 7         current_sentence ← concatenate character to current_sentence;
 8         if character_location == len(tweet)-1 then
 9             append current_sentence to sentences;
10         end
11     else
12         /* if at the end of the tweet                                    */
13         if character_location == len(tweet)-1 then
14             if current_sentence ends with punctuation_mark then
15                 concatenate punctuation_mark with current_sentence;
16             else
17                 concatenate character with current_sentence;
18             end
19             append current_sentence to sentences;
20             break;
21         else if next_character is punctuation_mark then
22             concatenate character with current_sentence;
23         else if current_character is punctuation_mark and next_character is not
           punctuation_mark then
24             concatenate character with current_sentence;
25             append current_sentence to sentences;
26             current_sentence ← empty string;
27         end
28     end
29     character_location ← character_location + 1
30 end
31 return sentences
```

Algorithm 4.3: Sentence Segmenting.

emoticons and unicode glyphs.

4.3.5 Tagger

Finally, after the tokenizing step, each sentence is POS-tagged. The chosen software for doing so was the TreeTagger (Refer to Section 4.1.3). The Tagger class was created as a wrapper around the "treetagger.py" script created by Mirko Otto (`https://github.com/miotto/treetagger-python`), which is a Python module for interfacing with the previously mentioned software. The script was slightly modified for making it compatible with the rest

of the implementation.

The final output of the preprocessing step is a list of sentences

$$\texttt{preprocessed\_tweet} = [\texttt{sentence}_1, \texttt{sentence}_2, \ \ldots, \ \texttt{sentence}_k]$$

where each sentence is, in turn, a list containing a triplet for each word composing it:

$$\texttt{sentence}_i = [(\texttt{word}_{i1}, \texttt{POS-tag}_{i1}, \texttt{lemma}_{i1}), \ \ldots, \ (\texttt{word}_{in}, \texttt{POS-tag}_{in}, \texttt{lemma}_{in})]$$

Later, this structure will be exploited for building the dependencies between words and applying the rules mentioned in Section 3.6.2.

4.4 Polarity Classification

In this section, the process for obtaining the polarity for a preprocessed tweet will be described. To understand the ideas behind what will be explained refer to Section 3.6.

4.4.1 Obtaining the Dependencies

Taking off from the final output described in Section 4.3.5 (POS-tagged tweets), the next step is to obtain the dependencies for each word composing each sentence of a tweet. To achieve this, first it is necessary to convert the previously mentioned output into a specific format so it can be fed to NLTK, which will handle the interfacing tasks with MaltParser. The script for performing this transformation used in this thesis was created by Ekatherina Ovchinnikova and can be found in `https://github.com/eovchinn/ADP-pipeline/blob/master/pipelines/Spanish/Scripts/to_malt.py` (Accessed August 04, 2015). Basically, what the script does is to simplify the POS tags returned by the TreeTagger.

After being formatted, the input is passed to the MaltParser class of the "parse" module of NLTK. This class interfaces with MaltParser by transforming the Python data structure returned by the previously mentioned script into a plain text file, which is then fed to MaltParser through the standard input. Later, MaltParser returns the same, modified file, which is read by the module and transformed into a dependency graph (or dependency tree, given its properties).

A dependency graph in this context is, simply put, the translation of the CoNLL file into a format that can be handled in Python. Each node of the graph contains the following fields:

address: Corresponding to the ID column of the CoNLL format; it represents the position of the token in the sentence (ID = 0 for the root of the dependency tree).

word: The token itself. It corresponds to the FORM column of the CoNLL format.

lemma: The lemma of the word.

ctag: The coarse-grained POS-tag of the token, or the CPOSTAG column of the CoNLL format.

tag: The fine-grained POS-tag of the token, or the POSTAG column of the CoNLL format. In the current implementation this field is always equal to the ctag field.

feats: Corresponds to the FEATS column of the CoNLL format. This field is not used in the current implementation.

head: Is analogous to the HEAD column. It represents the *address* of the node to which the current node is related.

rel: The dependency relationship that the current node holds with its parent node. Equivalent to the DEPREL column of the CoNLL format.

deps: Contains a list of the addresses of the current node's children. This field has no equivalent in the CoNLL file and is created by NLTK.

Refer to Appendix H to see the example of a sentence represented as a NLTK dependency graph.

In this phase, just before NLTK transforms the input file into a dependency tree, code was implemented for adding the artificial nodes representing adversative clauses. If any of the words "*pero*," "*mientras*," or "*sin_embargo*" is found, then a line is added at the end of the CoNLL file. The difference with a node representing a word is that this artificial node is represented by the string "[]" (in both the *word* and *lemma* fields), and its tag is of the form `art_adversative:restrictive@4` where `restrictive` represents the type of adversative clause (it could also be `excluding`) and the last number represents the *address* of the adversative word. Additionally, its parent is always the root of the tree, and every other node that was a child of the root becomes a child of this artificial node. Finally, the same process is applied for the words "*sino*," and "*sino_que*" with the difference that they represent an excluding adversative clause instead of a restrictive one.

After obtaining this data structure for each sentence, nodes are tagged with their intrinsic polarity, stored in the field *sent_orig*, in addition to creating a new field for storing the modified polarity in the propagation process. To obtain the polarity for each word, each *lemma* field is compared with the words of a sentiment lexicon created by Maite Taboada et al. [78], from which the polarities are extracted. The lexicon is separated in 5 files, each containing adjectives, adverbs, nouns, verbs, and intensifiers. The format for each file is two tab-separated columns, the first being the word lemma, and the second its polarity p ranging from -5 to 5 ($p \in \mathbb{Z}$). The only exception to this rule are the intensifiers, since they do not possess a polarity number but an intensification value i ranging from -3 to 1 ($i \in \mathbb{Q}$). Additionally, the field *intensified* is created with the default value 0, to be later used.

Since the dependency graph has the same properties as a tree, it can be subdivided into levels. The next step of the process is to label each node with the level it belongs to, in order to facilitate the polarity propagation later. So, up to this point each node has 4 new fields:

sent_orig: The original polarity coming from the lexicon.

sent: A field that will be used to store the modified polarity during the polarity propagation process.

intensified: The intensification value of each word.

level: Level of the tree to which the node pertains.

The final step is creating a map that links each level to the nodes it contains, which will later allow the propagation process to traverse the tree in a convenient way. With this, it is now possible to get to the next stage which is the application of the heuristics for acknowledging the negations, intensifications, and adversative clauses.

4.4.2 Application of the Rules

In order to apply the rules, the algorithm traverses each level of the tree, from the bottom to the top, and then each node of each level. Algorithm 4.4 depicts how the process is executed.

```
   Data: dependency_graph: The sentence represented as a dependency graph
   Result: dependency_graph: The dependency graph with the propagated polarity
 1 levels ← obtain levels from dependency_graph;
 2 foreach level in levels do
 3 |   foreach node in level do
 4 |   |   apply intensification rules;
 5 |   |   apply negation rules;
 6 |   |   apply adversative clause rules;
 7 |   |   update current_node_polarity;
 8 |   |   head_polarity ← current_node_polarity;
 9 |   end
10 end
11 return dependency_graph
```

Algorithm 4.4: Polarity Classification.

In the following subsections the implementation for each kind of rules will be explained.

Intensification

The first set of rules applied are those related to intensification. Recall the intensification rule defined in the Intensification subsection of Section 3.6.2: If an adverb is labeled as being a non-head determiner (SPEC, ESPEC), an adverbial phrase (sadv), or an adjunct (CC), the adverb is considered as an intensifier and its head is defined as the scope of the intensification. The algorithm for applying this rule is straightforward and is presented in Algorithm 4.5.

The intensification_strength rule just gets the intensification value from the lexicon.

if $node_{tag}$ *is adverb and* $node_{rel}$ *is* {*spec or espec or cc or sadv*} **then**
  | $\text{head}_{intensified}$ += intensification_strength(node_{word})
end

Algorithm 4.5: Application of the Intensification Rule

Negation

The second set of rules are those related to negation. Recall from the Negation subsection of section 3.6.2 that there are 4 negation rules: the subjective parent rule, the subject complement – direct object rule, the adjunct rule, and the default rule. All of these rules but the first require the algorithm to know the siblings of each node.

The general way negation rules are applied is depicted in Algorithm 4.6

else if $node_{word}$ *is* {*no or nunca or sin*} **then**
  | apply subjective parent rule;
  | apply subject complement – direct object rule;
  | apply the adjunct rule;
  | apply the default rule;
end

Algorithm 4.6: Application of the Negation Rules – Summarized.

Subjective Parent: This rule is applied whenever the parent of a given negation node has a prior polarity associated.

Subject Complement – Direct Object Rule: When the parent is not subjective, i.e. it has a polarity of 0, then the algorithm checks this rule.

Adjunct Rule: When none of the previous rules is applied, then the algorithm checks whether a sibling (node at the same level), of the current node is an Adjunct. In case there is one or more, then the rule is applied for the first occurrence.

Default Rule: When none of the previous rules apply, then the scope of negation is considered to be every sibling, hence the negation is applied to all of them.

Algorithm 4.7 presents how these rules are applied. Just like Algorithm 4.3 for segmenting sentences, this algorithm could *greatly* benefit from a refactoring session for avoiding duplicate code and improving overall readability.

Adversative Clauses

The final set of rules are those related to the restrictive adversative clauses defined by the conjunctions *pero* (but), *mientras* (while) and *sin embargo* (however), and to the exclusive adversative clauses defined by *sino* (bur rather), and *sino que* (but also).

```
   else if node_word is {no or nunca or sin} then
       /* Subjective Parent Rule                                          */
 1     if head_sent_orig > 0 then
 2         head_sent −= negation_strength
 3     else if head_sent_orig < 0 then
 4         head_sent += negation_strength
 5     /* Subject Complement -- Direct Object Rule                        */
 6     else if head_sent_orig == 0 then
 7         visited_siblings = [ ];
 8         foreach sibling in siblings do
 9             if sibling_rel is {atr or cd} then
10                 if sibling_sent > 0 then
11                     sibling_sent−= negation_strength
12                 else if sibling_sent < 0 then
13                     sibling_sent+= negation_strength
14                 end
15                 append sibling to visited_siblings;
16             /* Adjunct Rule                                            */
17             else if sibling_rel is cc then
18                 if cc not in visited_siblings then
19                     if sibling_sent > 0 then
20                         sibling_sent−= negation_strength
21                     else if sibling_sent < 0 then
22                         sibling_sent+= negation_strength
23                     end
24                 end
25                 append sibling to visited_siblings;
26             end
27         end
28         /* Default Rule                                                */
29         if visited_siblings == [ ] then
30             foreach sibling in siblings do
31                 if sibling_sent > 0 then
32                     sibling_sent−= negation_strength
33                 else if sibling_sent < 0 then
34                     sibling_sent+= negation_strength
35                 end
36             end
37         end
38     end
   end
```

Algorithm 4.7: Application of the Negation Rules – Complete.

The general rule is that whenever a restrictive node is found, then the polarity of the main clause, corresponding to the words that come before the conjunction, is attenuated

while the adversative clause – words coming after the conjunction–, is intensified. For more information on this refer to subsection Adversative Clauses of Section 3.6.2.

The process for applying these rules is described in Algorithm 4.8.

```
else if node_rel is art_rel_adversative then
    get adversation_type from node_tag;
    get conjunction_address from node_tag;
    define weight_main_clause depending on adversation_type;
    define weight_adversative_clause depending on adversation_type;
    main_clause_polarity ← 0;
    adversative_clause_polarity ← 0;
    foreach child in node_deps do
        if child_address < conjunction_address then
            main_clause_polarity += child_sent;
        else if child_address > conjunction_address then
            adversative_clause_polarity += child_sent;
        end
    end
    node_sent ← (weight_main_clause * main_clause_polarity) +
    (weight_adversative_clause * adversative_clause_polarity);
end
```

Algorithm 4.8: Application of the Adversative Clause Rules.

Current Node Polarity

Finally, after all the rules have been applied, the algorithm checks if the current node has been intensified in a previous iteration; in case it has been then the current polarity is updated accordingly. The last step is transmitting the current polarity to the parent node. The algorithm for executing these steps presented in Algorithm 4.9. The whole algorithm for applying the rules to the dependency graph is presented in appendix I .

```
if node_intensified != 0 and node_sent_orig == 0 then
    node_sent* = (1 + node_intensified);
else if node_intensified != 0 and node_sent_orig != 0 then
    node_sent+ = (node_sent_orig * (1 + node_intensified));
end
head_sent ← node_sent
```

Algorithm 4.9: Application of the Intensification Rules.

The final result of this algorithm is the dependency graph with updated polarities for each node. From this it follows that the polarity of the root node corresponds to the polarity of the whole sentence, after having considered intensification, negation and adversative clauses.

4.4.3 Overall Tweet Polarity

The implementation presented until now is for classifying sentences, so there is still a step needed for classifying tweets as a whole. In this final step, the sum of the polarity from each sentence composing a tweet is calculated therefore obtaining the polarity for the whole tweet.

One might wonder why the chosen method for aggregating the sentences composing a tweet was the simple sum, and the answer is based on the assumption that tweets are short enough to contain only one opinion. Different would be the case for reviews, for instance, where the reviewer has a larger space for expressing himself and often opines about more than one entity or aspect of that entity. In such case, a more fine-grained approach, such as aspect-based Opinion Mining, would be more appropriate.

Another consideration to bear in mind relates to the fact that often, the order in which opinions are expressed greatly influences the overall polarity of an utterance. Pang and Lee [49] give the following example to illustrate this claim:

(4.1) This film should be *brilliant*. It sounds like a *great* plot, the actors are *first grade*, and the supporting cast is *good* as well, and Stallone is attempting to deliver a *good* performance. However, it can't hold up.

Where the polarity of the whole review is negative, despite the fact of having mostly positive words. If a tweet fed to the classification algorithm were written in a similar fashion, it would be erroneously classified as positive.

To wrap up, the way the overall polarity is calculated from each individual sentence is by adding up their polarities, even if this method doesn't capture phenomena associated to the order in which sentences are placed within a tweet, such as the one mentioned earlier. In another context, such as review analysis, the aggregation method should be reconsidered.

4.5 Visualization

The final element of the Opinion Mining platform is its visualization module, implemented in Django (See Section 4.1.6). The main task of this module is to display the data obtained by the previous implementation in a user-friendly way. Keep in mind, however, that the final result of the visualization module is just a prototype. There are other senior students working on theses concerning how to best display the results given by the backend presented up to this point, and how to encapsulate it in a service.

Django is structured in a Model-View-Controller (MVC) fashion, which means that there are three distinct conceptual layers dealing with the tasks of obtaining and modeling the data from persistent storage (Model), applying business rules to them (Controller), and displaying them to the user (View). Django nomenclature is a little different from the traditional one; traditional "Controllers" are called "Views," and traditional "Views" are called "Templates." The "Models" have the same meaning.

The typical data flow when interacting with the Django website is the following: The user requests a URL which is matched by a regular expression in a file called "urls.py" and translated into a function defined in the "views.py" file. Usually, the function in the views file will contain a call to another function in the models layer for requesting the corresponding data to the database. Finally, the views function will render the data in a specific template.

Each layer is briefly described below.

Model: The Model layer was created as a simple interface with the database, where python functions are translated into MongoDB queries with the use of the pymongo module.[15] The interface contains the functions necessary to display the site that was presented in Section 3.7.2, namely, for obtaining tweets belonging to a certain polarity range, for calculating frequencies, and for searching for keywords. For example, the function that finds those tweets that are very positive (arbitrarily defined as those having a polarity greater than 8) executes the following query:

(4.2) `tweets.falabella.find({''malt_polarity'': {$gte: 8}})`

Which translates to: in the "falabella" collection of the "tweets" database, find those documents with the field "malt_polarity" greater than or equal to 8 (`$gte: 8`).

The results of this query are then returned to the calling function which, in turn, formats the data for later passing it to the caller belonging to the Views layer.

Views: There are 4 functions defined in the views: One that maps to the landing site, one that maps each polarity label (very negative, negative, neutral, positive, very positive) to its corresponding Web page, one that requests and displays search results and, finally, one for displaying the two statistical charts. Again, these functionalities have demonstration purposes only, and would have to be significantly improved before attempting to release or commercialize the application.

Templates: The idea behind templates is to avoid duplicating html code. For the Web page prototype 6 templates were created:

- **base.html**: Contains the `<head>` information and basic structure for each other Web page. Among others, this template loads the charting tools from Google[16] and the Boostrap front-end framework.[17]
- **navbar.html**: Extends the base.html template and contains the data for displaying the navigation bar on top of each Web Page.
- **table_frame.html**: Extends the navbar.html template and defines the structure for displaying the tweets and their corresponding polarity.
- **landing.html**: Extends the navbar.html template and contains a welcome message to be displayed in the home page.

[15] `https://api.mongodb.org/python/current/` Accessed on August 07, 2015
[16] `https://developers.google.com/chart/`, Accessed on August 07, 2015
[17] `http://getbootstrap.com/`, Accessed on August 07, 2015

– **search_results.html**: Extends the table_frame.html template and displays the data generated by a search term, or by the selection of a polarity tag by the user.
– **stats.html**: Extends the navbar.html template and displays a frequency bar chart and a pie chart.

For more information on how Django sites are built, refer to its documentation (`https://docs.djangoproject.com/en/1.8/`, Accessed on August 07, 2015), and its tutorial (`https://docs.djangoproject.com/en/1.8/intro/tutorial01/`, Accessed on August 07, 2015).

4.6 Implementation Architecture

To summarize, this section describes how the modules described until now interact with each other, similar to what was presented in Section 3.2 but in a lower level of abstraction. Figure 4.1 presents a simplified diagram of how the application modules interact with each other.

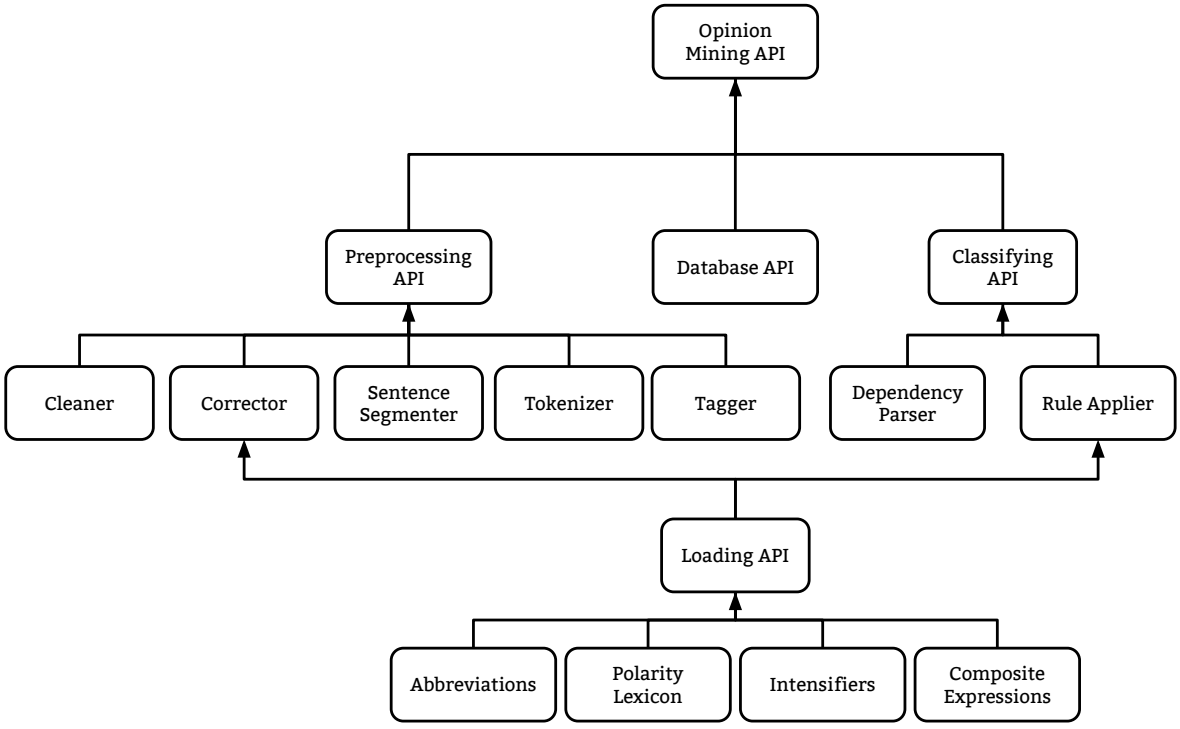


Figure 4.1: Implementation Architecture.

The Opinion Mining API corresponds to the highest level of abstraction of the whole application. It offers the required functionality for obtaining raw tweets saved in the database – previously obtained by the Data Extraction Module, which not presented in the diagram –, classifying them, and saving them back in the Processed Tweets database through the Database API. The Preprocessing API exposes the functionality of the classes for preprocessing the tweets, while the Classifying API exposes that for obtaining the polarity of a preproces-

sed tweet. The Loading API simply offers a layer of abstraction for dealing with plain text files containing the abbreviations, polarity lexicon, intensifiers and composite expressions.

There were some relationships omitted in the diagram to keep it simple and easily understandable; these are the following: the Tagger class interfaces with the TreeTagger software through a module mentioned in Section 4.3.5, which is not presented in the diagram, and the Dependency Parser class interfaces with the MaltParser software through a NLTK implementation that has another class hierarchy which is even more complex than the one presented in the diagram. Additionally, the Visualization Module is not depicted either. This because the front-end architecture was thought as an independent implementation, indeed, the only way the Opinion Mining implementation, exhibited in Figure 4.1, communicates with the visualization module is through the database. This loose coupling allows for making changes in any module without greatly affecting the others. For example, the current visualization prototype could be totally changed and every functionality of the Opinion Mining engine would continue to function normally, or if the algorithm for calculating the polarity of a tweet was replaced by, say, a machine learning implementation, the Web page prototype would continue to work normally, provided the input and output structures remained unchanged.

The previous statement is also true for any of the other modules composing the OM engine; if the TreeTagger was changed by a Brill Tagger implementation, while maintaining the way the tagger outputs its results, the rest of the software would not "know" and hence would not be affected by the change. In Code Complete [141], Steve McConnell devotes a whole section to point out the positive aspects of having loosely coupled classes and routines, and insists on this fact throughout the rest of the book.

Finally, one could argue that the correct way to represent the software would be by creating a Unified Model Language (UML) diagram, but the point of presenting it as in Figure 4.1 was to keep it simple and understandable for any reader without a Computer Science background.

Chapter 5

Validation and Case Study

In this chapter, the Opinion Mining algorithms used for obtaining the polarity of tweets will be validated, and later the results of applying them to real retail data will be presented. The validation process will give an overall quality measure of the algorithms, allowing to grasp how well they perform while classifying Spanish tweets with the system presented in previous chapters. Next, with the quality measure in mind, it will be possible to determine whether the results obtained in the Case Study are believable and whether they can be incorporated in the retail company's decision-making process.

This chapter is structured as follows: First the validation process will be described, beginning by the explanation of the metrics used for validation, the presentation of the corpus used as a ground truth, the description of the process itself, and finishing with the presentation of the results. Second, a case study concerning the Chilean retail company Falabella will be presented. The analysis will be separated in two subsections, the first will describe some important aspects of the dataset containing tweets mentioning Falabella unrelated to polarity or opinions, whereas the second will be devoted solely to polarity-related metrics.

5.1 Validation

In this section, the Opinion Mining pipeline described in the previous chapters will be validated. In order to do so, a corpus with known data, the TASS corpus, will be processed with the OM system and its results will be compared with the real labels. There are many metrics for describing how well a classification process performs, some of which will be presented in the next subsection, 5.1.1.

The following subsections will characterize the TASS corpus (5.1.2), and describe the validation process and its results (5.1.3).

5.1.1 Evaluation Metrics

The most frequently-used and basic evaluation metrics for Information Retrieval are *precision* and *recall* [97]. A system that classifies input data into two possible sets, Positives and Negatives, can have its output presented in what is called a *confusion matrix*. An example of such matrix is presented in Table 5.1.

		Classified	
		Negative	**Positive**
Real	**Negative**	True Negatives (TN)	False Positives (FP)
	Positive	False Negatives (FN)	True Positives (TP)

Table 5.1: Confusion Matrix Example.

Table 5.1 shows that each value can be classified in four possible ways:

True Negative: A real negative value classified as such.

False Negative: A value that is classified as negative but is not.

True Positive: A real positive value classified as such.

False Positive: A value that is classified as positive but is not.

With this in mind, positive precision (P_{pos}) and recall (R_{pos}) are defined as follows (negative precision and recall are define analogously):

$$P_{pos} = \frac{TP}{TP + FP} \tag{5.1}$$

$$R_{pos} = \frac{TP}{TP + FN} \tag{5.2}$$

Precision is a measure that represents the proportion of correctly tagged values among the values with the same obtained tag, whereas recall shows the proportion of known values (real values) correctly tagged. The problem of using these metrics separately is that they trade off against each other, meaning one can obtain high recall at the cost of precision and vice-versa. This is why the another metric capable of representing both precision and recall at the same time – the F-measure –, is created and defined as the harmonic mean of them.

$$F\text{-}measure_{pos} = 2 \cdot \frac{P_{pos} \cdot R_{pos}}{P_{pos} + R_{pos}} \tag{5.3}$$

A perfect classifier would classify correctly every value, so the cells in the diagonal of the confusion matrix would be the only ones with values greater than 0, and the recall, precision and F-measure would be 1.

Another measure frequently used is the *accuracy*, defined as the proportion of correctly tagged values among the whole dataset. Again, a perfect classifier would correctly classify every value, hence having an accuracy of 1.

$$Accuracy = \frac{TP + TN}{TN + FN + TP + FN} \tag{5.4}$$

The final metric used to evaluate the performance of the OM system corresponds to the Kappa measure (κ), which is used for measuring the agreement between two or more observers of the same observed phenomenon [142, 143]. In this case the two "observers" correspond to the ground truth and the classifier. The κ statistic ranges from -1 to 1, where 1 means perfect agreement, 0 means that the results are given purely by chance and negative values represent systematic disagreement between the observers. Refer to appendix J to understand how to interpret κ. To define κ, consider a classifier h, a dataset of m examples and the set of possible classes L. Any cell of the confusion matrix resulting from the classification process can be represented as C_{ij} where i corresponds to the real class of the value and j to the class defined by the classifier. Additionally p_0 and p_C are defined below:

$$p_0 = \sum_{i \in L} \frac{C_{ii}}{m} \tag{5.5}$$

$$p_C = \sum_{i \in L} \Big(\sum_{j \in L} \frac{C_{ij}}{m} \cdot \sum_{j \in L} \frac{C_{ji}}{m} \Big) \tag{5.6}$$

Finally κ is calculated as:

$$\kappa = \frac{p_0 - p_C}{1 - p_C} \tag{5.7}$$

So, for the confusion matrix presented in Table 5.1, $L = \{P, N\}$ (positive and negative class), C_{PP} corresponds to the amount of true positives, C_{NN} to the amount of true negatives, C_{PN} to the false negatives, and C_{NP} to the false positives. Knowing this, calculating κ for a given dataset is easy. Finally, it is worth noting that all of these definitions are applicable to data having two or more classes ($|L| \geq 2$).

5.1.2 Validation Corpus Description

The selected corpus for validating the Opinion Mining algorithms presented in this thesis was the TASS corpus [144]. The corpus is composed of a training set, containing 7219 annotated tweets, and a test set containing 60798 non-annotated tweets, all in Spanish. The former, which was the one used for validation, is structured as a XML file, where each tweet entry contains the Tweet ID, user ID, and creation date. Additionally, each tweet is annotated with 1 of 5 polarity levels: strongly negative (`N+`), negative(`N`), neutral (`NEU`),

positive (P), strongly positive (P+), and containing no sentiment (NONE). Because Twitter's terms of service do not allow to redistribute tweets obtained through its API, the annotated corpus only contained the Tweet ID, without the text, so it was necessary to create a script to redownload the full tweets through the API. Additionally, some of the tweets contained in the corpus were removed at the time the script was run, so only 6.969 of the original 7.219 tweets could be retrieved.

Moreover, the tweets are tagged with other fields that were not used in this process. In the cases where applicable, the entities within the tweets were annotated with their respective polarity, along with the Agreement level of the sentiment within the context, and the topics to which each tweet relates. Some of these topics include politics, soccer, literature, and entertainment. Table 5.2 summarizes the relevant characteristics of the corpus and provides some additional information. The three user types presented in the table, correspond to journalists, politicians and famous people.

Attribute	**Value**
Tweets	6969
Topics	10
Tweets Language	Spanish
User Amount	154
User Types	3
Date Start	2011-12-02 T00:47:55
Date End	2012-04-10 T23:40:36

Table 5.2: TASS Corpus Characteristics.

Further, the distribution of topics is presented in Table 5.3. The Current Frequency corresponds to the amount of tweets pertaining to the corpus of 6.969 tweets, as opposed to the Original Frequency corresponding to that of the corpus of 7.219 tweets[1]. Obviously, a tweet can belong to more than one topic, therefore the sum of the difference for each topic frequency, does not equal the difference of tweets between the original corpus and the current corpus.

[1]There are some discrepancies between the values reported in [144], and the values obtained by using a database engine on the downloaded corpus for the entertainment, literature and politics topics, however they do not amount to more than 5 tweets per topic.

Topic	Original Frequency	Current Frequency
Economy (economía)	942	911
Entertainment (entretenimiento)	1.678	1.646
Films (cine)	245	239
Literature (literatura)	103	102
Music (música)	566	562
Other (otros)	2.337	2.243
Politics (política)	3.120	2.996
Soccer (fútbol)	252	247
Sports (deportes)	113	106
Technology (tecnología)	217	209

Table 5.3: TASS Topic Distribution.

Finally, the polarity distribution for both the original and the current corpus are presented in Table 5.4

Polarity Tag	Original Frequency	Current Frequency
`N+`	847	822
`N`	1.335	1.295
`NEU`	670	651
`NONE`	1.483	1.428
`P`	1.232	1.198
`P+`	1.652	1.575
Total	**7.219**	**6.969**

Table 5.4: TASS Polarity Distribution.

5.1.3 Validation Process and Results

The validation process basically consisted in tagging the whole TASS corpus with the OM algorithms presented earlier. This later allowed to compare the real tag with the tag obtained by the classifier. It is worth mentioning that the current implementation returns the polarity as a number, whereas the labels from the TASS corpus are categorical. This is why the polarity had to be transformed into the same tags as the ones present in the Corpus. The criteria for this transformation are the following, considering $Tweet_{pol}$ as the numeric polarity of a tweet:

$$Tweet_{pol} < -4 \Rightarrow \texttt{N+} \tag{5.8}$$

$$-4 \leq Tweet_{pol} < 0 \Rightarrow \texttt{N} \tag{5.9}$$

$$0 \leq Tweet_{pol} < 1 \Rightarrow \texttt{NEU} \tag{5.10}$$

$$1 \leq Tweet_{pol} < 4 \Rightarrow \texttt{P} \tag{5.11}$$

$$4 \leq Tweet_{pol} \Rightarrow \texttt{P+} \tag{5.12}$$

Rules from (5.8) to (5.12) where defined after testing various parameters for the limits. In the end these values yielded the best performance metrics. Furthermore, an additional set of rules was created for dissambiguating NEU tags, since a tweet initially tagged as such could either have both positive and negative words that cancel each other, in which case they would be correctly tagged, or they could have few or no polar words in which case they should be tagged as NONE. Consider F_P to be the frequency of positive words in a given tweet, F_N the frequency of negative words, and τ a given threshold, then:

$$(x \in \{F_P, F_N\} | x \leq \tau) \Rightarrow \texttt{NONE} \tag{5.13}$$
$$(x \in \{F_P, F_N\} | x \geq \tau) \Rightarrow \texttt{NEU} \tag{5.14}$$
$$F_P < \tau \leq F_N \Rightarrow \texttt{N} \tag{5.15}$$
$$F_N < \tau \leq F_P \Rightarrow \texttt{P} \tag{5.16}$$

The tested values for τ ranged from 0 to 5, with $\tau = 1$ yielding the best results.

With all this in mind, the confusion matrix with all the classes is presented in Table 5.5.

		Classified						
		N+	N	NEU	NONE	P	P+	**Total**
Real	N+	130	274	27	192	140	59	**822**
	N	101	408	34	359	308	85	**1.295**
	NEU	27	145	9	171	200	99	**651**
	NONE	4	143	2	929	296	54	**1.428**
	P	18	117	16	336	468	243	**1.198**
	P+	4	75	16	331	530	619	**1.575**
	Total	**51**	**1.395**	**104**	**2.318**	**2.821**	**280**	**6.969**

Table 5.5: Confusion Matrix for 5 Classes.

By observing the table, it is possible to see that the values in the diagonal (dark green) are those perfectly tagged, those close to the diagonal (light green) are those tagged with the more or less intensified version of the same tag, and those away from the diagonal are misclassified values (red). Since obtaining neutral tweets and tweets without sentiment was not a priority when building the current system, a misclassification concerning these tags will not be considered as serious as a positive tweet being classified as negative and vice-versa.

It is clear, however, that there is a problem with both the NEU and NONE classes; there are many negative and positive tweets tagged as NONE (NONE column), and many NEU and NONE tweets tagged as either positive or negative. The former can be associated to the fact that there are language constructs that are not captured by the heuristic rules for detecting the phenomena previously mentioned. To give a simple example, the following tweet was tagged as NONE:

(5.1) "La Generalitat dice que no tiene dinero para pagar funcionarios. En el @TelediarioInter 20:30 les contamos para qué sí tiene dinero."

(5.2) "The Generalitat says it doesn't have any money to pay the government officials. In @TelediarioInter 20:30 we tell you what they do have money for."

To be able to classify the tweet presented in (5.2), first the algorithm should be able to understand that there are common phrases, instead of just words, that can convey a negative meaning. Take "doesn't have any money," for instance, which can be considered as something negative in almost any context. The current platform implementation recognizes every token of that phrase as being neutral, whereas not having something vital, such as money or food, clearly is negative. The previous statement raises the question of how it is possible to tell the algorithm what is vital to have and what is not, under what circumstances, and for whom. This is where the need for semantic analysis becomes vital; if one were able to represent the knowledge that to pay the government officials it is necessary to have money, and hence not having money means it is not possible to pay them, which obviously is negative, the classifier would be considerably better. Another subtlety, even more difficult to capture, is the fact that saying "we tell you what they do have money for" is indirectly blaming the Generalitat for using the money inappropriately.

Another way of "teaching" the algorithm these kind of phenomena is to use a machine-learning-based approach, although it still would be limited to the information learned from the training process, so, in a different context, it would not know that it is negative for the children of Africa not having medicine, or for a company not being solvent. The best solution would be the one that gets closer to the human knowledge, by incorporating information such as the fact that not having something necessary is negative or not having something bad is positive.

The other problem mentioned earlier, `NEU` and `NONE` tweets getting tagged as positive or negative, can be associated to the fact that some words in some contexts do not bear an intrinsic polarity, or to the fact that both the TreeTagger and MaltParser do not classify the words perfectly. For instance, tweet (5.4) is tagged as having a polarity of 3 (`P`), whereas it is clear that the tweet is just stating a fact, and should have no polarity associated. The misclassification is produced because the TreeTagger considers the word "Plenos" (plenary sessions) as an adjective instead of a noun, which is its intended meaning, and the word "pleno" (full) in the adjectives dictionaries, bears an intrinsic polarity of 3.

(5.3) Hoy dos Plenos (@ Ayuntamiento de Málaga) [pic]: http://t.co/9jdWZC1H

(5.4) Today two plenary sessions (@ Ayuntamiento de Málaga) [pic]: http://t.co/9jdWZC1H

Furthermore, for calculating the performance metrics, the tags `N+` and `N` were merged into `N`, `NEU` and `NONE` into `NEU`, and `P` and `P+` into `P`, resulting in Table 5.6 equivalent to Table 5.5.

		Classified			
		N	NEU	P	**Total**
Real	N	913	612	592	**2117**
	NEU	319	1.111	649	**2.079**
	P	214	699	1.860	**2.773**
	Total	**1446**	**2442**	**3101**	**6969**

Table 5.6: Confusion Matrix for 3 Classes.

Simply by observing Table 5.6 it is possible to see that the classifier performs better for positive tweets than negative ones. Table 5.7 displays the validation metrics while considering the three classes N, NEU, and P. This table clearly shows that the algorithm is better at recognizing positive tweets than it is for classifying negative and neutral ones. Additionally, the overall Accuracy and F-measures are not high enough to be accepted in a serious production environment, even if the kappa measure shows a fair agreement between the classifier and the ground truth.

	N	NEU	P
Precision (%)	63.14	45.87	59.98
Recall (%)	43.13	53.44	67.08
F-measure (%)	51.25	49.37	63.33
Accuracy (%)	55.73		
κ	0.325		

Table 5.7: Performance Metrics for 3 Classes.

Additionally, results when ignoring both the NEU and NONE tags are presented in Tables 5.8 and 5.9

		Classified		
		N	P	**Total**
Real	N	913	592	**1505**
	NEU	319	649	**968**
	P	214	1860	**2.074**
	Total	**1446**	**3101**	**4547**

Table 5.8: Confusion Matrix for 2 Classes, ignoring NONE and NEU tags.

	N	P
Precision (%)	63.14	59.98
Recall (%)	60.66	89.68
F-measure (%)	61.88	71.88
Accuracy (%)	60.99	
κ	0.332	

Table 5.9: Performance Metrics for 2 Classes, ignoring NONE and NEU tags.

The effect of ignoring the NEU and NONE tags, besides the decrease in size of the dataset, is an increase in both the negative and positive recalls, the accuracy, the F-measures and the kappa measure. These measures are the ones to be considered in case the platform was used in its actual state.

Alternatively, assuming an earlier step in the process effectively filters away tweets not containing any polarity and neutral tweets, in addition to ignoring tweets that are classified as such, meaning only the positive and negative classes are compared, the results would be much better. Table 5.10 presents the results under such assumptions, and is equivalent to Table 5.8 without the NEU row. To further clarify the meaning of ignoring the NEU tags, deleting the NEU row is equivalent to assuming that there is an oracle that perfectly classifies NEU and NONE tweets as such, before feeding them to the polarity classification algorithm. Correspondingly, deleting the NEU column of table 5.6 is equivalent to ignoring every tweet that is classified as NEU or NONE, even if that means ignoring tweets that are really positive or negative and were wrongly tagged. Performance metrics under these assumptions are presented in table 5.11

		Classified		
		N	P	**Total**
Real	N	913	592	**1505**
	P	214	1860	**2074**
	Total	**1127**	**2452**	**3579**

Table 5.10: Confusion Matrix for 2 Classes, ignoring NONE and NEU tags, and assuming perfect filtering of neutral tweets.

	N	P
Precision (%)	81.01	75.86
Recall (%)	60.66	89.68
F-measure (%)	69.38	81.19
Accuracy (%)	77.48	
κ	0.521	

Table 5.11: Performance Metrics for 2 Classes, ignoring NONE and NEU tags, and assuming perfect filtering of neutral tweets.

These results show how important it is to previously filter those tweets that are neutral or aren't polar. There is a great number of studies where a separate classifier is trained for classifying whether a text contains sentiment, before being fed to the main Opinion Mining algorithm; this task is called *subjectivity classification* [49]. Unfortunately applying such analysis was outside of the scope of this work, but implementing it would signify a considerable increase in performance. Other possible improvements are mentioned in Section 6.4.

Moreover, Table 5.12 presents a summary of the performance metrics presented earlier, under the different considerations for NEU and NONE tags.

	P_{pos}	P_{neg}	R_{pos}	R_{neg}	FM_{pos}	FM_{neg}	Accuracy	κ
Unchanged	59.98	63.14	67.08	43.13	63.33	51.25	55.73	0.325
Ignoring `NEU` tags	59.98	63.14	89.68	60.66	71.88	61.88	60.99	0.332
Ignoring `NEU` tags and perfect filtering	75.86	81.01	89.68	60.66	81.19	69.38	77.48	0.521

Table 5.12: Performance metrics with different treatments for `NEU` and `NONE` tags. All values except κ are percentages.

Clearly, the best performance occurs when both the tweets tagged as `NEU` and `NONE` are ignored, and those tweets that are really neutral are tagged as such. The current implementation, however, does not filter tweets that are really neutral, so the best performance occurs when ignoring the `NEU` and `NONE` tags, corresponding to the second row.

Finally, Table 5.13 presents a comparison between the baseline and the current system. The baseline is simply the current system without applying the negation, intensification and adversative rules. In other words, for obtaining the baseline performance metrics, tweets were tagged by adding the polarities of the words composing them, obtained from the lexicon.

	P_{pos}	P_{neg}	R_{pos}	R_{neg}	FM_{pos}	FM_{neg}	Accuracy	κ
Baseline – No Rules	57.82	60.71	88.28	58.24	69.88	59.45	58.74	0.300
Dependency-based Ignoring `NEU` tags	59.98	63.14	89.68	60.66	71.88	61.88	60.99	0.332
Variation	**+2.16**	**+2.73**	**+1,40**	**+2,42**	**+2,00**	**+2,43**	**+2,25**	**+0,032**

Table 5.13: Baseline Comparison. All values except κ are percentages.

It is possible to see that all metrics are better with the application of the rules, in particular, those concerning the classification of negative tweets are the most benefited. At this point, it is necessary to question whether the gain in classification performance is worth the time investment for each tweet, since the dependency parsing step of the process is by far the slowest. Further, the small gain in precision, recall, F-measure, accuracy and κ, as opposed to the values reported by Vilares et al. [43], might be explained by the fact that the author validated his algorithms with review corpora, while the application presented in this thesis deals with microblogging, whose written properties are different than those of reviews.

5.2 Retail Case Study

In this section, the results of applying the Opinion Mining algorithms to a dataset containing tweets related to the Chilean retail company Falabella will be presented. As previously stated in Section 4.2, the dataset contains a total of 56.773 tweets, ranging from October the 3rd 2014, to March the 4th 2015. The criterion for extracting a tweet was simply that it had to contain the keyword `falabella` either in its text field or in the screen name of its author.

5.2.1 Dataset Characterization

Before analyzing the metrics related to the polarity of the dataset, this subsection presents the most relevant aspects that characterize it.

User-related Metrics

The dataset contains a total of 24.595 Spanish-speaking users, which correspond both to corporate and normal accounts. Corporate accounts are defined as those that represent any entity different from a single user, whereas normal accounts as those representing only one person. It is also worth mentioning that these users are not limited to those that are only Chilean, meaning there are accounts from other countries such as Argentina and Colombia.

The first interesting pattern to mention is that the vast majority of Twitter users (75.2%), in this particular dataset, tweeted only once, followed by those that tweeted twice (13.2%), and so on. Another significant pattern is that even though users that have more than 10 tweets are a minority (1.4%), there is a considerable amount of tweets that was authored by them (36.8%). Further, this minority holds more tweets than the vast majority of Twitter users (32.6%). Figure 5.1 depicts the percentage of users and the percentage of tweets as a function of the amount of tweets users have in the dataset; by observing it, it is possible to deduce that 69.4% of tweets was authored either by 1-time tweeters or by frequent tweeters, 11.5% by 2-time tweeters and the remaining 19.1% by those users that tweeted between 3 and 10 times.

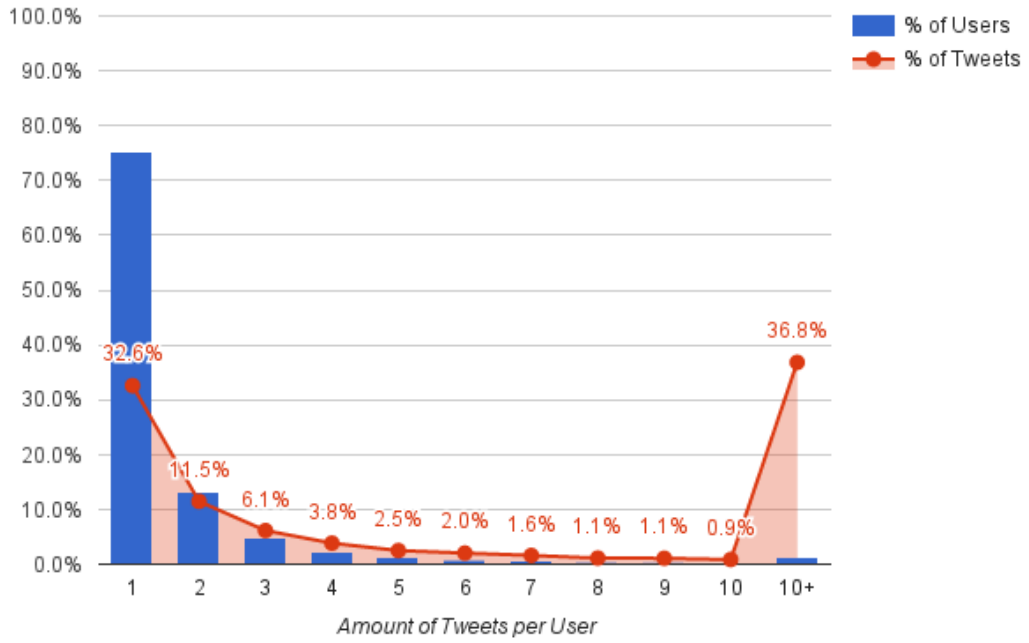


Figure 5.1: % of Users and % of Tweets With Respect to the Amount of Tweets per User.

Additionally, Figure 5.2 presents the cumulative frequency of tweets according to the cumulative frequency of users. The graph shows that 10% of the users hold more than 50%

of the total amount of tweets, and a little more than 50% of users hold 80% of the tweets. This implies that most tweets of the dataset belong to a minority of users.

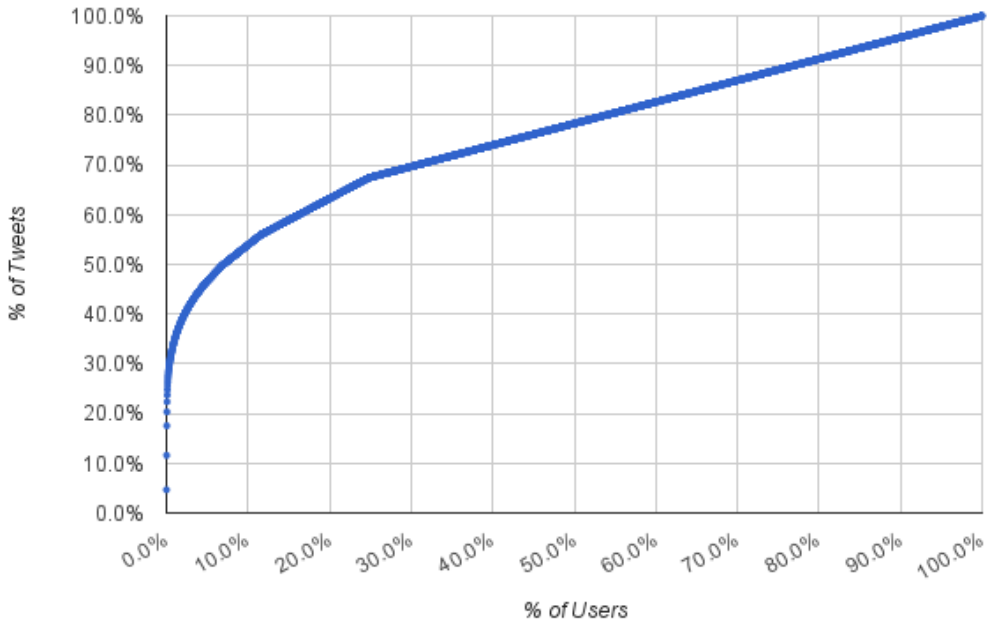


Figure 5.2: Cumulative % of Tweets With Respect to Cumulative % of Users.

Furthermore, Figure 5.3 depicts the amount of tweets by the 20 users that hold the greatest amount of tweets in the dataset. It comes as no surprise that most of these accounts don't present content that is very relevant for Opinion Mining. Indeed, their content can be categorized as being corporate[2] news, ads (short for advertisements), that often mention sales and discounts, mentions of the word "falabella" but in a context that is not related to the retail company, accounts containing the word falabella in their name and unrelated to the company, and viral ads. These 20 users account for 22.7% of the total amount of tweets in the dataset.

Table 5.14 assigns the most recurrent type of content of these 20 most-active users to one of the aforementioned categories. This means that the users reported in the table might have posted tweets from another category, but in every case the category represents at least 90% of each user's content.

The viral ad campaign was based on the hashtag `#NoCubre` (doesn't cover), and consisted in posting creative tweets mentioning the events that Falabella's insurance subsidiary did not cover. The company promised that 2 users would be awarded with a giftcard. Obviously none of the users presented in the table above were selected as winners, since they just used an automatic tweet generator which did not require them to be very creative. Annecdotally, the two winners only tweeted once and twice respectively.

[2]Corporate content is defined as all content tied to a corporate account. This often includes information on sales and promotions, advertisements, and sometimes answers to angry customers.

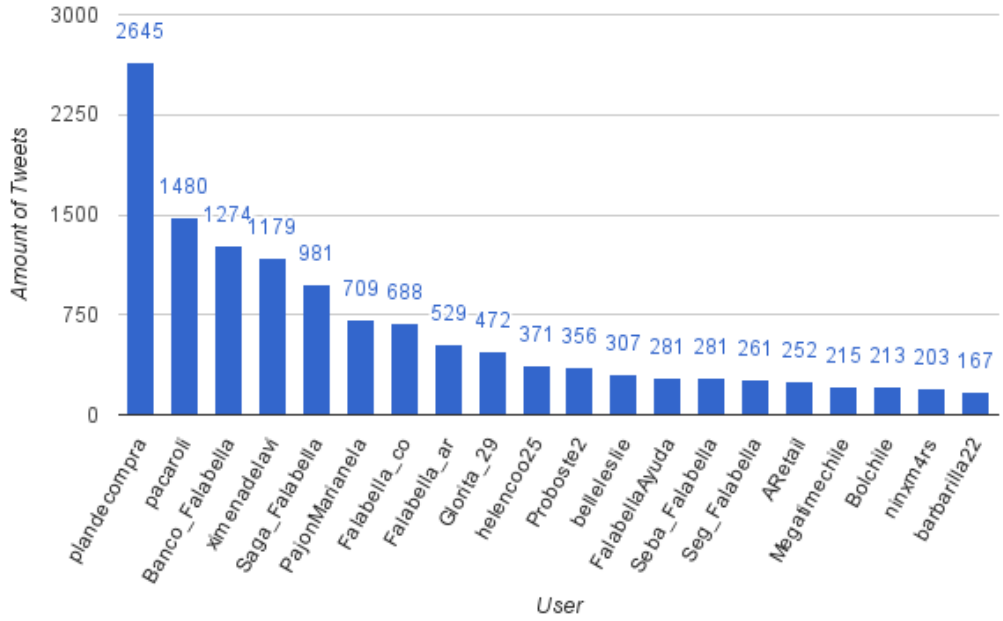


Figure 5.3: Top 20 Most Active Users.

Username	Type of Content	Username	Type of Content
plandecompra	Ads	Proboste2	Viral Ad Campaign
pacaroli	Viral Ad Campaign	belleleslie	Viral Ad Campaign
Banco_Falabella	Corporate	FalabellaAyuda	Corporate
ximenadelavi	Viral Ad Campaign	Seba_Falabella	Irrelevant Name
Saga_Falabella	Corporate	Seg_Falabella	Corporate
PajonMarianela	Irrelevant Content	ARetail	News
Falabella_co	Corporate	Megatimechile	Ads
Falabella_ar	Corporate	Bolchile	News
Glorita_29	Viral Ad Campaign	ninxm4rs	Irrelevant Content
helencoo25	Viral Ad Campaign	barbarilla22	Viral Ad Campaign

Table 5.14: The 20 Most Active Users and the Type of Content They Usually Post.

The previous work was extended to include the 100 most active users (0.41% of the total userbase, corresponding to 30.4% of all the tweets in the dataset), and results are presented in Figure 5.4. By observing it, it is possible to confirm that the content posted by the most active users is not the most relevant. Additionally, even though the sample is not at all representative for the total userbase, it is still important to consider that this 0.41% of users correspond to 30.4% of all the tweets in the dataset.

It is also possible to observe that the majority of content posted by the most active users is corporate, related to a viral ad campaign or plainly irrelevant. Complaints are also present, and often manifest themselves as the same repeated message concerning a problem a client had, or, less frequently, as various different messages also concerning the same problem. Finally, ad and news accounts have a smaller, but still considerable presence. The smaller category is that of the users that mostly post checkins from the Swarm application[3].

[3] `https://www.swarmapp.com/`, Accessed on August 19, 2015

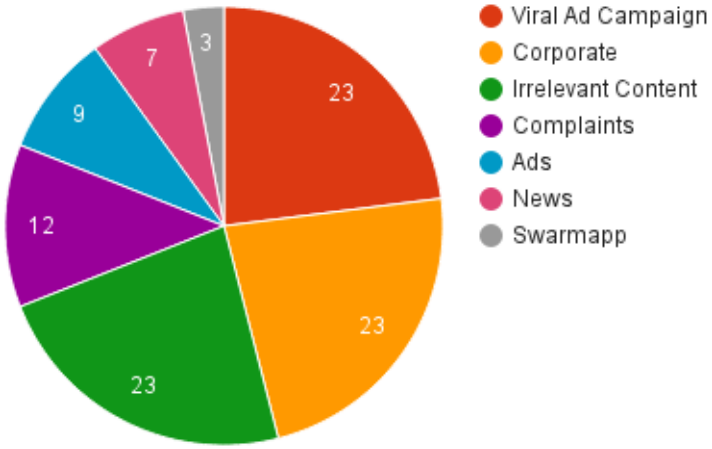


Figure 5.4: The Most Recurrent Content Posted by the Top 100 Most Active Users.

Date-related Metrics

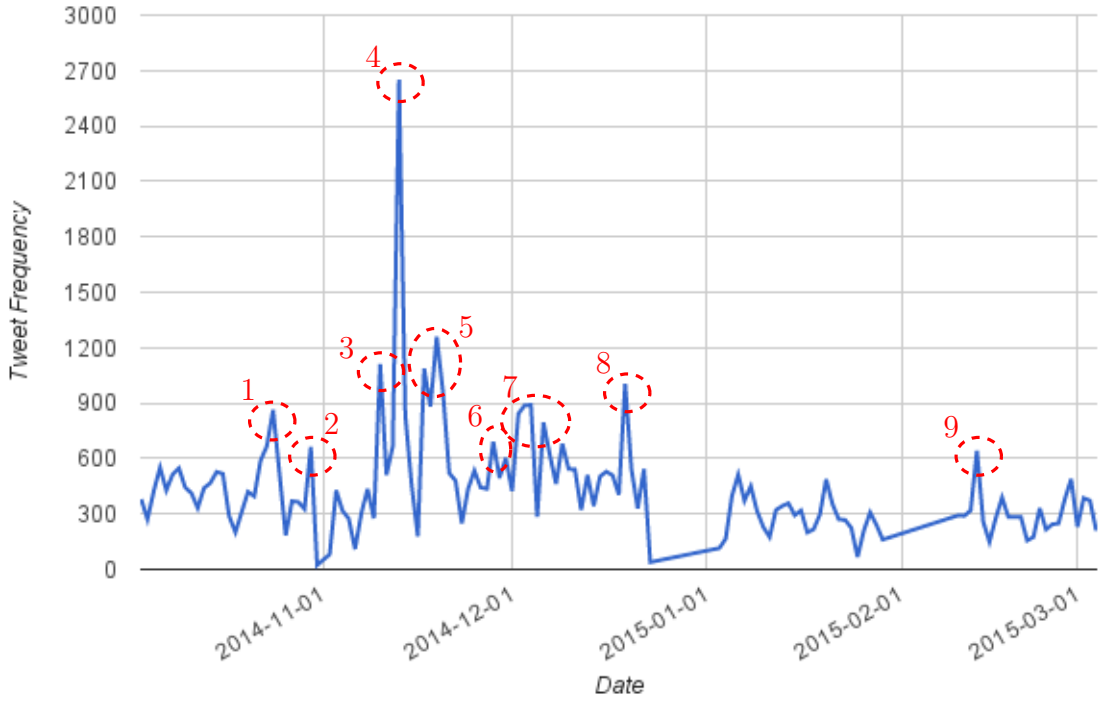


Figure 5.5: Daily Tweet Frequency.

Figure 5.5 displays the daily tweet frequency of the whole dataset; in it it is possible to observe 9 significant peaks corresponding to real-life events of varying degrees of importance, briefly described below. It is also possible to see there are two periods of time where tweets were not extracted; these correspond to the periods going from 2014-12-24 to 2015-01-02, and from 2015-01-30 to 2015-02-09. In both cases, the data loss was caused by the process being terminated by the remote Amazon EC2 server, and the impossibility of resuming it.

1. **Anniversary – 2014-10-24:** The first peak was caused by the 125$^{\text{th}}$ anniversary of Falabella, and the fact that that day Ricky Martin would perform in a concert in the National Stadium of Chile in name of the company.

2. **#Retrotubers – 2014-10-30:** A radio in Colombia offers listeners to participate for a voucher to be spent in Falabella by tweeting with the hashtag #Retrotubers.

3. **Cyber Monday Argentina – 2014-11-10:** Products are offered at a discount price in Falabella Argentina's online store, and people comment on the event.

4. **#NoCubre Viral Ad Campaign – 2014-11-13:** This campaign was created for promoting Falabella's insurance subsidiary's insurance plans by tweeting absurd and funny situations that their plans did not cover. Most of these tweets were authored by a few authors, presented in Table 5.14, that just spammed a site for automatically generating them.

5. **Cyber Monday Chile & #NoCubre – 2014-11-17 to 2014-11-20:** On 2014-11-17 Cyber Monday took place in Chile. Additionally, the #NoCubre viral ad campaign continued to generate tweets until 2014-11-20.

6. **Black Friday Chile and Colombia – 2014-11-28:** People comment on the Black Friday event.

7. **Racist Catalog Picture – 2014-12-03 to 2014-12-07:** A picture depicting four blonde girls was published in a Peruvian Christmas advertising catalog from Falabella. The outcry was caused because people felt these girls did not represent the most common Peruvian phenotype, which typically has darker features. The catalog had to be finally pulled out of circulation.[4]

8. ***Palacio Falabella* (Falabella Palace) – 2014-12-19:** On this date, the Chilean mayoress of the *Providencia* commune, used municipal dependencies, namely, Palace Falabella, for her nephew's marriage ceremony. This obviously caused indignation, which manifested itself as a great number of sarcastic tweets asking for renting Palace Falabella for personal events.[5] This event is unrelated to the retail company.

9. **#TrendingShoppingFalabella Viral Ad Campaign – 2015-02-13:** In this campaign, Twitter users were required to publish tweet with the hashtag #TrendingShoppingFalabella with a product they wished to buy, in order to participate and eventually get a discount in that product category.

5.2.2 Polarity Analysis

In this section, the metrics related with the tweets' polarity will be presented. Just like the previous section, this one will be divided in the analysis of users and the analysis of dates. It is important to remark that this analysis is an example of what could be achieved by using the results provided by the Opinion Mining platform, and does not represent a thorough study.

[4]Find more information in `http://goo.gl/asWejr` and `http://www.peruthisweek.com/news-saga-falabella-lima-peru-104665`, Accessed on August 24, 2015.

[5]More information in Spanish can be found in `http://www.elmostrador.cl/noticias/pais/2014/12/19/el-elegante-palacio-municipal-de-providencia-que-josefa-errazuriz-le-presto-a-su-sobrino-para-casarse/`, Accessed on August 24, 2015

User-related Polarity

The first interesting metric to analyze, concerning the users, is the average polarity of the content types of the 100 most active users, presented in Figure 5.6. In it, it is possible to see the viral ad campaign type separated in two: #TrendingShoppingFalabella and #NoCubre, explained in the previous section. It is also possible to observe that the results are what one might expect: corporate content and ads are positive in average, whereas complaints are negative. Swarmapp posts and News are slightly positive, but closer to the neutral mark. Finally, both viral ad campaigns have very opposed polarities, which is explained by #TrendingShoppingFalabella mostly promoting content related to people's desires –which should be intrinsically positive–, and #NoCubre promoting content that indicates negative events that Falabella's insurance plans do not cover, hence resulting in negative polarity.

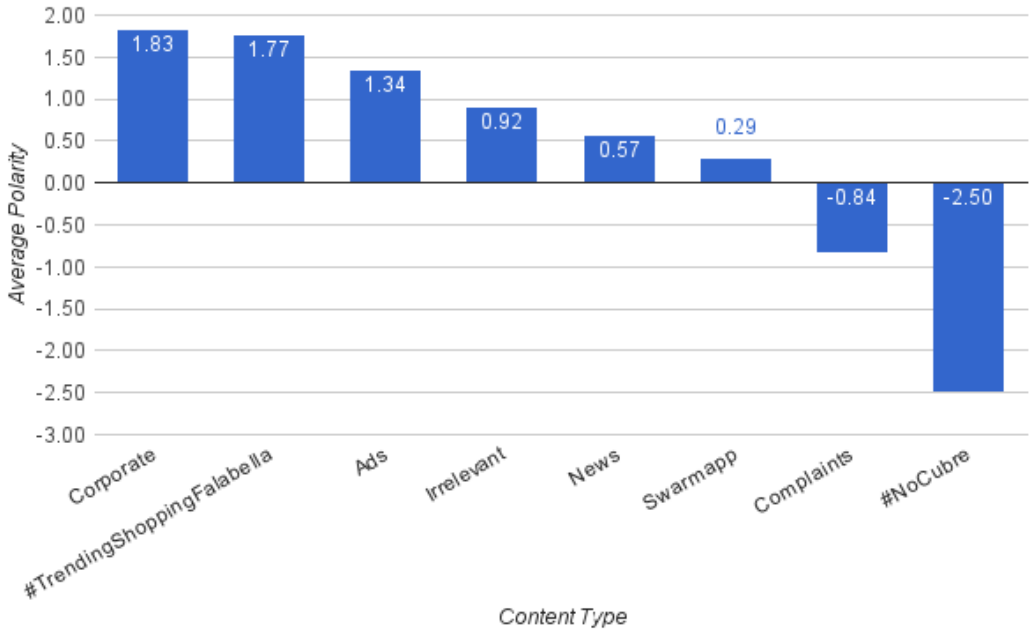


Figure 5.6: Average Polarity With Respect to the Content Type of the 100 Most Active Accounts.

Furthermore, Figure 5.7 presents a more granular polarity distribution for each content type.[6] Obviously, this figure was created only for exploratory purposes since the sample size is too small, and it would not be wise to extrapolate these results. With this in mind, it is possible to observe that corporate content is consistently positive, same as ads and the #TrendingShoppingFalabella campaign. News, on the other hand, are mostly neutral, whereas the #NoCubre campaign is consistently very negative for the reasons stated above. Finally, complaints are evenly distributed in the polarity range going from -4 to 2, which probably occurs because there are many ways to complain; some users just manifest their annoyance by describing their situation, while others post sarcastic comments, ask rhetoric questions, or plainly insult the company.

Moreover, Figure 5.8 presents the average polarity distribution for the users in the dataset. In order to build it, the average tweet polarity was calculated for each user, then rounded to

[6]Swarmapp and irrelevant content was left out intentionally.

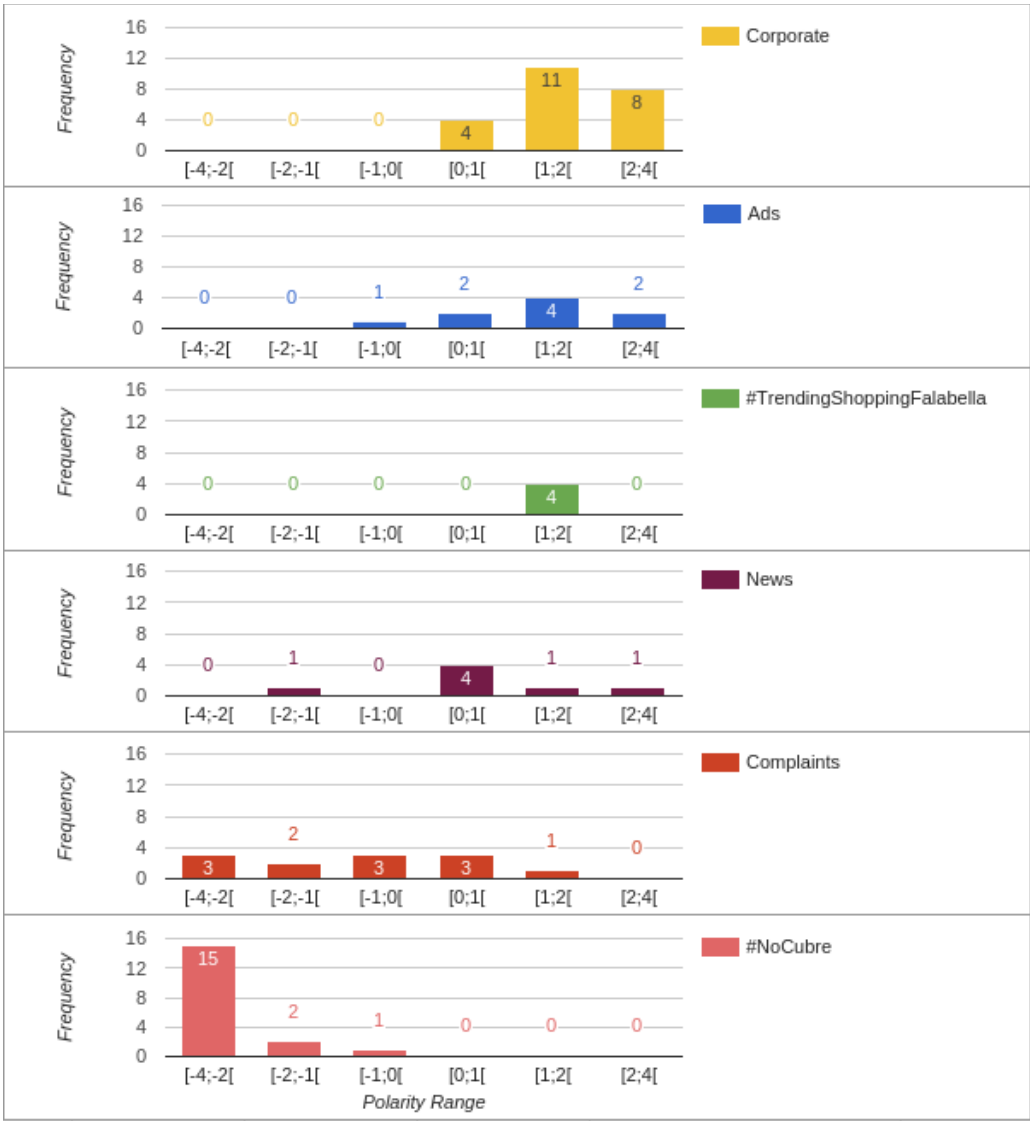


Figure 5.7: Content Type Polarity Distribution.

the unit, and finally counted. So, for instance, it is easy to see that users with an average polarity of 2 are the most common. Finally, those users with an average polarity of 0 (8285, or 33.7% of the users in the dataset) were removed. This distribution shows that users are mostly neutral or slightly polar towards the positive end. However, it also displays that there is considerable number of users both in the negative and positive ends of the distribution. Below, some negative and positive tweets, along with their translation, are presented.

Negative Tweets (-10)

(5.5) *mal, nva tienda en Chillan servicio atencion es deficiente, muchos vendedores todos conversando se molestan x pedir ayuda @Falabella_Chile*

(5.6) A shame, in new store in Chillan service quality is deficient, a lot of sales assistants talking between themselves get annoyed when client asks for help

(5.7) *@Falabella_Chile @FalabellaAyuda indignada nuevamente falabella y sus abusos, unos rotos falabella ahumada desde el gerente en adelante*

(5.8) @Falabella_Chile @FalabellaAyuda indignant again, falabella and its abuses, they're all peasants, from the manager on

(5.9) *Hay comerciales más desagradables q los de VISA Falabella y los CMR puntos? Detestables!!!*

(5.10) Are there any commercials more obnoxious than those of VISA Falabella and CMR points? Detestable!!!

Positive Tweets (10)

(5.11) *@Saga_Falabella @arizaga_a @NicolaPorcella @estoesguerra_tv eligieron a los mejores.!!! Me encantó el desfile.. espectacular final*

(5.12) @Saga_Falabella @arizaga_a @NicolaPorcella @estoesguerra_tv you chose the best.!!! I loved the parade.. spectacular ending

(5.13) *Y seguimos en compras ... pero tengo hambre :(igual, super feliz jejeje "Saga Falabella, te amo" http://instagram.com/p/vy2hBMxAqs/*

(5.14) And we're still shopping ... but I'm hungry :(anyway, super happy hehehe "Saga Falabella, I love you" http://instagram.com/p/vy2hBMxAqs/

(5.15) *@Saga_Falabella #arribamujeres La belleza de una mujer reside en sus ojos q son la puerta de entrada a su corazón....donde reside el Amor!!*

(5.16) @Saga_Falabella #arribamujeres A woman's beauty lives in her eyes which are the entrance to her heart....where Love lives!!

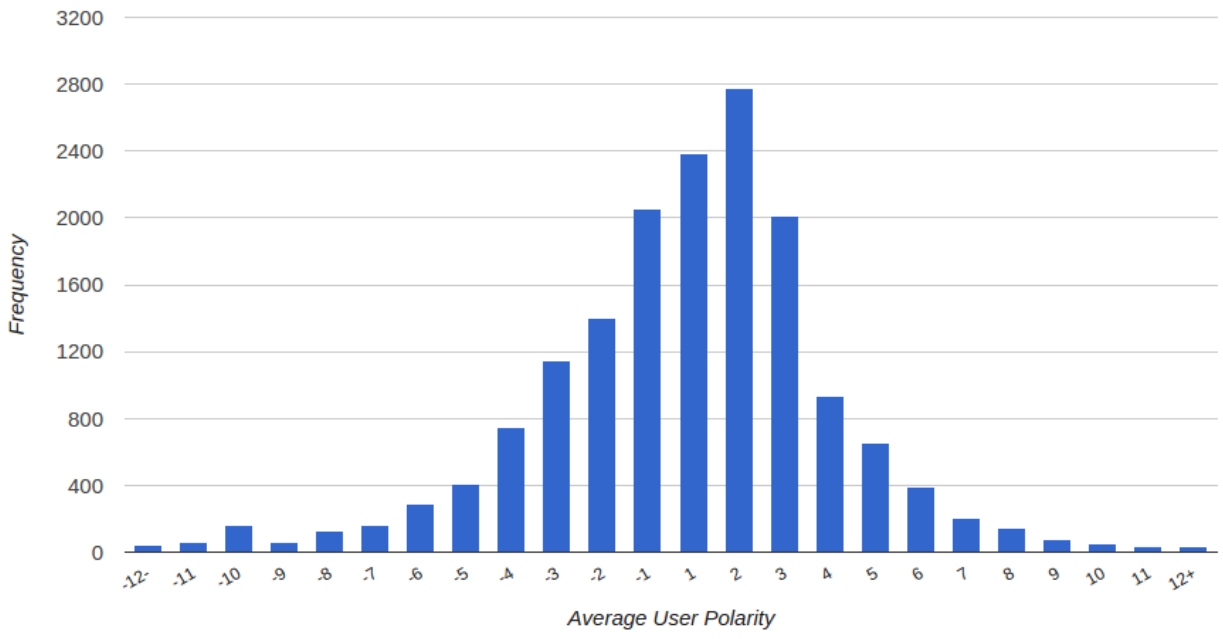


Figure 5.8: User Average Polarity Distribution.

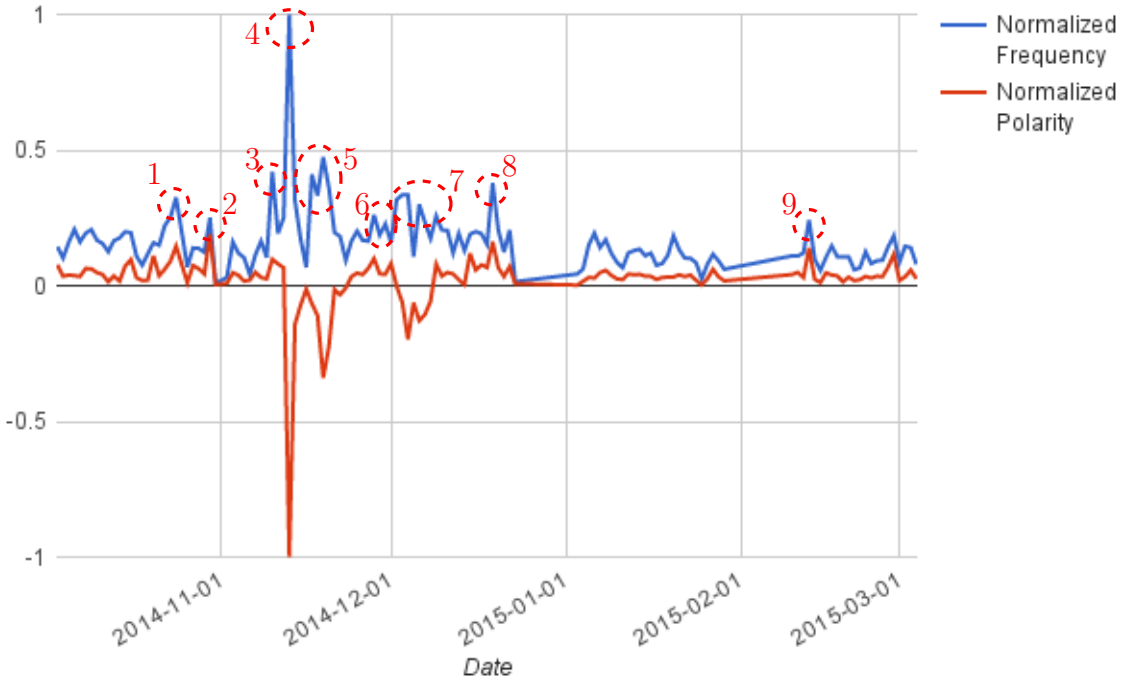


Figure 5.9: Normalized Daily Tweet Frequency and Polarity.

Date-related Polarity

Figure 5.9 presents the normalized daily tweet frequency and polarity. Both metrics were normalized in order to be able to compare them, since their values were considerably different (average daily polarities are close to 0 whereas daily frequencies are always greater than 100). The figure shows that some spikes in frequency, corresponding to the events mentioned in Section 5.2.1, have positive polarity, whereas some other have negative polarity. Below, each event is characterized, based on its polarity.

1. **Anniversary:** This event had an overall positive polarity. Indeed, on that day, 371 positive and 130 negative tweets were published. Further inspection shows that some positive tweets are actually sarcastic; to give an example:

   (5.17) *Disfrute del concierto de Ricky en el 13 por el aniversario de Falabella... fue pagado con la plata que le han quitado a todos.. ;) Enjoy it*

   (5.18) Enjoy Ricky's concert in channel 13 in honor to Falabella's anniversary... it was paid with money they have taken from everyone.. ;) Enjoy it

   However, most of the positive tweets were actually messages of gratefulness to Falabella and praise to Ricky Martin, like shown below:

   (5.19) *hoy adoro a FALABELLA !!!!*

   (5.20) today I love FALABELLA !!!!

   (5.21) *#RickyMartinEnEl13puchas q lindo ricky...gracias falabella...madre mia quiero estar alí...*

(5.22) #RickyMartinEnEl13puchas Ricky is so cute...thank you fallabella...my goodness I want to be there...

As for the negative tweets, inspection showed that most are unrelated to the Ricky Martin's concert or the anniversary, and are instead common complaints:

(5.23) *Quien vende productos usados? @Falabella_Chile El peor servicio al cliente de Chile ... pic.twitter.com/b8ezGqOpD1*

(5.24) Who sells used products? @Falabella_Chile The worst service quality in Chile ... pic.twitter.com/b8ezGqOpD1

The overall impression of this event is that there is a considerable amount of people that liked Falabella bringing Ricky Martin to Chile to celebrate its 125^{th} anniversary, but there is also a group of people that are unsatisfied with Falabella, in particular, with its financial subsidiary and its service quality.

2. **#Retrotubers:** All of the tweets related to this hashtag nicely ask for the prize offered by the radio, and most are classified as being positive or neutral. The event is not directly related to the retail company so it will not be analyzed further.

3. **Cyber Monday Argentina:** The average polarity for this day appears to be slightly positive, however the majority of tweets mentioning the Cyber Monday in Argentina are either negative or sarcastic. The overall impression is that Falabella does not have the required infrastructure and know-how for executing a Cyber Monday. People were confused since they were not able to pay with some credit cards, and the high traffic did not allow for clients to navigate the page normally.

4. **#NoCubre:** This campaign generated a high amount of tweets from a few users and all were negative. The real impact of the campaign is difficult to estimate since the fact of the tweets being negative does not necessarily implies a negative sentiment towards the company. An example of these tweets is:

   (5.25) *Seguro Catastrófico de @Seg_Falabella #NoCubre fin del mundo causado por un villano en las nubes http://bit.ly/NOCUBRESF*

   (5.26) Catastrophic Insurance of @Seg_Falabella #DoesntCover the end of the world caused by a villain in the clouds http://bit.ly/NOCUBRESF

5. **Cyber Monday Chile:** One week after Argentina's Cyber Monday, the event is repeated in Chile. This time comments are notoriously more negative in average, which could be explained by the fact that Chilean are better at complaining, use less sarcasm, and that there are some tweets of the #NoCubre campaign that add to the negative overall polarity. Tweeters complain about the queue to enter the e-commerce, long wait times, the fact that the stock information of products is not available before the purchase step, that customer service department does not pick up the phone, and even that the discounts are not good enough. In summary, there is a clear signal that Falabella must aim its efforts to improve the website and IT infrastructure for supporting the high traffic during the event.

6. **Black Friday in Chile and Colombia:** This event generated less tweets than the Cyber Monday. The slightly positive polarity is explained by the fact that most tweets were authored by corporate accounts promoting the event. Some users complain by saying that discounts are not good enough, either because prices are not low or because there are not enough products with discounts.

7. **Racist Catalog Picture:** The picture in Falabella's Christmas catalog caused irritation in the Peruvian nation, which partly manifested itself in a series of tweets either informing the existence of the catalog or denouncing it. Even though Falabella apologized and promised to pull the catalogs back, criticism continued during several days. Example 5.27 shows a tweets informing of the racism controversy, and example 5.29 a direct reproach:

   (5.27) *BBC Mundo - Perú: la polémica sobre racismo que obligó a la tienda Falabella a retirar su campaña navideña http://bbc.in/1zXdpJk*

   (5.28) BBC Mundo - Perú: the racism controversy that forced Falabella to withdraw its Christmas campaign http://bbc.in/1zXdpJk

   (5.29) *Falabella me fallaste*

   (5.30) Falabella, you failed me

8. **Palacio Falabella:** This event is mostly positive because most of the comments were sarcastic. It will not be analyzed further since it is not related to the company.

9. **#TrendingShoppingFalabella:** As with the other campaigns, tweets with this hashtag are supposed to have an intrinsic polarity, and in this case, a positive one. Most of the tweets, if not all, contain the hashtag and a short text describing what the user desires to buy at Falabella. Example 5.31 shows the tweet of a user qualifying Amphora handbags as being pretty and hoping to get a discount voucher for them.

   (5.31) *#TrendingShoppingFalabella de @Falabella_Chile carteras Amphora son las mas bonitas CC http://bit.ly/TrendingShopping*

   (5.32) #TrendingShoppingFalabella of @Falabella_Chile Amphora handbags are the prettiest CC http://bit.ly/TrendingShopping

Keywords-related Polarity

This final subsection presents the average polarity associated with certain keywords, chosen according to their frequency in the whole dataset and their relation to the retail company. Table 5.15 shows the keywords, their frequency and their average polarity.

Further, Figure 5.10 displays these keywords ordered according to their polarity in ascending order for comparison purposes. In it, it is possible to observe that keywords related to the service quality (`servicio`, `cliente`, `atencion`), complaints (`sernac`) and TV ads (`publicidad`, `comercial`), are negative, whereas keywords related to products (`computadores`, `ropa`), and brands (`sony`, `samsung`, `cencosud`, `cmr`), are often positive. Additionally, the keyword `quiero` (I want), is associated with the Ad campaign #TrendingShoppingFalabella hence its higher

Keyword	Frequency	Average Polarity	Keyword	Frequency	Average Polarity
atencion	456	−1.24	quiero	1119	2.53
cencosud	210	1.36	reclamo	240	−0.11
cliente	741	−0.51	retail	716	0.53
cmr	785	0.99	ropa	841	0.88
comercial	946	−0.64	samsung	320	1.01
computadores	239	0.80	sernac	487	−0.67
marca	1286	0.58	servicio	733	−1.29
paris	231	0.47	sony	349	1.2
precio	574	1.01	tarjeta	856	0.92
producto	1120	0.14	tienda	2031	−0.26
promo	2700	0.75	tv	593	0.15
publicidad	946	−1.85	vendedor	122	−0.122

Table 5.15: 24 Frequent Retail-related Keywords.

polarity. Furthermore, high polarity for products and brands is explained by the fact that the users that tweet the most about them are corporate, and do so for advertising purposes; common users, in contrast, don't speak often of products and brands in tweets containing the keyword "falabella." Negative polarity for TV ads is explained by the fact that most users that tweet about them do so in a negative fashion, as depicted in example (5.33). Similarly, negative polarity for keywords related to service quality is explained by the fact that users that comment on it, often do it for complaining, as illustrated by example (5.35). This is consistent with the main reasons for complains mentioned in Section 1.1.3.

(5.33) *Odio con toda mi vida el comercial de #falabella !! Escucho esa canción y desesperadamente busco el control para poner mute!!!!*

(5.34) I hate #falabella's commercial with my life !! I hear that song and desperately seek the remote to mute it!!!!

(5.35) *Leeenta la atención al cliente en falabella manquehue. Muuyy lenta. @Falabella_Chile*

(5.36) Slooow customer service in falabella manquehue. Veery slow. @Falabella_Chile

All this being said, it is important to consider that, from a potential user of the OM platform point of view, it would be very useful to be able to classify, at least at some degree, the types of accounts. That way it would be possible for them to really know what is being said about them in Twitter.

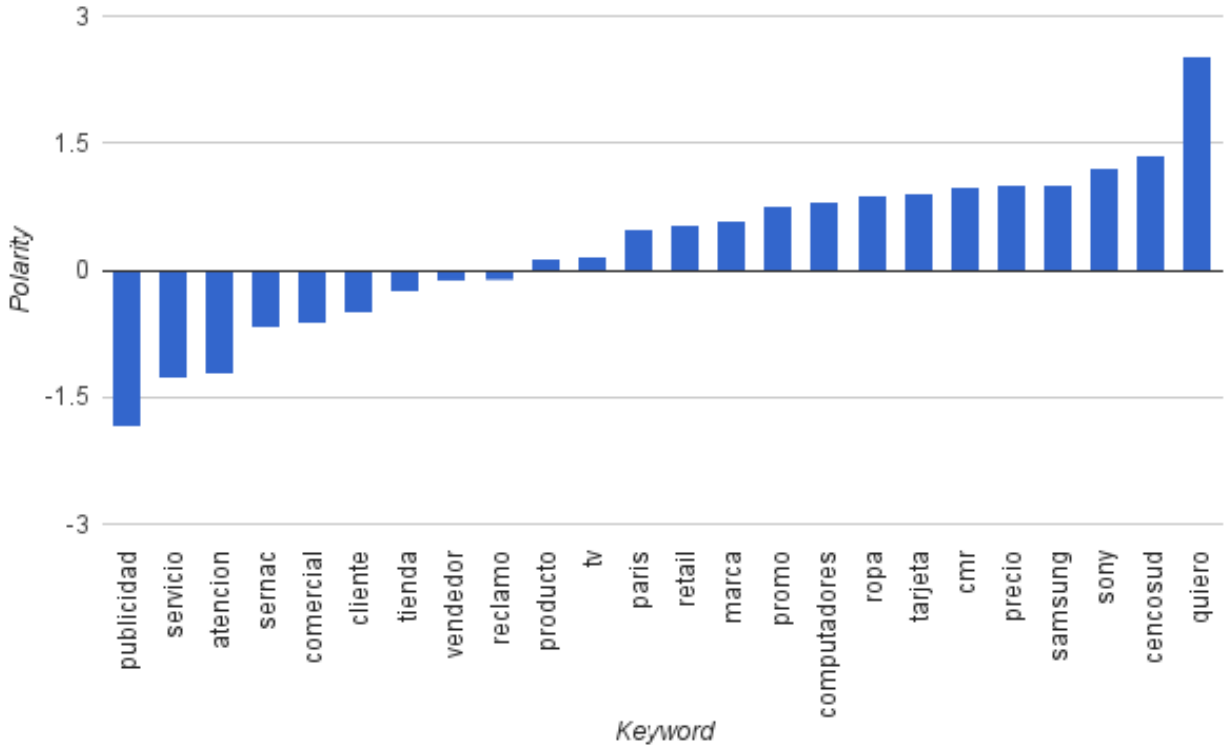


Figure 5.10: 24 Frequent Retail-related Keywords.

Chapter 6

Conclusions

6.1 Synthesis

This work presents the creation of a system capable of extracting Spanish tweets from Twitter, preprocessing them, assigning them a certain polarity, and visualizing the results. Specifically, the algorithms for classifying tweets are based on the study by Vilares et al. [43], who proposed an unsupervised, lexicon-based method relying on syntactic dependencies for performing Opinion Mining on reviews. Conversely, the current platform was applied to tweets, which present a structure that is radically different to them and pose greater challenges when being processed.

Admittedly, validation results while ignoring those tweets classified as neutral (61.88% negative, and 71.88% positive F-measures), are not as high as a production-level application would require, yet they are good enough for performing exploratory analyses such as the one presented in section 5.2. Low performance metrics might have been caused by several reasons, namely, the numerous complexities associated with classifying tweets, having used a corpus from a different domain to train the syntactic parser, the fact that the Part-of-Speech tagger was not specifically created for dealing with microblogging data, and the syntactic rules (intensification, negation, and adversative clauses), not being distinctly tailored for dealing with grammatical constructs often observed on tweets.

All these considerations aside, implementing the syntactic rules proposed by Vilares et al. resulted in a slight increase of performance. Indeed, Negative F-measure increased by 2.43% when compared with the baseline method, whereas Positive F-measure increased by 2.00%, suggesting that the incorporation of syntactic information to the process produces better results, a finding that is similar to the one reported by Jiang et al. [24]. Now, it is necessary to consider whether this slight improvement in performance is worth the increase of time required for processing each tweet, in fact, the dependency parser makes the whole process of classifying a tweet 10 times slower as compared to the baseline method. It could be argued that such an increase in processing time should only be justified by a significant increase of classifying performance. Therefore to justify the use of the dependency parser,

the rules presented in section 3.6 should be improved to better deal with Twitter data.

Furthermore, even if the validation metrics are no the optimal, the retail case study revealed several interesting facts that were manually corroborated. First of all, there is a considerable amount of information that can be exploited without the need for Opinion Mining, as presented in section 5.2.1. By using simple metrics it was possible to discover that there is a vast amount of tweets that is authored by very few users; 10% of users hold more than 50% of tweets. Further analysis showed that the most active users, meaning those that have the most amount of tweets in the analyzed dataset, often publish content that is irrelevant to the analysis of opinions, such as advertisements, tweets related to viral campaigns and corporate content, which is consistent with the findings presented in [122]. This supports the claim that being able to detect and filter different types of content is vital for obtaining better insights. Additionally, the vast majority of users do not tweet frequently, in fact, 75.2% of users present in the dataset tweeted only once, and 98.6% tweeted 10 times or less.

Moreover, by complementing the previous analysis with polarity data, it was possible to uncover more interesting characteristics associated to Twitter account types and the polarity of the content they usually publish; corporate and advertising accounts usually tweet positive content in average, which is logical since it would be unwise to attempt to improve brand awareness and brand image by posting negative content. News accounts often post content that is closer to the neutral part of the polarity spectrum, meaning they are mostly impartial, which is also what should be expected. Lastly, users whose content correspond mostly to complaints have a more evenly distributed polarity, which is explained by the fact that there are many ways to communicate one's discontent, ranging from describing an uncomfortable situation to insulting those responsible for it. Besides, the two observed major viral campaigns presented radically different overall polarities, which implies that, perhaps, polarity is not a good indicator of the success of such campaigns. Similarly, the coarse amount of tweets containing the campaign's hashtag is not accurate either, since usually a small number of users are responsible for most of them. The amount of users that used the campaign hashtag at least once would be a more illustrative metric to approximate the campaign's reach.

The analysis of the daily frequency at which tweets were published, along with the average daily polarity, also revealed several insights. Each one of the most prominent peaks of daily frequency proved to correspond to real-life events of varying degrees of importance, ranging from praises to a musical artist, to racism-related controversies. Further, there is evidence supporting the claim that the most popular events, meaning those with the highest amount of daily tweets, are often accompanied by a significant decrease of polarity, which reconciles with the findings suggested in [131]. Such was the case for the Chilean Cyber Monday, and the racist picture in the Peruvian catalog. Less popular events, on the other hand, often come with an associated slight increase in polarity, exemplified by Falabella's anniversary and by the #TrendingShoppingFalabella campaign. In addition, the most recurrent type of complaint was related to a deficient quality of service, which agrees with the results exhibited in the 2014 Consumer's National Service descriptive study [7], and mentioned in section 1.1.3. As a final note, the most significant peak in mentions of the keyword `falabella` was due to the Cyber Monday, just like reported in the study [8] from 2013, which means that the company should probably allocate more resources in attempting to improve the shopping

event's execution, lest its brand image continues to deteriorate.

Finally, there are several highlights to be made about the application's design and implementation. First, the application's development process proved to be very illustrative, indeed, the version presented in this thesis was the result of several iterations, in which improvements were supported by the insights obtained through the reading of the books by Steve McConnell [10], Matt Weisfeld [145], and Andrew Hunt and Dave Thomas [146]. Second, the application was modularily designed with extensibility and maintainability in mind; for example, the Data Extraction module could be easily modified for obtaining tweets through Twitter's API instead of a crawler, the Preprocessing Module could be improved to use a better-trained POS tagger, the Polarity Classification module could be extended to incorporate machine-learning-based algorithms, and the Visualization Module could be completely changed by a better visualization engine such as Tableau. None of these individual modifications would negatively affect the process, as long as as the input and output of each one of them remained constant.

6.2 Limitations

In its current state, the application is still a quite limited prototype. First, as mentioned earlier, the performance while classifying tweets is not high enough to represent a breakthrough in Opinion Mining, however they still allow the platform to be used for research purposes. Second, the platform is neither able to automatically recognize the type of user authoring a tweet, nor the category to which its content belongs. Third, the visualization module is not complete enough to provide enough insightful information out of the box; in order to do the exploratory analysis presented in section 5.2, the database had to be queried directly, and several further steps had to be manually performed in order to obtain the results.

Additionally, the classification implementation has only been tested for one user at a time, however it is highly likely that performing several requests coming from different users to it would only make the process classification process even slower. To be able to openly offer an API, or any related service for that matter, the application's processing speed *must* be improved.

6.3 Implications

Limitations aside, there are several implications issued from this work. The most immediate one is that the constructed platform, excluding the visualization module, has benefited two other senior students developing their theses, which helps confirm the research potential the application offers.

Additionally, the platform helped in creating an exploratory analysis of a single retail company, however the same could be easily replicated for any other person, institution, or event, such as politicians, celebrities, public institutions, and other private companies. Further, a comparative analysis including every Chilean retail company would not be so difficult to perform, allowing to obtain valuable information on the competition, which would

otherwise require considerably more resources.

The information extracted by the platform can potentially enable its users to take better-founded decisions. Consider the Cyber Monday event for example. From the retailer's point of view, the information available on Twitter unveiled what aspects of their online platform were the most criticized, whereas from a client's point of view, knowing that the company is not able to offer good service standards for their shopping event, might save him a considerable amount of time and an uncomfortable situation.

Moreover, accepting the fact that Twitter is a good medium for transmitting Word of Mouth, and considering the effect WoM has on brand knowledge, brand relationship, and behavioral outcomes, such as current and future purchases, it only follows that the tool presented in this thesis might be a valuable asset for supporting brand management.

6.4 Future Work

The application can be improved in several ways. Some guidelines for doing so are:

1. Concerning the Data Extraction Module, an interface should be created for managing several instances of the crawler. The first step would be to encapsulate current functionality in a way that is abstract enough for future developers to build upon, or alternatively, use an already-established crawler such as Scrapy. Additionally, a command line interface, and later a graphical user interface, should be created for making user interaction simpler.

2. Regarding the Preprocessing Module, several additions to the way tweets are preprocessed should be created. The most important are creating algorithms for normalizing a wider variety of written laughs ("ja ja," "jejeje," "jksajsa"), for incorporating the analysis of abnormally long words ("I looveeeeee it"), and for better dissambiguating periods (abbreviations, decimal numbers, thousand separators depending of the language, ends of sentences).

3. Furthermore, a more appropriate POS tagger should be implemented, or at least a one that is trained with Twitter data. The hypothesis behind this suggestion is that when trained with more suitable data, the POS tagger will be more accurate and will provide better-quality results to the dependency parser, which in turn will create better dependency trees.

4. As for the Dependency Parser which is the current bottleneck of the whole applications, there are several measures that should be taken. First, a way for interfacing with it, without relying on Input/Output operations, should be created. Now, the way for interfacing with the parser is by writing a file to disk, which is then read and later written by the parser, and finally read again by the program that first called it. This process repeated for every tweet makes the tagging of a big corpus very slow, which is why I/O operations should be completely avoided, or at least minimized. A way to do so would be by enabling MaltParser to process tweets in batches, by rewriting

it in Python, or by switching to a parser that offers a better interface. Besides from this, it would also be useful to tune the parser's parameters for improving classification performance.

5. A new module capable of classifying tweets according to their topic should be implemented. It would be very useful to separate corporate tweets from ads, complaints, news, conversational, and completely irrelevant content. For this, Opinion Mining techniques should be combined with Topic Modeling. Similarly, discriminating among types of account would also be useful.

6. More Natural Language Processing techniques should be incorporated into the pipeline, namely, researching semantic analysis techniques, and how to incorporate them in the process, would greatly help in advancing both the NLP and OM fields.

7. Different Opinion Mining approaches, namely, Machine Learning techniques should also be incorporated into the whole system for attempting to improve classification performance, even if they are more domain-dependent than lexicon-based approaches. Additionally, the combination of lexicon-based, machine-learning-based, and ontology-based approaches should also be researched.

8. The level of granularity at which the platform perform should also be improved. Currently the platform is only capable of classifying tweets at the sentence level, but it would be useful to do so at the aspect level. A first step towards this goal would be by adding an opinion-target detector to the pipeline. In turn, this also would allow future researchers to try different summarization techniques in order to display more usable data to the end-user.

9. A step for detecting and filtering irrelevant, objective and neutral tweets should be added to the pipeline, before the polarity classification stage. This would significantly improve both positive and negative precisions, and the overall classification performance.

10. More efforts should be devoted to creating lexicons containing Twitter jargon, or even better, for creating systems capable of automatically expanding currently-existing lexicons, and adapting to the highly dynamic language found on Twitter.

11. The application's front-end should be considerably improved before attempting to offer it as a service. For instance, better graphs could be created by using more complex libraries such as D3.js. Additionally, the front-end should be designed as a web application allowing higher user interactivity. Furthermore, other options could also be considered, for example, instead of offering a service through a web page, a proprietary software such as Tableau could be used for creating interactive dashboards and offering these as a service.

12. Finally, the ultimate application would be the one that approximates human understanding the best. In order to create such application, massive effort should be devoted to understanding how a human brain decodes the information contained within a text, and then encodes it as knowledge that it can use later. Ontology-based approaches represent the first steps toward reaching this goal.

6.5 Closing Remarks

The objectives proposed at the beginning of this thesis were successfully accomplished. Indeed, the state of the art was investigated and a paper summarizing it was published, the platform was designed, implemented and validated, and a simple visualization system was created for facilitating the interpretation of the results obtained by the rest of the platform. Furthermore the additional, unforeseen objective of creating a RESTful API for the rest of the research group was also achieved.

Additionally, the research hypothesis was partially confirmed. By developing this thesis it was possible to prove that there is a vast amount of user-generated data available on Twitter, and that this data can yield useful results, in fact, by analyzing the results obtained by the platform, the same conclusions as a validated market research, performed by a respectable research firm were reached.

Finally, besides from contributing with the published paper issued from the state-of-the-art-related research, and the API available for the rest of the research group, hopefully the application will be, at least to some extent, useful for future students, and contribute to laying the foundations for future research concerning the understanding of consumer behavior through online channels, and how to legitimately exploit electronic Word of Mouth to benefit both companies and consumers.

Bibliography

[1] Consejo Nacional de la Cultura y las Artes, "Segunda Encuesta Nacional de Participaición y Consumo Cultural," 2011. 2

[2] Consejo Nacional de la Cultura y las Artes, "Tercera Encuesta Nacional de Participaición y Consumo Cultural," 2012. 2

[3] Pew Research Center, "Emerging Nations Embrace Internet, Mobile Technology," 2013. 2, 3

[4] European Commission, "Digital Agenda for Europe." `https://ec.europa.eu/digital-agenda/sites/digital-agenda/files/KKAH12001ENN-PDFWEB_1.pdf`, 2012. Accessed on 2015–02–04. 2

[5] comScore, "Futuro Digital Chile," 2014. 2, 3, 4

[6] comScore, "State of The Internet with a Focus on Chile," 2011. 3, 4

[7] Servicio Nacional del Consumidor - SERNAC, "Estudio Descriptivo del E-Commerce en Chile y Análisis de Reclamos ante SERNAC," 2014. 4, 5, 118

[8] Gfk Adimark, "Índice de Presencia de Marcas en Redes Sociales," 2013. 4, 5, 118

[9] J. A. Balazs and J. D. Velásquez, "Opinion Mining and Information Fusion: A survey," *Information Fusion*, vol. 27, pp. 95–110, January 2016. 7, 8

[10] S. McConnell, "Code Complete," Microsoft Press, 2nd ed., 2004. 7, 119, 131, 133, 134

[11] G. L. Urban and J. R. Hauser, ""Listening in" to Find and Explore new Combinations of Customer Needs," *Journal of Marketing*, vol. 68, no. 2, pp. 72–87, 2004. 10

[12] O. Netzer, R. Feldman, J. Goldenberg, and M. Fresko, "Mine your own business: Market-structure surveillance through text mining," *Marketing Science*, vol. 31, no. 3, pp. 521–543, 2012. 10, 15

[13] B. Liu, "Sentiment analysis and subjectivity," in *Handbook of natural language processing* (R. Dale, H. Moisl, and H. Somers, eds.), Machine Learning and Pattern Recognition, pp. 627–666, New York, NY, USA: CRC Press, 2nd edition ed., 2010. 11

[14] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the Joint Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, (Vancouver, Canada), pp. 347–354, Association for Computational Linguistics, 2005. 11

[15] R. Feldman, "Techniques and applications for sentiment analysis," *Communications of the ACM*, vol. 56, no. 4, pp. 82–89, 2013. 11, 18, 20

[16] G. Vinodhini and R. Chandrasekaran, "Sentiment analysis and opinion mining: a survey," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 6, pp. 282–292, 2012. 11, 18, 19

[17] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 15–21, 2013. 11, 18

[18] A. Neviarouskaya, H. Prendinger, and M. Ishizuka, "Recognition of fine-grained emotions from text: An approach based on the compositionality principle," in *Modeling Machine Emotions for Realizing Intelligence* (T. Nishida, L. C. Jain, and C. Faucher, eds.), vol. 1 of *Smart Innovation, Systems and Technologies*, pp. 179–207, Springer Berlin Heidelberg, 2010. 11

[19] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012. 11, 15, 18, 19, 21, 23

[20] R. Arora and S. Srinivasa, "A faceted characterization of the opinion mining landscape," in *Proceedings of the 6th International Conference on Communication Systems and Networks (COMSNETS 2014)*, (Bangalore, India), pp. 1–6, IEEE, 2014. 13, 19

[21] L. Dey and S. M. Haque, "Opinion mining from noisy text data," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 12, no. 3, pp. 205–226, 2009. 13

[22] F. H. Khan, S. Bashir, and U. Qamar, "TOM: Twitter opinion mining framework using hybrid classification scheme," *Decision Support Systems*, vol. 57, pp. 245–257, 2014. 13

[23] A. Bakliwal, J. Foster, J. van der Puil, R. O'Brien, L. Tounsi, and M. Hughes, "Sentiment analysis of political tweets: Towards an accurate classifier," in *Proceedings of the NAACL Workshop on Language Analysis in Social Media (LASM 2013)*, (Atlanta, GA, USA), pp. 49–58, Association for Computational Linguistics, 2013. 13

[24] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent Twitter sentiment classification," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, vol. 1, (Portland, OR, USA), pp. 151–160, Association for Computational Linguistics, 2011. 13, 50, 117

[25] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of Twitter data," in *Proceedings of the Workshop on Languages in Social Media (LSM 2011)*, (Portland, OR, USA), pp. 30–38, Association for Computational Linguistics, 2011. 13

[26] B. Krishnamurthy, P. Gill, and M. Arlitt, "A few chirps about Twitter," in *Proceedings of the 1st Workshop on Online Social Networks (WOSN 2008)*, (Seattle, WA, USA), pp. 19–24, ACM, 2008. 13, 43

[27] Y. H. Gu and S. J. Yoo, "Rules for Mining Comparative Online Opinions," in *Proceedings of the 4th International Conference on Computer Sciences and Convergence Information Technology (ICCIT 2009)*, (Seoul, Korea), pp. 1294–1299, IEEE, 2009. 13

[28] X. Hu, J. Tang, H. Gao, and H. Liu, "Unsupervised sentiment analysis with emotional signals," in *Proceedings of the 22nd International Conference on World Wide Web (WWW 2013)*, (Rio de Janeiro, Brazil), pp. 607–618, International World Wide Web Conferences Steering Committee, 2013. 13

[29] C. Olston and M. Najork, "Web crawling," *Foundations and Trends in Information Retrieval*, vol. 4, no. 3, pp. 175–246, 2010. 13, 14

[30] K. Guo, L. Shi, W. Ye, and X. Li, "A survey of Internet public opinion mining," in *Proceedings of the International Conference on Progress in Informatics and Computing (PIC 2014)*, (Shanghai, China), pp. 173–179, IEEE, 2014. 13, 15, 19

[31] T. Fu, A. Abbasi, D. Zeng, and H. Chen, "Sentimental spidering: leveraging opinion information in focused crawlers," *ACM Transactions on Information Systems*, vol. 30, no. 4, pp. 24:1–24:30, 2012. 14

[32] A. G. Vural, *Sentiment-Focused Web Crawling*. PhD thesis, Middle East Technical University, 2013. 14

[33] A. Hippisley, "Lexical Analysis," in *Handbook of natural language processing* (R. Dale, H. Moisl, and H. Somers, eds.), Machine Learning and Pattern Recognition, pp. 31–58, New York, NY, USA: CRC Press, 2nd edition ed., 2010. 14, 29

[34] D. Palmer, "Text Preprocessing," in *Handbook of natural language processing* (R. Dale, H. Moisl, and H. Somers, eds.), Machine Learning and Pattern Recognition, pp. 9–30, New York, NY, USA: CRC Press, 2nd edition ed., 2010. 14, 25, 26, 28

[35] B. Liu, *Web data mining: exploring hyperlinks, contents, and usage data*, ch. 6. New York, NY, USA: Springer Berlin Heidelberg, 1st edition ed., 2007. 14

[36] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 40, no. 3, pp. 211–218, 2006. 14, 29

[37] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*, vol. 1, ch. 2. Cambridge University Press, 2008. 14

[38] T. Kiss and J. Strunk, "Unsupervised multilingual sentence boundary detection," *Computational Linguistics*, vol. 32, no. 4, pp. 485–525, 2006. 14

[39] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python*. O'Reilly Media, Inc., 1st edition ed., 2009. 14, 30, 41, 77

[40] E. Brill, "A simple rule-based part of speech tagger," in *Proceedings of the Workshop on Speech and Natural Language*, (Harriman, New York), pp. 112–116, Association for Computational Linguistics, 1992. 15, 33

[41] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith, "Part-of-speech tagging for Twitter: Annotation, features, and experiments," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, vol. 2, (Portland, Oregon), pp. 42–47, Association for Computational Linguistics, 2011. 15

[42] T. Güngör, "Part-of-Speech Tagging," in *Handbook of natural language processing* (R. Dale, H. Moisl, and H. Somers, eds.), Machine Learning and Pattern Recognition, pp. 9–30, New York, NY, USA: CRC Press, 2nd edition ed., 2010. 15, 33, 34

[43] D. Vilares, M. A. Alonso, and C. Gómez-Rodríguez, "A syntactic approach for opinion mining on Spanish reviews," *Natural Language Engineering*, vol. 21, no. 01, pp. 139–163, 2015. 15, 21, 41, 61, 63, 64, 65, 66, 75, 76, 77, 103, 117

[44] M. Joshi and C. Penstein-Rosé, "Generalizing dependency features for opinion mining," in *Proceedings of the 4th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2009)*, (Suntec, Singapore), pp. 313–316, Association for Computational Linguistics, 2009. 15, 21, 37

[45] V. Hangya and R. Farkas, "Target-oriented opinion mining from tweets," in *Proceedings of the 4th International Conference on Cognitive Infocommunications (CogInfoCom 2013)*, (Budapest, Hungary), pp. 251–254, IEEE, 2013. 15

[46] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, vol. 10, (Philadelphia, PA, USA), pp. 79–86, Association for Computational Linguistics, 2002. 15, 20, 21, 22, 49

[47] D. Vilares, M. Á. Alonso, and C. Gómez-Rodríguez, "Supervised polarity classification of Spanish tweets based on linguistic knowledge," in *Proceedings of the 13th Symposium on Document Engineering (DocEng 2013)*, (Florence, Italy), pp. 169–172, ACM, 2013. 15, 52

[48] A. Bakliwal, P. Arora, S. Madhappan, N. Kapre, M. Singh, and V. Varma, "Mining Sentiments from Tweets," in *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2012)*, (Jeju, Korea), pp. 11–18, Association for Computational Linguistics, July 2012. 15

[49] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008. 15, 18, 23, 90, 102

[50] Y. Seki, K. Eguchi, and N. Kando, "Analysis of multi-document viewpoint summarization using multi-dimensional genres," in *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, pp. 142–145, 2004. 15, 16

[51] H. D. Kim, K. Ganesan, P. Sondhi, and C. Zhai, "Comprehensive Review Of Opinion Summarization," tech. rep., University of Illinois at Urbana-Champaign, 2011. 15, 16, 18

[52] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004)*, (Barcelona, Spain), pp. 271–278, Association for Computational Linguistics, 2004. 15, 22

[53] E. Hovy and C.-Y. Lin, "Automated text summarization and the SUMMARIST system," in *Proceedings of a Workshop on Held at Baltimore, Maryland: October 13-15, 1998*, pp. 197–214, Association for Computational Linguistics, 1998. 16

[54] K. Ganesan, C. Zhai, and J. Han, "Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions," in *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 340–348, Association for Computational Linguistics, 2010. 16

[55] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 168–177, ACM, 2004. 16, 20, 21

[56] G. Carenini, R. T. Ng, and A. Pauls, "Multi-Document Summarization of Evaluative Text," in *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 3–7, 2006. 16

[57] H. D. Kim and C. Zhai, "Generating comparative summaries of contradictory opinions in text," in *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 385–394, ACM, 2009. 16

[58] G. Mishne and M. De Rijke, "MoodViews: Tools for Blog Mood Analysis," in *Proceedings of the AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pp. 153–154, 2006. 17

[59] G. Draper and R. F. Riesenfeld, "Who votes for what? a visual query language for opinion data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1197–1204, 2008. 17

[60] Y. Wu, F. Wei, S. Liu, N. Au, W. Cui, H. Zhou, and H. Qu, "OpinionSeer: interactive visualization of hotel customer feedback," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1109–1118, 2010. 17

[61] C. Wang, Z. Xiao, Y. Liu, Y. Xu, A. Zhou, and K. Zhang, "SentiView: Sentiment analysis and visualization for internet popular topics," *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 6, pp. 620–630, 2013. 17

[62] I. Khozyainov, E. Pyshkin, and V. Klyuev, "Spelling out opinions: Difficult cases of sentiment analysis," in *Proceedings of the International Joint Conference on Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA 2013)*, (Aizu-Wakamatsu, Japan), pp. 231–237, IEEE, 2013. 18

[63] D. Maynard, K. Bontcheva, and D. Rout, "Challenges in developing opinion mining tools for social media," in *Proceedings of the LREC workshop @NLP can u tag #usergeneratedcontent?! (LREC 2012)*, (Istambul, Turkey), pp. 15–22, 2012. 18

[64] E. Martínez-Cámara, M. T. Martín-Valdivia, L. A. Urena-López, and A. Montejo-Ráez, "Sentiment analysis in Twitter," *Natural Language Engineering*, vol. 20, no. 01, pp. 1–28, 2014. 18, 42, 44, 48, 52

[65] E. Marrese-Taylor, C. Rodríguez, J. D. Velásquez, G. Ghosh, and S. Banerjee, "Web Opinion Mining and Sentimental Analysis," in *Advanced Techniques in Web Intelligence-2*, Studies in Computational Intelligence, pp. 105–126, Springer Berlin Heidelberg, 2013. 18

[66] N. Medagoda, S. Shanmuganathan, and J. Whalley, "A comparative analysis of opinion mining and sentiment classification in non-English languages," in *Proceedings of the International Conference on Advances in ICT for Emerging Regions (ICTer 2013)*, (Colombo, Sri Lanka), pp. 144–148, IEEE, 2013. 19

[67] V. Singh and S. K. Dubey, "Opinion mining and analysis: A literature review," in *Proceedings of the 5th International Conference – Confluence The Next Generation Information Technology Summit (CONFLUENCE 2014)*, (Noida, India), pp. 232–239, IEEE, 2014. 19

[68] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005)*, (Ann Arbor, MI, USA), pp. 115–124, Association for Computational Linguistics, 2005. 20

[69] E. Riloff and J. Wiebe, "Learning extraction patterns for subjective expressions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2003)*, (Sapporo, Japan), pp. 105–112, Association for Computational Linguistics, 2003. 20

[70] H. Yu and V. Hatzivassiloglou, "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2003)*, (Sapporo, Japan), pp. 129–136, Association for Computational Linguistics, 2003. 20

[71] A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," in *Natural Language Processing and Text Mining* (A. Kao and S. R. Poteet, eds.), pp. 9–28, Springer London, 2007. 20

[72] E. Marrese-Taylor, J. D. Velásquez, F. Bravo-Marquez, and Y. Matsuo, "Identifying customer preferences about tourism products using an aspect-based opinion mining approach," *Procedia Computer Science*, vol. 22, pp. 182–191, 2013. 20

[73] E. Marrese-Taylor, J. D. Velásquez, and F. Bravo-Marquez, "Opinion Zoom: A modular tool to explore tourism opinions on the Web," in *Proceedings of the the IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT 2013)*, (Atlanta, GA, USA), pp. 261–264, IEEE, 2013. 20

[74] Y. Wu, Q. Zhang, X. Huang, and L. Wu, "Phrase dependency parsing for opinion mining," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, vol. 3, (Singapore), pp. 1533–1541, Association for Computational Linguistics, 2009. 21

[75] T. Nakagawa, K. Inui, and S. Kurohashi, "Dependency tree-based sentiment classification using CRFs with hidden variables," in *Proceedings of The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2010)*, (Los Angeles, CA, USA), pp. 786–794, Association for Computational Linguistics, 2010. 21

[76] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2002)*, (Philadelphia, PA, USA), pp. 417–424, Association for Computational Linguistics, 2002. 21, 49

[77] G. A. Miller, "WordNet: A lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995. 21

[78] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, 2011. 21, 63, 66, 81, 85

[79] T. Mitchell, *Machine Learning*. McGraw-Hill, 1997. 21

[80] A. Pak and P. Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining," in *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, vol. 10, (Valletta, Malta), pp. 1320–1326, European Language Resources Association, 2010. 22, 50

[81] D. Davidov, O. Tsur, and A. Rappoport, "Enhanced sentiment learning using Twitter hashtags and smileys," in *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, (Beijing, China), pp. 241–249, Association for Computational Linguistics, 2010. 22, 50

[82] L. T. Nguyen, P. Wu, W. Chan, W. Peng, and Y. Zhang, "Predicting collective sentiment dynamics from time-series social media," in *Proceedings of the 1st International Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM 2012)*, no. 6, (Beijing, China), pp. 6:1–6:8, ACM, 2012. 22

[83] W. Peng and D. H. Park, "Generate adjective sentiment dictionary for social media sentiment analysis using constrained nonnegative matrix factorization," in *Proceedings of the 5th International Conference on Weblogs and Social Media (ICWSM 2011)*, (Barcelona, Spain), pp. 273–280, AAAI Press, 2011. 22

[84] L. Zhou and P. Chaovalit, "Ontology-supported polarity mining," *Journal of the American Society for Information Science and Technology*, vol. 59, no. 1, pp. 98–110, 2008. 22

[85] E. Cambria, R. Speer, C. Havasi, and A. Hussain, "SenticNet: A Publicly Available Semantic Resource for Opinion Mining," in *Proceedings of the Fall Symposium on Computational Models of Narrative*, (Arlington, VA, USA), pp. 14–18, AAAI, 2010. 22

[86] S. Poria, A. Gelbukh, A. Hussain, N. Howard, D. Das, and S. Bandyopadhyay, "Enhanced SenticNet with affective labels for concept-based opinion mining," *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 31–38, 2013. 22

[87] C. Strapparava, A. Valitutti, *et al.*, "WordNet Affect: An Affective Extension of WordNet," in *In Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, vol. 4, (Lisbon, Portugal), pp. 1083–1086, European Language Resources Association, 2004. 22

[88] Q. Miao, Q. Li, and D. Zeng, "Mining fine grained opinions by using probabilistic models and domain knowledge," in *Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technologies (WI-IAT 2010)*, vol. 1, (Toronto, Canada), pp. 358–365, IEEE, 2010. 22

[89] M. Grassi, E. Cambria, A. Hussain, and F. Piazza, "Sentic Web: A new paradigm for managing social media affective information," *Cognitive Computation*, vol. 3, no. 3, pp. 480–489, 2011. 22

[90] A. Aue and M. Gamon, "Customizing sentiment classifiers to new domains: A case study," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2005)*, (Borovets, Bulgaria), 2005. 22

[91] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, (Prague, Czech Republic), pp. 440–447, Association for Computational Linguistics, 2007. 22

[92] K. Mote, "Natural Language Processing - A Survey," *Computing Research Repository*, 2012. 23, 29, 34, 39, 40, 41

[93] N. Indurkhya and F. J. Damerau, eds., *Handbook of Natural Language Processing*. CRC Press, 2nd ed., 2010. 24, 42

[94] R. Dale, "Classical Approaches to Natural Language Processing," in *Handbook of natural language processing* (N. Indurkhya and F. J. Damerau, eds.), pp. 3–7, CRC Press, 2nd ed., 2010. 25

[95] J. Goyvaerts and S. Levithan, *Regular Expressions Cookbook*. O'Reilly, 2nd ed., 2012. 27

[96] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, 1st ed., 2000. 29, 32, 34, 39, 40, 41

[97] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*, vol. 1. Cambridge, England: Cambridge University Press, 2008. 29, 31, 95

[98] Ref. 10, pp. 11–12. 29

[99] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *Proceedings of the International Conference on New Methods in Language Processing*, (Manchester, UK), 1994. 30, 41, 75, 76

[100] A. Voutilainen, "A syntax-based part-of-speech analyser," in *Proceedings of the 7th Conference on European Chapter of the Association for Computational Linguistics*, pp. 157–164, Morgan Kaufmann, 1995. 32

[101] B. Merialdo, "Tagging English text with a probabilistic model," *Computational linguistics*, vol. 20, no. 2, pp. 155–171, 1994. 33

[102] A. Ratnaparkhi, *Maximum entropy models for natural language ambiguity resolution*. PhD thesis, University of Pennsylvania, 1998. 33

[103] E. Brill, "Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging," *Computational Linguistics*, vol. 21, no. 4, pp. 543–565, 1995. 33

[104] N. Chomsky, "Three models for the description of language," *IRE Transactions on Information Theory*, vol. 2, no. 3, pp. 113–124, 1956. 36

[105] P. Lungjölf and M. Wirén, "Syntactic Parsing," in *Handbook of natural language processing* (N. Indurkhya and F. J. Damerau, eds.), pp. 59–92, CRC Press, 2nd ed., 2010. 36, 37

[106] M.-C. De Marneffe and C. D. Manning, "Stanford typed dependencies manual." `http://nlp.stanford.edu/software/dependencies_manual.pdf`, 2008. Accessed on 2015–04–14. 37

[107] M. A. Martí, M. Taulé, M. Bertran, and L. Màrquez, "Ancora: Multilingual and multilevel annotated corpora." `http://clic.ub.edu/corpus/webfm_send/13`, 2007. Accessed on 2015–04–14. 37, 39, 77, 152, 153, 154

[108] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi, "MaltParser: A language-independent system for data-driven dependency parsing," *Natural Language Engineering*, vol. 13, no. 02, pp. 95–135, 2007. 37, 76

[109] C. Goddard and A. C. Schalley, "Semantic Analysis," in *Handbook of natural language processing* (N. Indurkhya and F. J. Damerau, eds.), pp. 93–120, CRC Press, 2nd ed., 2010. 39, 40, 41

[110] X. Liu, S. Zhang, F. Wei, and M. Zhou, "Recognizing named entities in tweets," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 359–367, Association for Computational Linguistics, 2011. 41

[111] E. F. Tjong Kim Sang and S. Buchholz, "Introduction to the CoNLL-2000 shared task: Chunking," in *Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning*, vol. 7, pp. 127–132, Association for Computational Linguistics, 2000. 41

[112] E. Marresse Taylor, "Diseño e implementación de una aplicación de Web Opinion Mining para identificar preferencias de usuarios sobre productos turísticos de la X región de Los Lagos." Tesis de Pregrado, Universidad de Chile, Departamento de Ingeniería Industrial, 2013. 41

[113] M.-C. De Marneffe, B. MacCartney, C. D. Manning, *et al.*, "Generating typed dependency parses from phrase structure parses," in *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, vol. 6, pp. 449–454, 2006. 41

[114] J. Nivre, J. Hall, and J. Nilsson, "Maltparser: A data-driven parser-generator for dependency parsing," in *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, vol. 6, pp. 2216–2219, 2006. 41

[115] D. Zhao and M. B. Rosson, "How and why people Twitter: the role that micro-blogging plays in informal communication at work," in *Proceedings of the ACM 2009 International Conference on Supporting Group Work*, pp. 243–252, ACM, 2009. 43

[116] Pew Research Center, "Social Media Update 2014," January 2015. `http://www.pewinternet.org/files/2015/01/PI_SocialMediaUpdate20144.pdf` Accessed August 31, 2015. 43

[117] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury, "Twitter power: Tweets as electronic word of mouth," *Journal of the American society for information science and technology*, vol. 60, no. 11, pp. 2169–2188, 2009. 44, 46, 47

[118] J. Kietzmann and A. Canhoto, "Bittersweet! Understanding and managing electronic word of mouth," *Journal of Public Affairs*, vol. 13, no. 2, pp. 146–159, 2013. 44, 45

[119] T. Hennig-Thurau, K. P. Gwinner, G. Walsh, and D. D. Gremler, "Electronic word-of-mouth via consumer-opinion platforms: what motivates consumers to articulate themselves on the internet?," *Journal of Interactive Marketing*, vol. 18, no. 1, pp. 38–52, 2004. 44

[120] C. Dellarocas, "The digitization of word of mouth: Promise and challenges of online feedback mechanisms," *Management Science*, vol. 49, no. 10, pp. 1407–1424, 2003. 44, 45

[121] C. Park and T. M. Lee, "Information direction, website reputation and eWOM effect: A moderating role of product type," *Journal of Business Research*, vol. 62, no. 1, pp. 61–67, 2009. 45

[122] R. Kelly, "Twitter Study Reveals Interesting Results About Usage – 40% is Pointless Babble." `http://pearanalytics.com/blog/2009/twitter-study-reveals-interesting-results-40-percent-pointless-babble/`, 2009. Accessed on 2015–04–21. 45, 118

[123] C. Campbell, N. Piercy, and D. Heinrich, "When companies get caught: The effect of consumers discovering undesirable firm engagement online," *Journal of Public Affairs*, vol. 12, no. 2, pp. 120–126, 2012. 45

[124] B. Barton, "Ratings, reviews & ROI: How leading retailers use customer word of mouth in marketing and merchandising," *Journal of Interactive Advertising*, vol. 7, no. 1, pp. 47–50, 2006. 46

[125] F.-R. Esch, T. Langner, B. H. Schmitt, and P. Geus, "Are brands forever? How brand knowledge and relationships affect current and future purchases," *Journal of Product & Brand Management*, vol. 15, no. 2, pp. 98–105, 2006. 46

[126] F. Aisopos, G. Papadakis, K. Tserpes, and T. Varvarigou, "Content vs. context for sentiment analysis: a comparative analysis over microblogs," in *Proceedings of the 23rd ACM Conference on Hypertext and Social Media*, pp. 187–196, ACM, 2012. 48

[127] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003. 48

[128] J. Read, "Using emoticons to reduce dependency in machine learning techniques for sentiment classification," in *Proceedings of the ACL Student Research Workshop*, (Ann Arbor, Michigan, United States), pp. 43–48, Association for Computational Linguistics, 2005. 49, 50

[129] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, pp. 1–12, 2009. 49

[130] L. Barbosa and J. Feng, "Robust sentiment detection on Twitter from biased and noisy data," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 36–44, Association for Computational Linguistics, 2010. 50

[131] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment in Twitter events," *Journal of the American Society for Information Science and Technology*, vol. 62, no. 2, pp. 406–418, 2011. 51, 118

[132] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 12, pp. 2544–2558, 2010. 51

[133] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu, and B. Liu, "Combining lexicon-based and learning-based methods for twitter sentiment analysis," Tech. Rep. HPL-2011-89, HP Laboratories, 2011. 51, 52

[134] D. Vilares, M. A. Alonso, and C. Gómez-Rodríguez, "Clasificación de polaridad en textos con opiniones en español mediante análisis sintáctico de dependencias," *Procesamiento del Lenguaje Natural*, vol. 50, pp. 13–20, 2013. 52

[135] Ref. 10, pp. 23–60. 53, 54

[136] J. Brooke, M. Tofiloski, and M. Taboada, "Cross-Linguistic Sentiment Analysis: From English to Spanish," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2009)*, (Borovets, Bulgaria), pp. 50–54, Association for Computational Linguistics, September 2009. 66

[137] E. S. Raymond, *The Art of Unix Programming*. Addison-Wesley, 1st ed., 2003. 74

[138] MongoDB, "Top 5 Considerations When Evaluating NoSQL Databases," *White Paper*, June 2015. URL `https://s3.amazonaws.com/info-mongodb-com/10gen_Top_5_NoSQL_Considerations.pdf`. 74

[139] H. Schmid, "Improvements in part-of-speech tagging with an application to German," in *Proceedings of EACL-SIGDAT*, (Dublin, Ireland), 1995. 76

[140] B. O'Connor, M. Krieger, and D. Ahn, "TweetMotif: Exploratory Search and Topic Summarization for Twitter," in *Proceedings of the 4th Internation Conference on Weblogs and Social Media (ICWSM)*, (Washington, DC, USA), AAAI Press, 2010. 82

[141] Ref. 10, pp. 100–101. 93

[142] A. Bifet and E. Frank, "Sentiment knowledge discovery in Twitter streaming data," in *Discovery Science*, pp. 1–15, Springer, 2010. 96

[143] A. J. Viera and J. M. Garrett, "Understanding Interobserver Agreement: The Kappa Statistic," *Family Medicine*, vol. 37, no. 5, pp. 360–363, 2005. 96, 158

[144] J. Villena Román, S. Lana Serrano, E. Martínez Cámara, and J. C. González Cristóbal, "TASS-Workshop on sentiment analysis at SEPLN," *Procesamiento del Lenguaje Natural*, vol. 50, pp. 37–44, March 2013. 96, 97

[145] M. Weisfeld, "The Object-Oriented Thought Process," Addison-Wesley, 4th ed., 2013. 119

[146] A. Hunt and D. Thomas, "The Pragmatic Programmer: From Journeyman to Master," Addison-Wesley, 1st ed., 1999. 119

Appendix

A Paper: "Opinion Mining and Information Fusion: A survey"

See next page.

Information Fusion 27 (2016) 95–110

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Opinion Mining and Information Fusion: A survey

Jorge A. Balazs, Juan D. Velásquez *

Department of Industrial Engineering, Universidad de Chile, Av. República 701, P.O. Box: 8370439, Santiago, Chile

ARTICLE INFO

Article history:
Received 31 March 2015
Received in revised form 4 June 2015
Accepted 8 June 2015
Available online 17 June 2015

Keywords:
Information Fusion
Survey
Opinion Mining
Sentiment Analysis

ABSTRACT

Interest in Opinion Mining has been growing steadily in the last years, mainly because of its great number of applications and the scientific challenge it poses. Accordingly, the resources and techniques to help tackle the problem are many, and most of the latest work fuses them at some stage of the process. However, this combination is usually executed without following any defined guidelines and overlooking the possibility of replicating and improving it, hence the need for a deeper understanding of the fusion process becomes apparent. Information Fusion is the field charged with researching efficient methods for transforming information from different sources into a single coherent representation, and therefore can be used to guide fusion processes in Opinion Mining. In this paper we present a survey on Information Fusion applied to Opinion Mining. We first define Opinion Mining and describe its most fundamental aspects, later explain Information Fusion and finally review several Opinion Mining studies that rely at some point on the fusion of information.

1. Introduction

With the advent of the Web 2.0 and its continuous growth, the amount of freely available user-generated data has reached an unprecedented volume. Being so massive, it is impossible for humans to make sense of its whole in a reasonable amount of time, which is why there has been a growing interest in the scientific community to create systems capable of extracting information from it.

Moreover, the diversity of available data in terms of content, format and extension is huge. Indeed, the data available in microblogs such as Twitter are short and written without much concern for grammar, while review-related data are more extensive and follow stricter grammatical rules [1]. So it is also necessary to bear these differences in mind when attempting to perform any kind of analysis.

In this work, we will focus on two fields charged with dealing with the aforementioned problems, Opinion Mining (OM) and Information Fusion (IF). Opinion Mining (also known as *Sentiment Analysis* [2,3]) is a sub-field of text mining in which the main task is to extract opinions from content generated by Web users. Opinions play a fundamental role in the decision-making process of both individuals and organizations since they deeply influence people's attitudes and beliefs [4]. Such is the interest in harnessing the power to automatically detect and understand opinions that today this field is one of the most popular areas of research in the Natural Language Processing (NLP) and Computer Science communities, with more than 7000 articles published [5].

To mention some examples, mining opinions enables e-commerce businesses to gain deeper knowledge of their customers and products without having to pay for surveys [6], it allows politicians to understand the political sentiment of the community towards them without having to rely on polls [7], lets companies anticipate their stock trading volumes and financial returns [8], and helps strengthening the deliberation process in the public policy context [9].

Additionally, extracting opinions from reviews, blogs and microblogs, combined with the fusion of different sources of information presents several advantages such as higher authenticity, reduced ambiguity and greater availability [10]. Information Fusion is defined as "the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human or automated decision making" [11]. Most of the research in Information Fusion has been done in fields related to the military where data is generated by electronic sensors, however there is growing interest in the fusion of data generated by humans (also called *soft* data) [10,12].

In this paper we attempt to review the state of the art in Opinion Mining studies that explicitly or implicitly use the fusion of information. Our aim is to provide both new and experienced researchers with insights on how to better perform the fusion

* Corresponding author. Tel.: +56 2 2978 4834; fax: +56 2 2689 7895.
E-mail addresses: jabalazs@ug.uchile.cl (J.A. Balazs), jvelasqu@dii.uchile.cl (J.D. Velásquez).

http://dx.doi.org/10.1016/j.inffus.2015.06.002
1566-2535/

process in an Opinion Mining context while also supplying enough information to help them understand both fields separately.

The remainder of this work is structured as follows: In Section 2 we show an overview of Opinion Mining by formally defining it, describing the usual process pipeline, explaining the different levels of analysis at which it performs, the different approaches that it uses and the most common challenges it faces. In Section 3 we review the state of the art in Opinion Mining combined with Information Fusion and present a simple framework for guiding the fusion process in the Opinion Mining context. Finally, in Section 4 we present some of the reviews that have been published both for Opinion Mining and Information Fusion.

2. Opinion Mining

Merriam-Webster's Online Dictionary[1] defines an opinion as a belief, judgement or way of thinking about something. Opinions are formed by the experiences lived by those who hold them. A consumer may look for another's opinion before buying a product or deciding to watch a movie, to gain insights into the potential experiences they would have depending on the decisions they make. Moreover, businesses could benefit from knowing the opinions of their customers by discovering cues on what aspects of a certain service to improve, which features of a determined product are the most valued, or which are new potential business opportunities [13,14]. In essence, a good Opinion Mining system could eliminate the need for polls and change the way traditional market research is done.

2.1. Definition

Opinion Mining is the field charged with the task of extracting opinions from unstructured text by combining techniques from NLP and Computer Science.

Liu [15] defines an opinion as a 5-tuple containing the target of the opinion (or *entity*), the attribute of the target at which the opinion is directed, the sentiment (or polarity) contained in the opinion which can be positive, negative or neutral, the opinion holder and the date when the opinion was emitted. Formally, an opinion is defined as a tuple:

$$(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$$

where e_i is the ith opinion target, a_{ij} is the jth attribute of e_i, h_k is the kth opinion holder, t_l is the time when the opinion was emitted and s_{ijkl} is the polarity of the opinion towards the attribute a_{ij} of entity e_i by the opinion holder h_k at time t_l.

Note that we described the sentiment contained in an opinion as positive, negative or neutral, notwithstanding it could also be numerically represented. For instance -5 could denote a very negative opinion while 5 a very positive one. Also, in case the analysis did not require much level of detail, the attributes of an entity could be omitted and denoted by *GENERAL* instead of a_{ij}.

Therefore the main objective of Opinion Mining is to find all the opinion tuples $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$ within a document, collection of documents (called *corpus*) or across many corpora. Other works define Opinion Mining as "the task of identifying positive and negative opinions, emotions and evaluations" [16], "the task of finding the opinions of authors about specific entities" [5], "tracking the mood of the public about a particular product or topic" [17], or simply "the task of polarity classification" [18]. These definitions present different scopes and levels of granularity, however all of them can be adapted to fit Liu's opinion model.

There are other approaches, like the one presented in [19], in which the authors attempt to classify emotional states such as "anger", "fear", "joy", or "interest" instead of just positive or negative. In this case, Liu's model could be enriched by adding another element to the opinion tuple model to represent this information.

2.2. Opinion Mining process: previous steps

The usual Opinion Mining process or pipeline usually consists of a series of defined steps [20–22]. These correspond to corpus or data acquisition, text preprocessing, Opinion Mining core process, aggregation and summarization of results, and visualization. In this paper we will give an overview of the first three. Particularly, in this section we will briefly review the two first steps previous to the core OM process: data acquisition and text preprocessing.

2.2.1. Data acquisition

The first step of any Opinion Mining pipeline is called corpus or data acquisition and consists of obtaining the corpus that is going to be mined for opinions. Currently there are two approaches to achieving this task. The first is through a website's Application Programming Interface (API) being Twitter's[2] one of the most popular [22–25]. The second corresponds to the use of Web crawlers in order to scrape the data from the desired websites [26–28]. Olston and Najork portray a robust survey of Web crawling in [29].

Both approaches present some advantages and disadvantages so there is a trade-off between using either. In [30] the authors briefly compare them.

With the API-based approach the implementation is easy, the data gathered is ordered and unlikely to change its structure, however it presents some limitations depending on the provider. For instance search queries to the Twitter REST API are limited to 180 per 15-min time window.[3] Additionally, the Streaming API has no explicit rate limits for downloading tweets, but is limited in other aspects such as the number of clients from the same IP address connected at the same time, and the rate at which clients are able to read data.[4] This approach is also subject to the availability of an API since not all websites provide one, and even if they do it might not present every needed functionality.

In contrast, crawler-based approaches are more difficult to implement, since the data obtained is noisier and its structure is prone to change, but have the advantage of being virtually unrestricted. Still, using these approaches requires to respect some good etiquette protocols such as the *robots exclusion standard*,[5] not issuing multiple overlapping requests to the same server and spacing these requests to prevent putting too much strain on it [29]. Furthermore, Web crawlers can prioritize the extraction of subjective and topically-relevant content. In [31], the authors propose a focused crawler that collects opinion-rich content regarding a particular topic and in [32] this work is further developed by proposing a formal definition for sentiment-based Web crawling along with a framework to facilitate the discovery of subjective content.

2.2.2. Text preprocessing

The second step in the OM pipeline is Text Preprocessing and is charged with common NLP tasks associated with lexical analysis [33]. Some of the most common techniques are:

Tokenization: task for separating the full text string into a list of separate words. This is simple to perform in space-delimited languages such as English, Spanish or French, but becomes

[1] http://www.merriam-webster.com/dictionary/opinion (Visited May 11, 2015).

[2] https://dev.twitter.com/rest/public (Visited May 11, 2015).

[3] https://dev.twitter.com/rest/public/rate-limiting (Visited May 11, 2015).

[4] https://dev.twitter.com/streaming/overview/connecting (Visited May 11, 2015).

[5] http://www.robotstxt.org/robotstxt.html (Visited May 11, 2015).

considerably more difficult in languages where words are not delimited by spaces like in Japanese, Chinese and Thai [34].

Stemming: heuristic process for deleting word affixes and leaving them in an invariant canonical form or "stem" [35]. For instance, *person*, *person's*, *personify* and *personification* become *person* when stemmed. The most popular English stemmer algorithm is Porter's stemmer [36].

Lemmatization: algorithmic process to bring a word into its non-inflected dictionary form. It is analogous to stemming but is achieved through a more rigorous set of steps that incorporate the morphological analysis of each word [37].

Stopword Removal: activity for removing words that are used for structuring language but do not contribute in any way to its content. Some of these words are *a*, *are*, *the*, *was* and *will*.[6]

Sentence Segmentation: procedure for separating paragraphs into sentences [38]. This step presents its own challenges since periods are often used to mark the ending of a sentence but also to denote abbreviations and decimal numbers [39].

Part-of-Speech (POS) Tagging: is the step of labeling each word of a sentence with its part of speech, such as *adjective*, *noun*, *verb*, *adverb* and *preposition* [40–42], either to be used as input for further processing like dependency parsing [43] or to be used as features for a machine learning process [44,45].

Note that all of these steps are not always necessary and have to be selected accordingly for every Opinion Mining application. For example, a machine-learning-based system that relies on a bag-of-words approach will probably use all of the mentioned methods in order to reduce dimensionality and noise [46], while an unsupervised approach might need some of the stopwords' parts of speech to build the dependency rules later used in the Opinion Mining core process [43] therefore omitting the stopword removal process. We present a more detailed analysis of supervised versus unsupervised OM approaches in Section 2.3.2.

Moreover, there are other steps that depend heavily on the data source and acquisition method. In particular, data obtained through a Web crawler will have to be processed to remove HTML tags and nontextual information (images and ads) [14,30,47], and text extracted from Twitter will need special care for hashtags, mentions, retweets, poorly written text, emoticons, written laughs, and words with repeated characters [46,48,49].

2.3. Opinion Mining process: core

The third phase in the pipeline is the Opinion Mining core process. In this section we will review the levels of granularity at which it is performed and the different approaches utilized.

2.3.1. Levels of analysis

Since Opinion Mining began to rise in popularity, the sought-after level of analysis has passed through several stages. First it was performed at the document level where the objective was to find the general polarity of the whole document. Then, the interest shifted to the sentence level and finally to the entity and aspect level. It is worth noting that the analyses that are more fine-grained can be aggregated to form the higher levels. For example an aspect-based Opinion Mining process could simply calculate the average sentiment in a given sentence to produce a sentence-level result.

Document Level: Opinion Mining at this level of analysis attempts to classify an opinionated document into positive or negative. The applicability of this level is often limited and usually resides within the context of review analysis [4]. Formally, the objective in the document-level Opinion Mining task can be defined as a modified version of the one presented in Section 2.1 and corresponds to finding the tuples:

$$(-, GENERAL, s_{GENERAL}, -, -)$$

where the entity e, opinion holder h, and the time when the opinion was stated t are assumed known or ignored, and the attribute a_j of the entity e corresponds to *GENERAL*. This means that the analysis will only return the generalized polarity of the document. To give a few examples, in [47], Pang and Lee attempted to predict the polarity of movie reviews using three different machine learning techniques: Naïve Bayes, Maximum Entropy classification and Support Vector Machine (SVM). Similarly, in [50] the same authors tried to predict the rating of a movie given in a review, instead of just classifying the review into a positive or negative class.

Sentence Level: This level is analogous to the previous one since a sentence can be considered as a short document. However, it presents the additional preprocessing step consisting of breaking the document into separate sentences, which in turn poses challenges similar to tokenization in languages not delimited by periods. In [51] Riloff and Wiebe used heuristics to automatically label previously unknown data and discover extraction patterns to extract subjective sentences. In [52] the authors achieved high recall and precision (80–90%) for detecting opinions in sentences by using a naïve Bayes classifier and including words, bigrams, trigrams, part-of-speech tags and polarity in the feature set.

Entity and Aspect Level: This represents the most granular level at which Opinion Mining is performed. Here, the task is not only to find the polarity of the opinion but also its target (entity, aspect or both), hence the 5-tuple definition described in Section 2.1 fully applies. Both document-level and sentence-level analyses work well when the text being examined contains a single entity and aspect, but they falter when more are present [5]. Aspect-based Opinion Mining attempts to solve this problem by detecting every mentioned aspect in the text and associating them with an opinion.

The earliest work addressing this problem is [6] in which Hu and Liu detect product features (aspects) frequently commented on by customers, then identify the sentences containing opinions, assess their polarity and finally summarize the results. Likewise, in [53] the process to perform the aspect-based Opinion Mining task is to first identify product features, then identify the opinions regarding these features, later estimate their polarity and finally rank them based on their strength.

Marrese-Taylor et al. [54] extend the opinion definition provided by Bing Liu by incorporating *entity expressions* and *aspect expressions* into the analysis. Later they follow the steps of aspect identification, sentiment prediction and summary generation and apply their methodology to the tourism domain by mining opinions from TripAdvisor reviews. They achieved high precision and recall (90%) in the sentiment polarity extraction task but were only able to extract 35% of the explicit *aspect expressions*. In [55], the authors further developed their methodology and integrated it into a modular software that considers all of the previous steps with the addition of a visualization module.

2.3.2. Different approaches

There are two well-established approaches to carry out the OM core process. One is the unsupervised lexicon-based approach, where the process relies on rules and heuristics obtained from linguistic knowledge [43], and the other is the supervised machine learning approach where algorithms learn underlying information from previously annotated data, allowing them to classify new, unlabeled data [47]. There have also been a growing number of studies reporting the successful combination of both approaches [44,56,57]. Furthermore there is an emerging trend that uses

[6] For a more complete list, visit: http://snowball.tartarus.org/algorithms/english/stop.txt (Visited May 11, 2015).

ontologies to address the Opinion Mining problem. This is called concept-based Opinion Mining.

Unsupervised Lexicon-based Approaches: Also called semantic-based approaches, attempt to determine the polarity of text by using a set of rules and heuristics obtained from language knowledge. The usual steps to carry them out are first, to mark each word and phrase with its corresponding sentiment polarity with the help of a lexicon, second, to incorporate the analysis of sentiment shifters and their scope (intensifiers and negation), and finally, to handle the adversative clauses (*but-clauses*) by understanding how they affect polarity and reflecting this in the final sentiment score [4]. Later steps could include opinion summarization and visualization.

The first study to tackle Opinion Mining in an unsupervised manner was [58], in which the author created an algorithm that first extracts bigrams abiding certain grammatical rules, then estimates their polarity using the Pointwise Mutual Information (PMI) and finally, computes the average polarity of every extracted bigram to estimate the overall polarity of a review. In [6], Hu and Liu created a list of opinion words using WordNet [59] to later predict the orientation of opinion sentences by determining the prevalent word orientation. Later, in [60], Taboada et al. incorporated the analysis of intensification words (*very*, *a little*, *quite*, *somewhat*) and negation words (*not*) to modify the sentiment polarity of the affected words. In [43], Vilares et al. further incorporated the analysis of syntactic dependencies to better assess the scope of both negation and intensification, and to deal with adversative clauses (given by the adversative conjunction: *but*).

Supervised Learning-based Approaches: Also known as machine-learning-based approaches or statistical methods for sentiment classification, consist of algorithms that learn underlying patterns from example data [61], meaning data whose class or label is known for each instance, to later attempt to classify new unlabeled data [62]. Usually the steps in a machine-learning approach consist of engineering the features to represent the object whose class is to be predicted, and then using its representation as input for the algorithm. Some features frequently used in Opinion Mining are: term frequency, POS tags, sentiment words and phrases, rules of opinion, sentiment shifters and syntactic dependency, among others [4,44].

In [47] the authors were the first to implement such an approach. They compared the results of using the Naïve Bayes, Maximum Entropy classification and SVM approaches, and found that using unigrams as features (bag-of-words approach) yielded good results.

In [63], Pak and Paroubek rely on Twitter happy and sad emoticons to build a labeled training corpus. They later train three classifier algorithms: Naïve Bayes Classifier, Conditional Random Fields (CRF) and SVM, and find that the first yielded the best results. In [64], Davidov, Tsur and Rappoport in addition to emoticons also use hashtags as labels to train a clustering algorithm similar to k-Nearest Neighbors (kNN) to predict the class of unlabeled tweets.

In [65] the authors attempt to predict sentiment dynamics in the media by using 80 features extracted from tweets with two different machine-learning approaches, Dynamic Language Model (DynamicLM) [66] and a Constrained Symmetric Nonnegative Matrix Factorization (CSNMF) [67], achieving a 79% sentiment prediction accuracy with the latter, whereas only 60% with the former. This is caused mainly because DynamicLM performs better in long texts and tweets are limited to 140 characters.

Concept-based Approaches: These approaches are relatively new and consist of using ontologies for supporting the OM task. An *ontology* is defined as a model that conceptualizes the knowledge of a given domain in a way that is understood by both humans and computers. Ontologies are usually presented as graphs where concepts are mapped to nodes linked by relationships. The study presented in [68] displays a good background study on ontologies, their applications and development. It also describes how the authors incorporated them into an Opinion Mining system to extract text segments containing concepts related to the movie domain to later classify them. In [69], Cambria et al. present a semantic resource for Opinion Mining based on common-sense reasoning and domain-specific ontologies, and describe the steps they took to build it. This resource is improved in [70], where it is enriched with affective information by fusing it with WordNet-Affect [71], another semantic resource, to add emotion labels such as *Anger*, *Disgust*, *Joy* and *Surprise*. In [72], the author presents a new method to classify opinions by combining ontologies with lexical and syntactic knowledge. The work in [73] describes the steps in creating what the authors call a "Human Emotion Ontology" (HEO) which encompasses the domain of human emotions, and shows how this resource can be used to manage affective information related to data issued by online social interaction.

One of the advantages of using unsupervised methods is in not having to rely on large amounts of data for training algorithms, nevertheless it is still necessary to obtain or create a sentiment lexicon. Unsupervised methods are also less domain-dependent than supervised methods. Indeed, classifiers trained in one domain have consistently shown worse performance in other domains [74,75].

Furthermore it is worth noting that there are several other facets of Opinion Mining that are beyond the scope of this survey such as the lexicon creation problem, comparative opinions, sarcastic sentences, implicit features, cross-lingual adaptation, co-reference resolution, and topic modeling, among others. To get more information on these topics refer to the surveys [1,4].

Finally, in Table 1 we provide a brief overview on some of the most popular datasets used for training and validating Opinion Mining systems.

3. Information Fusion applied to Opinion Mining

3.1. An overview of Information Fusion

Information Fusion has many definitions, indeed some define it as the process of integrating information from multiple sources, others as the process of combining large amounts of dissimilar information into a more comprehensive and easily manageable form. Boström et al. [11] integrate these and several other definitions to create a single and universal one: "Information Fusion is the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human or automated decision making." The authors further explain that by "transformation" they mean any kind of combination and aggregation of data. They also state that the sources of data can be of many kinds such as databases, sensors, simulations, or humans, and the data type might also vary (numbers, text, graphics, ontologies).

The benefits of fusing information as opposed to using data from a single source are many. Khalegi et al. [10] compile some of the benefits of applying Information Fusion in the military context and then generalize them to be applied into other fields. The main advantages are increased data authenticity and availability. The first implies improved detection, confidence, reliability and reduction in data ambiguity, and the second means a wider spatial and temporal coverage. In Section 3 we will show specific examples issuing from the application of Information Fusion to the OM task.

Another important fact is that Information Fusion deals with two kinds of fusion, the fusion of data generated by electronic

Table 1
Datasets for Opinion Mining.

Dataset	References	Languages	Used In	Description
SemEval Twitter Dataset	[76–78]	English	NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets [79] NRC-Canada-2014: Recent Improvements in the Sentiment Analysis of Tweets [80] UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification [81]	Dataset containing regular and sarcastic tweets, SMS and LiveJournal entries, all of which are tagged with their polarity (positive, negative or neutral)
SemEval Aspect-Based OM Dataset	[82,83]	English	NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews [84] DLIREC: Aspect Term Extraction and Term Polarity Classification System [85] Sentiue: Target and Aspect based Sentiment Analysis in SemEval-2015 Task 2 [86]	Dataset composed of restaurant and laptop reviews. Each review sentence is tagged with the target of the opinion, its category and the polarity towards it (positive, negative or neutral)
Movie Review Data	[47,50,66]	English	Lexicon-Based Methods for Sentiment Analysis [60] Learning to Shift the Polarity of Words for Sentiment Classification [87]	Dataset containing movie reviews tagged at the document level as positive or negative
OpinRank Dataset	[88]	English	Good Location, Terrible Food: Detecting Feature Sentiment in User-Generated Reviews [89] CONSENTO: A New Framework for Opinion Based Entity Search and Summarization [90]	Dataset containing reviews on cars and hotels. The former are composed of the full textual review and a "favorite" field where each reviewer wrote what he deemed positive about the car. The latter are composed of unlabeled hotel reviews from various major cities, along with TripAdvisor metadata from each hotel such as its overall rating, cleanliness, service and value, among others
English Product Reviews	[6,91]	English	Movie Review Mining and Summarization [92]	Corpus composed of several product reviews tagged at the aspect level with the polarity and intensity towards it (-3 very negative; +3 very positive)
Pressrelations Dataset	[93]	German	Integrating viewpoints into newspaper Opinion Mining for a media response analysis [94]	Dataset containing German news articles tagged at the document level as positive, negative or neutral
Chinese Product Reviews	[95]	Chinese	Incorporating sentiment prior knowledge for weakly supervised Sentiment Analysis [96]	Corpora containing Chinese reviews on different products tagged at the document level as positive or negative
CLEF Replab Dataset	[97,98]	English, Spanish	LyS at CLEF RepLab 2014: Creating the State of the Art in Author Influence Ranking and Reputation Classification on Twitter [99] LIA@Replab 2014: 10 methods for 3 tasks [100]	Collection of tweets comprising several entities from the automotive, banking, universities and music domains. Each tweet is annotated with a tag showing whether it is related to the entity, a tag with its polarity (positive, negative or neutral), one depicting the topic to which it belongs and another representing the topic's priority
TASS Corpora	[101–103]	Spanish	TASS: A Naive-Bayes strategy for Sentiment Analysis on Spanish tweets [104] Elhuyar at TASS 2013 [105] LyS at TASS 2013: Analysing Spanish tweets by means of dependency parsing, semantic- oriented lexicons and psychometric word- properties [106] LyS at TASS 2014: A Prototype for Extracting and Analysing Aspects from Spanish tweets [107]	Dataset containing Spanish tweets about personalities concerning politics, economy, communication, mass media and culture. Each tweet is tagged with its polarity (very positive, positive, neutral, negative, very negative), both at the global and entity levels Additionally if a tweet does not contain sentiment it is tagged as "NONE." Furthermore, each tweet contains an agreement tag detailing whether its sentiment agrees with its content and, finally, a tag representing the topics to which the tweet belongs. Similar datasets exist exclusively for the political domain and for a discussion concerning a football championship final.

Some of these datasets are available in Kavita Ganesan's Blog,[a] Lillian Lee's homepage[b] and Bing Liu's website.[c]

[a] http://www.text-analytics101.com/2011/07/user-review-datasets_20.html (Visited May 28, 2015).
[b] http://www.cs.cornell.edu/home/llee/data/ (Visited May 28, 2015).
[c] http://www.cs.uic.edu/liub/FBS/sentiment-analysis.html#datasets (Visited May 28, 2015).

sensors, called *hard data*, and data generated by humans, called *soft data* [10]. The main differences between both reside fundamentally in the accuracy, bias, levels of observation and inferences provided by each [108]. A sensor will be better than a human in measuring the velocity of a missile or the electric current passing through a cable, while a human will be better at recognizing relationships between entities and inferring underlying reasons for observed phenomena.

Additionally, most of the research in Information Fusion has been concerned with hard data and very little with soft data [12]. However, the number of roles humans are playing in this field is growing. With the fast expansion of the Web, humans are acting as soft sensors to generate input for traditional fusion systems, and collaborating between them to perform distributed analysis and decision-making processes through multiple digitized mediums (like social media or review sites) [109]. Take a review site like Yelp for instance,[7] where users comment on various services such as restaurants, pubs and healthcare, by describing their experiences when using them. Here, each human plays the role of a soft sensor giving its impressions on a given number of aspects of the service, some of which could be quality of service, tastiness of food or overall ambience. By fusing or aggregating their opinions, it would be possible to obtain an accurate depiction of the service being evaluated and its aspects. Hence, aspect-based Opinion Mining could be considered as a form of soft, high-level information fusion.

Furthermore, Khalegi et al. [10] introduce the work done by Kokar et al. [110] as the first step towards a formalization of the

[7] http://www.yelp.com (Visited May 11, 2015).

theory of information fusion. The proposed framework captures every type of fusion, including data fusion, feature fusion, decision fusion and fusion of relational information. They also state that the most important novelty of the work is that it is able to represent both the fusion of data and the fusion of processing algorithms, and it allows for consistent measurable and provable performance. Finally, Wu and Crestani [111] present a geometric framework for Information Fusion in the context of Information Retrieval. The purpose of this framework is to represent every component in a highly dimensional space so that data fusion can be treated with geometric principles, and the Euclidean Distance can be used as a measure for effectiveness and similarity.

Now that we have explained both Opinion Mining and Information Fusion, we focus on reviewing studies that apply these fields jointly, either explicitly, meaning the authors state that they used Information Fusion techniques, or implicitly, indicating they used some form of fusion without acknowledging it. The remainder of this section is structured similarly to the typical Opinion Mining pipeline described in Sections 2.2 and 2.3. We will first review those studies in which the fusion was performed within the data sources, and later those in which it was applied during the main process, either by fusing lexical resources or techniques from different fields.

3.2. Fusion of data sources

The studies that fuse information in this step are those that use raw data from different sources, such as for example, those that combine information coming from tweets and reviews from an e-commerce site.

The work by Shroff et al. [112] presents an "Enterprise Information Fusion" framework that exploits many techniques to provide a better understanding of an enterprise's context, including client feedback and important news about events that could affect it. This framework relies on numerous sources of information for news and feedback, Twitter being the source for the former, and emails, comments on discussion boards and RSS feeds from specific blogs, sources for the latter. They also include the analysis of corporate data to understand how the events and opinions mined from external sources could impact the enterprise's business. To perform the fusion of information they use a "blackboard architecture" described in [113]. Basically, a blackboard system is a belief network in which nodes represent propositions with associated probability distributions and edges denote conditions on the nodes. The authors finally report that they observed a dip in sales of a given product after a raise in negative feedback, and state that even though their analysis was ex post, the mining of unstructured data synchronized with sales data could have provided insights to perform better marketing campaigns and find a better market niche for this product.

Dueñas-Fernández et al. [114] describe a framework for trend modeling based on LDA and Opinion Mining consisting of four steps. The first corresponds to crawling a set of manually-selected seed sources, the second to finding new sources and extracting their topics, the third and fourth to retrieving opinionated documents from social networks for each detected topic and then extracting the opinions from them. They later used a set of 20 different Rich Site Summary (RSS) feeds discussing technology topics as seed documents, and discovered 180 "feasible" feeds utilized for discovering additional information. By mining these newly found feeds, the authors extracted more than 200.000 opinionated tweets and factual documents containing 65 significant events. Finally, they were able to depict the overall polarity of these events over a period of 8 months. All things considered, the authors were able to consistently fuse information from different sources bound together by their topics, which represents a clear example of Information Fusion applied in the data extraction process of an OM application.

3.3. Fusion in the Opinion Mining core process

In this section we focus on the studies that fuse either the resources or the techniques necessary to execute the OM core process. By resources, as opposed to the data sources mentioned in Section 3.2, we mean knowledge bases that influence the OM process directly. Resources for Opinion Mining consist of lexicons, ontologies, or any annotated corpus.

3.3.1. Fusion of resources

In this section we review a few of the latest studies that apply the fusion of resources in the OM core process.

In [70] the authors fused two semantic resources to create a richer one. They enhanced the SenticNet resource [69] with affective information from WordNet-Affect (WNA) [71]. To accomplish this task, the authors assigned one of the six WNA emotion labels (*surprise*, *joy*, *sadness*, *anger*, *fear* and *disgust*) to each SenticNet concept. Further, they performed two sets of experiments, one relying only on features based on similarity measures between concepts and another considering these features with the addition of statistical features from the *International Survey of Emotion Antecedents and Reactions* (ISEAR),[8,9] containing statements associated with a particular emotion. They also experimented with three machine learning approaches, Naïve Bayes, Neural Networks and Support Vector Machines, and found the best results when using ISEAR-based features with a SVM. The final product of this work is a new resource that combines polar concepts with emotions.

Hai et al. [115] present a new method to identify opinion features from online reviews by taking advantage of the difference between a domain-specific corpus and a domain-independent one. Their methodology is first to obtain a set of candidate features based on syntactic rules, then compare these candidates with the domain-specific corpus to calculate the *intrinsic-domain relevance* (IDR) and with the domain-independent corpus to obtain the *extrinsic-domain relevance* (EDR). Those candidates with high IDR scores and low EDR scores are accepted as opinion features. Therefore, fusion occurs in the feature-extraction process of the unsupervised Opinion Mining approach, by combining information close to the domain of the review being analyzed, with more general domain-independent information. This allows for obtaining a better estimation of the degree of membership a candidate feature has with the review's domain. Finally by pruning those candidates that are not strongly related to the domain and accepting those with a high degree of relevance, the authors obtain a better set of opinion features.

The work by Xueke et al. [116] exhibits a new methodology to expand sentiment lexicons. The authors propose a generative topic model based in Latent Dirichlet Allocation (LDA) [117], to extract aspect-specific opinion words and their correspondent sentiment polarity. More specifically, their model enriches words from already existing sentiment lexicons by incorporating contextual sentence-level co-occurrences of opinion words under the assumption that usually only one sentiment is present in a sentence. They also compare the performance of their expanded lexicon on three aspect-based Opinion Mining tasks, *implicit aspect identification*, *aspect-based extractive opinion summarization* and *aspect-level sentiment classification*, and find it performs better overall than a non-expanded lexicon. To summarize, the authors found a methodology to fuse the contextual information of a given word

[8] http://www.affective-sciences.org/system/files/webpage/ISEAR_0.zip(Visited May 11, 2015).

[9] http://www.affective-sciences.org/researchmaterial (Visited May 11, 2015).

with the sentiment prior of said word, thus incorporating new information to it and producing better results.

In [118] the authors present a domain-independent opinion relevance model based on twelve features characterizing the opinion. It is worth noting that the model considers different relevancies of an opinion for different users depending on different parameters. For example, if a certain user is looking for opinions, those authored by a friend will have higher relevance than those of a stranger, since it is natural to consider a friend's opinion as more important. Additional parameters considered to assess the relevance of an opinion are the *author experience*, given by the amount of opinions the author has expressed, *age similarity*, which gives a notion of the differences in age between the opinion author and the opinion consumer, and *interest similarity*, among others. Evidently the more experience, age similarity and interest similarity an author has with a user, the more relevant the opinion will be. The novelty presented in this work is the fact of fusing information concerning the opinion's author and his network of contacts to obtain the opinion relevance metric. This would enable a generic opinion-search engine to provide better search results.

Similarly, the work presented in [119] combines the information given by the activities and relationship networks of the opinion authors to assess the opinion relevance in a social commerce context. The purpose of this analysis is to reflect the honesty, expertise and influence level of the author in the opinion domain. This work, akin to [118], presents a methodology that fuses the information concerning the author's activities and social network with the opinion information in order to estimate its relevance, veracity and objectivity, and to enhance the trust of consumers in providers within an e-commerce setting.

Schuller and Knaup [120] designed a method for Opinion Mining applied to reviews that relies on the combined knowledge of three online resources: The General Inquirer [121], WordNet [59] and ConceptNet [122]. The General Inquirer returns the sentiment valence of a given verb or adjective with 1 corresponding to a positive valence and −1 to a negative valence. If the given word is not found there, they use WordNet to look for synonyms until a match is found. Finally they rely on ConceptNet to identify features toward which the sentiments are directed. All these extracted features are then used as an input for a machine learning algorithm that will classify the review as positive, negative or neutral. Moreover, the authors test the impact of applying early fusion and late fusion methods. Early fusion corresponds simply to the aggregation of scores given by the online knowledge sources as an additional feature for the input feature vector, whereas late fusion corresponds to the combination of the output of several methods on a semantic layer. They found that early fusion yielded a slightly better accuracy and negative recall than the baseline approach at the expense of neutral recall, while late fusion for a given set of parameters, significantly increased accuracy and positive recall at a cost of a significant decrease in negative and neutral recall.

Karamatsis et al. [123] used more than 5 lexicons for creating a system that performs subjectivity detection and polarity classification in social network messages. Each lexicon provides seven features for each message, later used as inputs for a SVM classifier. They tested their system with several datasets containing data from different sources and obtained good results with LiveJournal entries, Twitter messages and sarcastic texts. Likewise, in [80] the authors used features issued from three manually constructed and two automatically generated lexicons. However, in neither work were the lexicons technically combined. The fusion took place in a higher level of abstraction, when the corresponding machine learning algorithms "learned" underlying patterns from features coming from different sources.

3.3.2. Fusion of techniques

Here we will review some of the studies that combine Opinion Mining techniques with other disciplines.

In [124], the authors jointly extract opinion targets and words by using a word-alignment model. First they find opinion targets and word candidates and later use an *Opinion Relation Graph* to assess their confidence. Finally those candidates with a confidence superior to a certain threshold are accepted as opinion targets/-words. The fusion occurs when they use information given by the word-alignment model together with that given by the opinion-relation graphs to find the opinion targets and words. Finally the authors applied their method to three different corpora and found that it outperformed state-of-the-art techniques.

Duan and Zeng [125] propose a method to forecast stock returns by mining opinions from web forums. First they extract the sentiment of a post with a purely lexical approach, meaning they use only a sentiment lexicon to obtain the polarity of sentiment-bearing words, and aggregate their scores as they appear without incorporating syntactic or semantic information. Later they use a Bayesian inference model to predict the stock returns according to the previously obtained sentiments. Here the authors fuse Opinion Mining techniques with stock prediction techniques to obtain better prediction results than those obtained by using purely numerical methods. They also propose to fuse different prediction methods, such as time series, to further improve their model.

Miao et al. [72] merged the product feature extraction and opinion extraction into one single task by using Conditional Random Fields [126]. Later, they "propagated" the found features and opinions by looking for their synonyms and antonyms, and estimated the strength of association between opinion words and product features to generate a domain-specific lexicon. This lexicon is later used to identify the polarity of opinion words in a text by following heuristic rules.

In [127], the authors present an Opinion Mining system that utilizes a supervised machine-learning approach with n-gram and lexicon features. They explicitly state "The main novelty in our system lies not in the individual techniques but rather in the way they are combined and integrated". Certainly, they not only combine four different lexicons (*MPQA* [16], *SentiWordNet* [128], *General Inquirer*,[10] and *Bing Liu's Opinion Lexicon*[11,12]) but also present new ways to combine unsupervised semantic-based techniques with supervised machine learning techniques. Specifically, they build a rule-based system which relies only on lexicon information to classify polarity, to later explore different approaches for transforming it into features for the machine-learning algorithm. They report that the combination of both approaches performs better than the systems being implemented separately, and propose to further investigate the individual contribution of each component to the overall system.

Similarly, Rosenthal et al. [129] combined two systems to obtain better results than by using each system individually. The first phrase-based sentiment-detection system relies on lexicon-based knowledge from the *Dictionary of Affect in Language* (DAL) [130], *WordNet* [131], *SentiWordNet* [128] and *Wikitionary* [132]. These and some other features are used as input for a logistic-regression classifier first presented in [133], to obtain the overall polarity of the whole input phrase. The second system uses an emoticon and acronym dictionary, as well as the DAL. The emoticon dictionary contains emoticons labeled as extremely

[10] http://www.wjh.harvard.edu/~inquirer/inqtabs.txt (Visited May 11, 2015).
[11] http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar (Visited May 11, 2015).
[12] http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon (Visited May 11, 2015).

negative, negative, neutral, positive and extremely positive, whereas the acronym dictionary presents the expansions for many internet terms such as *lol* and *fyi*. By using this information they classify the polarity of each tweet. Finally the authors found that the first system had better recall while the second presented higher precision, so they decided to combine both. To implement this they simply created the rule to use the second system when the first presented a precision lower than 70%. With this they achieved better results than when using each system individually.

In [134], Mudinas et al. showcase an Opinion Mining system that integrates both lexicon-based and learning-based techniques. Lexicon-based techniques are used for the detection of common idioms and emoticons, and for the generation of features such as *negations*, *intensifiers*, *sentiment words*, *lexicon-based sentiment scores* and for the detection of new adjectives. Later, learning-based techniques rely on a linear implementation of SVM to measure sentiment polarity. The authors state "The main advantage of our hybrid approach using a lexicon/learning symbiosis, is to attain the best of both worlds," and later specify that they successfully combined the stability and readability from a lexicon with the high accuracy and robustness from a machine-learning algorithm. Their results show that the performance of their system is higher than the state of the art.

Wu et al. [135] propose an Opinion Mining system to evaluate the usability of a given product. After the usual Opinion Mining process they use factor analysis to extract those feature-opinion pairs related to usability. Here, the fusion occurs between the usual lexicon-based OM process and some additional statistical techniques to obtain metrics related to usability.

Table 2 summarizes the papers described in this section and categorizes them according to the type of fusion they display.

Table 2
Summary of papers exemplifying different types of Information Fusion.

Type of Fusion	Study	Year
Fusion of Data Sources	Enterprise Information Fusion for Real-Time Business Intelligence [112]	2011
	A Bayesian Blackboard for Information Fusion [113]	2004
	Detecting Trends on the Web: A Multidisciplinary Approach [114]	2014
Fusion of OM Resources	Enhanced SenticNet With Affective Labels for Concept-Based Opinion Mining [70]	2013
	Identifying Features in Opinion Mining Via Intrinsic and Extrinsic Domain Relevance [115]	2014
	Aspect-Level Opinion Mining of Online Customer Reviews [116]	2013
	A Graph-Based Comprehensive Reputation Model: Exploiting the Social Context of Opinions to Enhance Trust in Social Commerce [119]	2014
	SORM: A Social Opinion Relevance Model [118]	2014
	Learning and Knowledge-Based Sentiment Analysis in Movie Review Key Excerpts [120]	2011
	AUEB: Two Stage Sentiment Analysis of Social Network Messages [123]	2014
	NRC-Canada–2014: Recent Improvements in the Sentiment Analysis of Tweets [80]	2014
Fusion of OM Techniques	Mining Fine Grained Opinions by Using Probabilistic Models and Domain Knowledge [72]	2010
	Co-Extracting Opinion Targets and Opinion Words from Online Reviews Based on the Word Alignment Model [124]	2014
	Mining Opinion and Sentiment for Stock Return Prediction Based on Web-Forum Messages [125]	2013
	Aspect-Based Polarity Classification for SemEval Task 4 [127]	2014
	Columbia NLP: Sentiment Detection of Sentences and Subjective Phrases in Social Media [129]	2014
	Combining Lexicon and Learning Based Approached for Concept-Level Sentiment Analysis [134]	2012
	A Novel Approach Based on Review Mining for Product Usability Analysis [135]	2013

3.4. A conceptual framework for applying Information Fusion to the Opinion Mining process

In this section we provide a simple framework for applying Information Fusion techniques to the Opinion Mining pipeline. The most popular fusion model is the one presented by the Joint Directors of Laboratories (JDL) [136], which has been proposed as a fusion model in other fields such as Intrusion Detection [137]. The JDL Fusion Model was originally designed for addressing the combined effects of different levels of abstraction and problem-space complexity, and was divided in 5 levels at which fusion could be performed [137,138]. Below, these levels are described and linked to the Opinion Mining pipeline depicted in Section 2:

Level 0 – Data Refinement: Just as its name suggests, this level deals with data at the lowest level of abstraction by filtering and calibrating them. In the Opinion Mining pipeline, this fusion level would be used while combining different data sources in the Data Acquisition step, as presented in Section 3.2. Furthermore, according to Dasarathy's model [139] this step is analogous to *Data In-Data Out Fusion*, meaning data is fed to this level as input and data is received as output. Dueñas-Fernández et al. [114] implicitly executed this step by filtering feeds that did not add valuable information to the process.

Level 1 – Object Refinement: In this level, data must be aligned to a common frame of reference or data structure. This step is the logical successor to level 0, indeed, after having gathered, calibrated and filtered raw data it is necessary to correlate them in order to process them jointly. In the Opinion Mining context this step corresponds to obtaining features from raw text through processes such as POS tagging and lemmatization in the data preprocessing step. This concept is consistent with the *Data In-Feature Out Fusion* presented in Dasarathy's study. For example, if we wanted to align a blog post and a review to a common representation, it would be necessary to depict both types of text according to the features they share, like sentences and the corresponding POS tags of their tokens. In general, this step will be composed of a feature extraction process which will transform data in a set of features, thus allowing to represent different documents in a common frame of reference, such as a vector space [140].

Level 2 – Situation Refinement: This level is executed at a higher level of abstraction, farther from the data and closer to the knowledge. Here, the objects represented as a set of features in a common frame of reference are evaluated according to their coordinated behavior or other high-level attribute. In Dasarathy's model this level corresponds to *Feature In-Feature Out Fusion*. In OM, this step is analogous to the Opinion Mining core process in which features are fed to an algorithm which returns other features such as the target aspects of a given opinion, along with their associated polarity.

Level 3 – Threat Assessment: Here, situation knowledge is used to analyze objects and aggregated groups against *a priori* data to provide an assessment of the current situation and suggest or identify future external conditions. In Dasarathy's model, this type of fusion is called *Feature In-Decision Out Fusion* since refined features are fed to the process and the resulting output corresponds to decisions made either by an expert system or a human at an even higher level of abstraction. For example, a manager could use a summarized opinion report to make better-informed decisions, or alternatively, an expert system

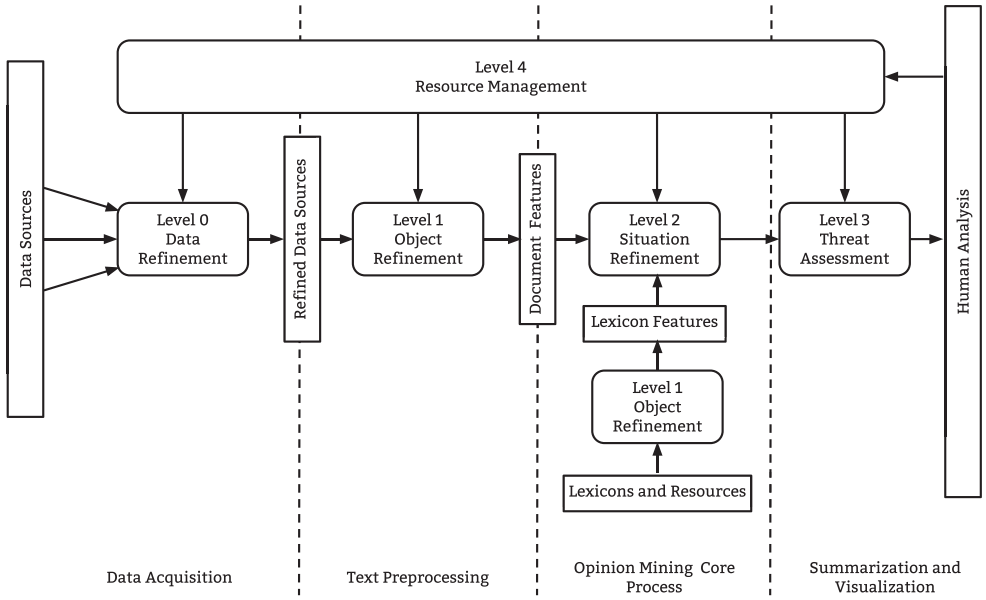


Fig. 1. Framework for applying Information Fusion to Opinion Mining.

could detect a negative trend concerning a specific product and alert those in charge of handling the situation.

Level 4 – Resource Management: In this final stage, the previous levels are further refined by using the information on the current situation and performing a more thorough analysis.

To summarize, level 0 of the JDL could be used to fuse different data sources in the data acquisition step of the Opinion Mining process. Further, level 1 of the JDL model could be used to obtain features from these different data sources and locate them in the same frame of reference in the data preprocessing step. Additionally, a different level 1 process could be used to fuse different sentiment lexicons as in the studies presented in Section 3.3.1. Likewise, the OM core process would take the features produced by level 1 and combine them in level 2 of the JDL model by producing opinion-related output. Moreover, both the summarization and visualization step of the OM process correspond to level 3 since they further aggregate the output created by level 2 in order to support decision making by processes in a higher level of abstraction (see Fig. 1).

Additionally, in order to categorize the level at which the fusion of a particular set of techniques occurs, a deeper analysis has to be performed since the category will depend on their characteristics. For example, in the work by Duan and Zeng [125] the authors fused the output generated by an OM system and the one produced by a Bayesian inference model in a level of abstraction higher than any of these two, meaning the fusion took place at level 3. Furthermore, Miao et al. [72] merged product feature extraction and opinion extraction into a single process which implies fusion took place at level 2.

Finally, it is worth mentioning that there are other, more complex Information Fusion frameworks, such as the one presented by Kokar et al. [110], that would enable researchers to represent the integration of Information Fusion techniques to Opinion Mining more formally.

4. Related work

In this final section we present surveys related to both the Opinion Mining and Information Fusion fields.

4.1. Opinion Mining

There are several surveys that cover Opinion Mining thoroughly. The work by Pang and Lee [1] considers more than 300 publications and presents diverse applications and challenges, as well as the OM problem formulation and the different approaches to solving it. The authors also mention opinion summarization, study the economic implications of reviews and comment on a plethora of publicly available resources.

A more recent review was written by Bing Liu and covers more than 400 studies [4]. Here the author covers the OM subject more exhaustively by defining an opinion model and giving a stricter definition of Sentiment Analysis. He also addresses the different levels at which OM systems are implemented (document, sentence and aspect level), deals with sentiment lexicon generation, opinion summarization, comparative and sarcastic opinions, opinion spam detection, and the quality of reviews, among others.

In [18], Cambria et al., review the Opinion Mining task in general terms, describe its evolution, and discuss the direction the field is taking. In a similar fashion, Feldman [5] describes the task and places greater emphasis on its applications and some of the common issues faced by the research community, such as sarcasm and noisy texts.

More specific OM reviews include the work by Vinodhini and Chandrasekaran [17] in which they cover subjects such as commonly employed Sentiment Analysis data sources as well as different approaches like machine learning and unsupervised learning, or as they call it, "Semantic Orientation approach". They also explain some of the challenges faced in the field such as negation handling and mention some of the applications and tools available. They finish their work by presenting a table comparing different studies, the mining techniques used in them, their feature selection approaches, data sources utilized and performance metrics (accuracy, recall and F-measure).

Khozyainov et al. [141] direct their study towards the difficulties often encountered in OM such as multidimensionality, indirect opinions, bad spelling and grammar, feature interinfluence in feature-based approaches, and the temporal dependency of opinions. Similarly, [142] studies the challenges encountered in developing Sentiment Analysis tools in the social media context, and

Table 3
Summary of Opinion Mining reviews.

Depth	Scope	Study	Year	# Refs.	Main Discussed Topics
Exhaustive	General	Opinion Mining and Sentiment Analysis [1]	2008	333	Different Approaches to OM, OM Applications, OM Challenges, Opinion Summarization, OM Resources, Economic Impact of Product Reviews
		Sentiment Analysis and Opinion Mining [4]	2012	403	OM at the Three Levels of Granularity, Opinion Summarization, Lexicon Generation, Comparative Opinions, Sarcastic Opinions, Opinion Spam Detection, Quality of Reviews
	Focused	Comprehensive Review of Opinion Summarization [143]	2011	66	Aspect-based Summarization, Non-Aspect-based Summarization, Topic Modeling, Opinion Visualization, OM Challenges
		Sentiment Analysis in Twitter [144]	2012	65	Twitter Overview, Twitter Sociological Aspects, Word-of-Mouth Importance, Latest OM Studies Applied to Twitter, Temporal Prediction of Events, Political OM, Open Research Issues
Brief	General	Techniques and Applications for Sentiment Analysis [5]	2013	40	OM at the Three Levels of Granularity, Comparative Opinions, Lexicon Generation, OM Applications, Open Research Issues
		Sentiment Analysis and Opinion Mining: A survey [17]	2012	45	Data Sources for OM, Different Approaches to OM, OM Challenges, OM Applications
		New Avenues in Opinion Mining and Sentiment Analysis [18]	2013	33	OM at the Three Levels of Granularity, Different Approaches to OM Concept-level OM, Multimodal Sentiment Analysis, Future Tendencies
		A Faceted Characterization of the Opinion Mining Landscape [20]	2014	30	OM at the Three Levels of Granularity, OM Pipeline, Lexicon Generation, Sarcastic Opinions
		Web Opinion Mining and Sentimental Analysis [145]	2013	25	Data Sources for OM, Document-level OM, Opinion Summarization, Opinion Visualization
		Opinion Mining and Analysis: A Literature Review [147]	2014	40	Document-level OM, Sentence-level OM, Learning-based approaches to OM, OM Data Sources
	Focused	A Survey of Internet Public Opinion Mining [30]	2014	47	Data Acquisition, Preprocessing, Topic Modeling, Opinion Tendency, Future Directions
		Spelling out Opinions: Difficult Cases of Sentiment Analysis [141]	2013	20	Available Tools for OM, Opinion Characteristics, OM Challenges
		Challenges in Developing Opinion Mining tools for Social Media [142]	2012	34	Specific Challenges for Applying OM in a Social Media Context (Relevance Target Identification, Negation, Context and Volatility)
		A Comparative Analysis of Opinion Mining and Sentiment Classification in Non-English Languages [146]	2013	20	Different Approaches to OM, Latest OM Studies in Hindi, Russian and Chinese

Reviews are categorized either as *Exhaustive* or *Brief*, the former meaning surveys cover their main topics in a thorough way, while the latter implies they just mention the topic and explain it briefly. Furthermore, *General* reviews are those that present Opinion Mining as a whole whereas *Focused* reviews focus on a particular Opinion Mining sub-topic. Finally, *# Refs.* represent the amount of studies cited by each survey (references).

covers additional concepts such as relevance, contextual information and volatility over time.

In [143] the authors survey the state of the art in opinion summarization in which they describe the background of Opinion Mining, define a conceptual framework for opinion summarization, and deepen their analysis in aspect-based and non-aspect-based opinion summarization. Finally they discuss how to evaluate summarization methods and mention some of the open challenges in this field.

Martínez-Cámara et al. [144] focus on the latest advancements in Sentiment Analysis as applied to Twitter data. They begin by giving an overview of this microblogging site mentioning some of its sociological aspects as well as the importance of the word of mouth, and later discuss the research concerning polarity classification, temporal prediction of events and political Opinion Mining. In a similar fashion, Marrese-Taylor et al. [145] present an overview of Opinion Mining, describe some of the most popular sources for extracting opinionated data, discuss summarization and visualization techniques, and finally exhibit an example of a document-level Opinion Mining application for finding the most influential users on Twitter.

Medagoda et al. [146] focus on recent advancements in Opinion Mining achieved in Hindi, Russian and Chinese. Guo et al. [30] define the concept of "Public Opinion Mining," compare different approaches used in each step of the OM pipeline and propose future directions for the field. In [20] the authors propose a faceted characterization of Opinion Mining composed of two main branches, namely *opinion structure* which deals with the relation between unstructured subjective text and structured conceptual elements, and *Opinion Mining tools and techniques* which are the means to achieve the OM task. They also tackle the problems of entity discovery and aspect identification, lexicon acquisition and sarcasm detection. Finally [147] covers some of the usual OM tasks and presents a table similar to the one presented in [17] but instead of using known metrics it just shows an arbitrary "performance" metric without clarifying whether if it represents accuracy, precision, recall, F-measure or some other measure.

Table 3 presents a summary of Opinion Mining reviews presented in this section.

4.2. Information Fusion

One of the most recent surveys on Information Fusion corresponds to the work by Khalegi et al. [10]. In it, the authors focus on reviewing the state of the art in multisensor data fusion. They begin by explaining the potential benefits of implementing an information fusion system and the usual challenges faced while doing so. They also present the work done by Kokar et al. [110] and describe it as one of the first attempts to formally define the Information Fusion theory. They later review the techniques for the fusion of *hard data* (generated by sensors), namely by describing the algorithms used for data fusion in detail, and classifying them according to the challenges they tackle. Finally, the authors mention some of the efforts made towards the fusion of *soft data* (generated by humans) and the new tendency of attempting to fuse them with hard data.

General surveys include the work by Bloch [148], in which she compares and classifies the different operators used to combine the data gathered by multiple sensors in information fusion systems. She classifies these operators as "Context Independent Constant Behavior Operators (CICB)", "Context Independent Variable Behavior Operators (CIVB)" and "Context Dependent

Table 4
Summary of Information Fusion Reviews.

Depth	Scope	Study	Year	# Refs.	Main Discussed Topics
Exhaustive	General	Multisensor Data Fusion [10]	2013	197	Multisensor Data Fusion, Hard-Data Fusion, Soft-Data Fusion, IF Challenges, IF Algorithms, Emerging Paradigms, Open Research Issues
		Formalizing Classes of Information Fusion Systems [110]	2004	36	Fusion Definition, IF Theory, Multi-Source IF, Single-Source IF, Effectiveness of IF Systems
		Information Combination Operators for Data Fusion: A Comparative Review with Classification [148]	1996	20	Multi-Source IF, Behavior of IF Operators, Image Fusion
		An Introduction to Multisensor Data Fusion [149]	1997	130	IF Terminology, Military IF Applications, Non-Military IF Applications, JDL Model, IF Process Model, IF Architectures
		Approaches to Multisensor Data Fusion in Target Tracking: A survey [150]	2006	195	JDL Model Stages (Object Refinement, Situation Assessment, Threat Assessment, Process Assessment), Multisensor-Tracking Challenges
	Focused	Ontology-based Integration of Information: A Survey of Existing Approaches [151]	2001	60	Role of Ontologies, Representation of Ontologies, Creation of Ontologies, Use of Mappings to Integrate Ontologies and IF Systems
	Applications	Information Fusion for Computer Security: State of the Art and Open Issues [155]	2009	59	Computer Security, Intrusion Detection, JDL Model, Current IF Applications in Computer Security, Proposal of new IF approach applied to Computer Security, Open Research Issues
		Data Fusion in Intelligent Transportation Systems: Progress and Challenges– A Survey [156]	2011	94	IF Approaches, Opportunities of Applying IF to ITS, Challenges of Applying IF to ITS, Current IF Applications in ITS, Future Directions
		Information Fusion in Data Privacy: A Survey [157]	2012	139	Data Privacy Basic Concepts, Data Protection Approaches, IF Applied to Data Privacy, Record Linkage
		A Survey of Multi-Source Domain Adaptation [159]	2014	43	Domain Adaptation Basic Concepts, Domain Adaptation Algorithms, IF Applied to Multi-Source Domain Adaptation, Datasets for Domain Adaptation, Open Research Issues
Brief	General	Data Fusion Lexicon [136]	1991	N/A	Data Fusion Terms, JDL Model Proposition
	Focused	Reliability in Information Fusion: Literature Survey [152]	2004	40	*Reliability* Definition, Incorporating Reliability into IF Operators, Reliability Coefficients, Reliability of Fusion Results
		Web Information Fusion: A Review of the State of the Art [153]	2008	33	The Web, IF Overview, Ontologies, Semantic Web, Relationship Between IF and the Web, Web-Based Support Systems
	Applications	Image Fusion: Advances in the State of the Art [154]	2007	66	Image Fusion Basic Concepts, Image-Fusion-Algorithms Classification, Image Registration, Image Fusion Applications, Emerging Image Fusion Technologies, Future Directions

The categories for *Depth* and *Scope* are equal to those presented in Table 3, with the addition of *Applications*, which represents those surveys that review the latest advancements of Information Fusion applied to a specific field.

Operators (CD)," and describe the theory underlying each one of them. Furthermore, Hall et al. [149] review both the military and non-military applications for Information Fusion, describe a data fusion process model and some of the architectures for data fusion (Centralized, Autonomous and Hybrid Fusion). Additionally, Smith et al. [150] comment on several methods for target tracking through sensor data fusion. The authors structure their work according to the Joint Directors of Laboratories (JDL) model [136] by reviewing the advancements for each one of its levels: object refinement, situation assessment, threat assessment and process assessment.

More specific studies include the survey by Wache et al. [151] in which the authors review the use of ontologies for the fusion of data issued from different sources. Specifically, they define the role of ontologies, their representations, the use of mappings designed to integrate them into the fusion systems and their engineering process. In [152] the authors introduce the concept of *reliability* and discuss the theory and approaches for incorporating it into common IF operators. They define reliability coefficients as the measure of how well each belief model represents reality. Yao et al. [153] define "Web Information Fusion" as the task of combining all kinds of information on the Web. They give an overview of the advances in this field by reviewing some of the contributions made to it by the Artificial Intelligence (AI) and database communities to it. Furthermore, they comment on the role that ontologies and the "Semantic Web" play in Web Information Fusion.

Additionally, there are other surveys reviewing the application of Information Fusion in specific fields. The work in [154] presents the state of the art in image fusion. The authors begin by describing this field, then review its history, categorize the most common image fusion algorithms into low, mid and high level, describe some of the applications, and finish by mentioning some emerging technologies and future directions for the field. Corona et al. [155] review the state of the art of Information Fusion applied to computer security. They first define computer security as the quantitative evaluation of three qualities of an information flow: availability, confidentiality and integrity. They then describe the intrusion-detection problem, state that it corresponds to a pattern recognition task and define the role Information Fusion plays in it. Later, the authors present a high-level framework for information fusion, comment on the current applications, and finish by proposing a new approach for data fusion in computer security. Faouzi et al. [156] provide a survey of the application of Information Fusion in different areas of Intelligent Transport Systems (ITS). First, they describe the background on data fusion, secondly, they enumerate the opportunities and challenges of ITS Information Fusion, and finally review the applications in which IF is applied to ITS. In [157] the authors review the role of IF in data privacy [158]. They begin by defining data privacy, next they comment on several protection methods used in the literature, such as *microaggregation* which provides privacy by clustering data and representing it as the clusters' centroids, and *record linkage* which in the context of data privacy represents a way to provide disclosure risk assessment of protected data. The authors also demonstrate how both of these methods are greatly benefited from the use of Information Fusion. Finally, Sun et al. [159] exhibit a survey on multi-source domain adaptation, in which they comment on the latest advancements concerning the problem of adapting training data to test data from a different domain. Their work includes the review of algorithms, theoretical results and the discussion on open problems and future work.

The Information Fusion reviews described in this section are summarized in Table 4.

5. Conclusions

In this paper we presented a short survey of the most popular Opinion Mining techniques, defined the Information Fusion field, proposed a simple framework for guiding the fusion process in an Opinion Mining system and reviewed some of the studies that have successfully implemented Information Fusion techniques in the Opinion Mining context. Indeed, the future of Opinion Mining relies on creating better and deeper sources of knowledge, which can be achieved by fusing already existing knowledge bases such as ontologies and lexicons. Nevertheless, few studies have done so by explicitly applying well-established techniques. In fact, studies in which authors fuse different lexical resources or techniques without following any standard procedure are the most common.

However, even if a fusion process does not follow a strict framework, the results of applying it are consistently better than not doing so. From this it follows that both fields could greatly benefit from a more standardized and consistent way to fuse opinion-related data. This is why the knowledge generated in the Information Fusion field becomes essential. Broadening the knowledge on soft fusion for instance, would facilitate the fusion of data from different online sources such as Twitter and review sites, increasing its authenticity and availability, which would in turn allow the production of higher-quality Opinion Mining systems. Furthermore, advancements in the fusion of soft data with hard data would make possible the combination of audiovisual content with textual data and push forward the Multimodal Sentiment Analysis field [18].

Admittedly, using Information Fusion jointly with Opinion Mining would allow for a better understanding of the effects of every fused component in the final system while enabling researchers to improve the fusion process and ultimately lay the foundations for creating better systems.

Acknowledgements

The authors would like to acknowledge the continuous support of the Chilean Millennium Institute of Complex Engineering Systems (ICM: P-05-004-F, CONICYT: FBO16). This work was partially funded by *Corporación de Fomento de la Producción* (CORFO) under project number 13IDL2-223170 entitled *OpinionZoom* (www.opinionzoom.cl).

References

[1] B. Pang, L. Lee, Opinion mining and sentiment analysis, Found. Trends Inf. Retrieval 2 (1–2) (2008) 1–135, http://dx.doi.org/10.1561/1500000011.
[2] P.A. Tapia, J.D. Velásquez, Twitter sentiment polarity analysis: a novel approach for improving the automated labeling in a text corpora, in: Active Media Technology, Springer International Publishing, 2014, pp. 274–285.
[3] G.L. Rodrigo Dueñas-Fernández, J.D. Velásquez, Sentiment polarity of trends on the web using opinion mining and topic modeling, in: Proceedings of the First Workshop On Social Web Intelligence (WOSWI) in Conjunction with 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology, WI-IAT Í3, IEEE Computer Society, Washington, DC, USA, 2013.
[4] B. Liu, Sentiment analysis and opinion mining, Synth. Lect. Hum. Lang. Technol. 5 (1) (2012) 1–167, http://dx.doi.org/10.2200/S00416ED1V01Y201204HLT016.
[5] R. Feldman, Techniques and applications for sentiment analysis, Commun. ACM 56 (4) (2013) 82–89, http://dx.doi.org/10.1145/2436256.2436274.
[6] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2004, pp. 168–177, http://dx.doi.org/10.1145/1014052.1014073.
[7] A. Tumasjan, T.O. Sprenger, P.G. Sandner, I.M. Welpe, Predicting elections with Twitter: what 140 characters reveal about political sentiment, in: Proceedings of the 4th International Conference on Weblogs and Social Media (ICWSM 2010), vol. 10, AAAI Press, Washington, USA, 2010, pp. 178–185. <https://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/view/1441>.
[8] W. Zhang, S. Skiena, Trading strategies to exploit blog and news sentiment, in: Proceedings of the 4th International Conference on Weblogs and Social Media (ICWSM 2012), vol. 10, AAAI Press, Washington, USA, 2010, pp. 375–378. http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/view/1529.
[9] J.D. Velásquez, P. González, Expanding the possibilities of deliberation: the use of data mining for strengthening democracy with an application to education reform, Inf. Soc. 26 (1) (2010) 1–16, http://dx.doi.org/10.1080/01972240903423329.
[10] B. Khaleghi, A. Khamis, F.O. Karray, S.N. Razavi, Multisensor data fusion: a review of the state-of-the-art, Inf. Fusion 14 (1) (2013) 28–44, http://dx.doi.org/10.1016/j.inffus.2011.08.001.
[11] H. Boström, S.F. Andler, M. Brohede, R. Johansson, A. Karlsson, J. Van Laere, L. Niklasson, M. Nilsson, A. Persson, T. Ziemke, On the definition of information fusion as a field of research, Tech. Rep. HS-IKI-TR-07-006, Informatics Research Centre, University of Skövde, 2007. <http://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-1256>.
[12] K. Sambhoos, J. Llinas, E. Little, Graphical methods for real-time fusion and estimation with soft message data, in: Proceedings of the 11th International Conference on Information Fusion (FUSION 2008), IEEE, Cologne, Germany, 2008, pp. 1–8. <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4632405>.
[13] G.L. Urban, J.R. Hauser, "Listening in" to find and explore new combinations of customer needs, J. Market. 68 (2) (2004) 72–87, http://dx.doi.org/10.1509/jmkg.68.2.72.27793.
[14] O. Netzer, R. Feldman, J. Goldenberg, M. Fresko, Mine your own business: market-structure surveillance through text mining, Market. Sci. 31 (3) (2012) 521–543, http://dx.doi.org/10.1287/mksc.1120.0713.
[15] B. Liu, Sentiment analysis and subjectivity, in: R. Dale, H. Moisl, H. Somers (Eds.), Handbook of Natural Language Processing, second ed., Machine Learning and Pattern Recognition, CRC Press, New York, NY, USA, 2010, pp. 627–666.
[16] T. Wilson, J. Wiebe, P. Hoffmann, Recognizing contextual polarity in phrase-level sentiment analysis, in: Proceedings of the Joint Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP 2005), Association for Computational Linguistics, Vancouver, Canada, 2005, pp. 347–354, http://dx.doi.org/10.3115/1220575.1220619.
[17] G. Vinodhini, R. Chandrasekaran, Sentiment analysis and opinion mining: a survey, Int. J. Adv. Res. Comput. Sci. Softw. Eng. 2 (6) (2012) 282–292.
[18] E. Cambria, B. Schuller, Y. Xia, C. Havasi, New avenues in opinion mining and sentiment analysis, IEEE Intell. Syst. 28 (2) (2013) 15–21, http://dx.doi.org/10.1109/MIS.2013.30.
[19] A. Neviarouskaya, H. Prendinger, M. Ishizuka, Recognition of fine-grained emotions from text: an approach based on the compositionality principle, in: T. Nishida, L.C. Jain, C. Faucher (Eds.), Modeling Machine Emotions for Realizing Intelligence, Smart Innovation, Systems and Technologies, vol. 1, Springer, Berlin, Heidelberg, 2010, pp. 179–207, http://dx.doi.org/10.1007/978-3-642-12604-8_9.
[20] R. Arora, S. Srinivasa, A faceted characterization of the opinion mining landscape, in: Proceedings of the 6th International Conference on Communication Systems and Networks (COMSNETS 2014), IEEE, Bangalore, India, 2014, pp. 1–6, http://dx.doi.org/10.1109/COMSNETS.2014.6734936.
[21] L. Dey, S.M. Haque, Opinion mining from noisy text data, Int. J. Doc. Anal. Recogn. (IJDAR) 12 (3) (2009) 205–226, http://dx.doi.org/10.1007/s10032-009-0090-z.
[22] F.H. Khan, S. Bashir, U. Qamar, TOM: Twitter opinion mining framework using hybrid classification scheme, Decis. Support Syst. 57 (2014) 245–257, http://dx.doi.org/10.1016/j.dss.2013.09.004.
[23] A. Bakliwal, J. Foster, J. van der Puil, R. O'Brien, L. Tounsi, M. Hughes, Sentiment analysis of political tweets: towards an accurate classifier, in: Proceedings of the NAACL Workshop on Language Analysis in Social Media (LASM 2013), Association for Computational Linguistics, Atlanta, GA, USA, 2013, pp. 49–58. <http://doras.dcu.ie/19962/>.
[24] L. Jiang, M. Yu, M. Zhou, X. Liu, T. Zhao, Target-dependent Twitter sentiment classification, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011), vol. 1, Association for Computational Linguistics, Portland, OR, USA, 2011, pp. 151–160. <http://dl.acm.org/citation.cfm?id=2002472.2002492.
[25] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau, Sentiment analysis of Twitter data, in: Proceedings of the Workshop on Languages in Social Media (LSM 2011), Association for Computational Linguistics, Portland, OR, USA, 2011, pp. 30–38. <http://dl.acm.org/citation.cfm?id=2021109.2021114>.
[26] B. Krishnamurthy, P. Gill, M. Arlitt, A few chirps about Twitter, in: Proceedings of the 1st Workshop on Online Social Networks (WOSN 2008), ACM, Seattle, WA, USA, 2008, pp. 19–24, http://dx.doi.org/10.1145/1397735.1397741.
[27] Y.H. Gu, S.J. Yoo, Rules for mining comparative online opinions, in: Proceedings of the 4th International Conference on Computer Sciences and Convergence Information Technology (ICCIT 2009), IEEE, Seoul, Korea, 2009, pp. 1294–1299, http://dx.doi.org/10.1109/ICCIT.2009.16.
[28] X. Hu, J. Tang, H. Gao, H. Liu, Unsupervised sentiment analysis with emotional signals, in: Proceedings of the 22nd International Conference on World Wide Web (WWW 2013), International World Wide Web Conferences Steering Committee, Rio de Janeiro, Brazil, 2013, pp. 607–618. <http://dl.acm.org/citation.cfm?id=2488388.2488442>.
[29] C. Olston, M. Najork, Web crawling, Found. Trends Inf. Retrieval 4 (3) (2010) 175–246. <http://dl.acm.org/citation.cfm?id=1734789>.
[30] K. Guo, L. Shi, W. Ye, X. Li, A survey of internet public opinion mining, in: Proceedings of the International Conference on Progress in Informatics and

Computing (PIC 2014), IEEE, Shanghai, China, 2014, pp. 173–179, http://dx.doi.org/10.1109/PIC.2014.6972319.
[31] T. Fu, A. Abbasi, D. Zeng, H. Chen, Sentimental spidering: leveraging opinion information in focused crawlers, ACM Trans. Inf. Syst. 30 (4) (2012) 24:1–24:30, http://dx.doi.org/10.1145/2382438.2382443.
[32] A.G. Vural, Sentiment-focused Web crawling, Ph.D. thesis, Middle East Technical University, 2013. <http://etd.lib.metu.edu.tr/upload/12616409/index.pdf>.
[33] A. Hippisley, Lexical analysis, in: R. Dale, H. Moisl, H. Somers (Eds.), Handbook of Natural Language Processing, second ed., Machine Learning and Pattern Recognition, CRC Press, New York, NY, USA, 2010, pp. 31–58.
[34] D. Palmer, Text preprocessing, in: R. Dale, H. Moisl, H. Somers (Eds.), Handbook of Natural Language Processing, 2nd Edition., Machine Learning and Pattern Recognition, CRC Press, New York, NY, USA, 2010, pp. 9–30.
[35] B. Liu, Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, first ed., Springer, Berlin, Heidelberg, New York, NY, USA, 2007. Ch. 6.
[36] M.F. Porter, An algorithm for suffix stripping, Program 40 (3) (2006) 211–218, http://dx.doi.org/10.1108/00330330610681286.
[37] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, vol. 1, Cambridge University Press, 2008. Ch. 2.
[38] T. Kiss, J. Strunk, Unsupervised multilingual sentence boundary detection, Comput. Linguist. 32 (4) (2006) 485–525, http://dx.doi.org/10.1162/coli.2006.32.4.485.
[39] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python, first ed., O'Reilly Media, Inc., 2009.
[40] E. Brill, A simple rule-based part of speech tagger, in: Proceedings of the Workshop on Speech and Natural Language, Association for Computational Linguistics, Harriman, New York, 1992, pp. 112–116, http://dx.doi.org/10.3115/1075527.1075553.
[41] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, N.A. Smith, Part-of-speech tagging for Twitter: annotation, features, and experiments, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011), vol. 2, Association for Computational Linguistics, Portland, Oregon, 2011, pp. 42–47. <http://dl.acm.org/citation.cfm?id=2002736.2002747>.
[42] T. Güngör, Part-of-speech tagging, in: R. Dale, H. Moisl, H. Somers (Eds.), Handbook of Natural Language Processing, 2nd Edition., Machine Learning and Pattern Recognition, CRC Press, New York, NY, USA, 2010, pp. 9–30.
[43] D. Vilares, M.A. Alonso, C. Gómez-Rodríguez, A syntactic approach for opinion mining on Spanish reviews, Nat. Lang. Eng. 21 (01) (2015) 139–163, http://dx.doi.org/10.1017/S1351324913000181.
[44] M. Joshi, C. Penstein-Rosé, Generalizing dependency features for opinion mining, in: Proceedings of the 4th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2009), Association for Computational Linguistics, Suntec, Singapore, 2009, pp. 313–316. <http://dl.acm.org/citation.cfm?id=1667583.1667680>.
[45] J.D. Velásquez, V. Palade, Adaptive Web Sites, vol. 170, IOS Press, 2008.
[46] V. Hangya, R. Farkas, Target-oriented opinion mining from tweets, in: Proceedings of the 4th International Conference on Cognitive Infocommunications (CogInfoCom 2013), IEEE, Budapest, Hungary, 2013, pp. 251–254, http://dx.doi.org/10.1109/CogInfoCom.2013.6719251.
[47] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up?: sentiment classification using machine learning techniques, Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), vol. 10, Association for Computational Linguistics, Philadelphia, PA, USA, 2002, pp. 79–86, http://dx.doi.org/10.3115/1118693.1118704.
[48] D. Vilares, M.Á. Alonso, C. Gómez-Rodríguez, Supervised polarity classification of Spanish tweets based on linguistic knowledge, in: Proceedings of the 13th Symposium on Document Engineering (DocEng 2013), ACM, Florence, Italy, 2013, pp. 169–172, http://dx.doi.org/10.1145/2494266.2494300.
[49] A. Bakliwal, P. Arora, S. Madhappan, N. Kapre, M. Singh, V. Varma, Mining sentiments from tweets, in: Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2012), Association for Computational Linguistics, Jeju, Korea, 2012, pp. 11–18. <http://dl.acm.org/citation.cfm?id=2392963.2392970>.
[50] B. Pang, L. Lee, Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales, in: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005), Association for Computational Linguistics, Ann Arbor, MI, USA, 2005, pp. 115–124, http://dx.doi.org/10.3115/1219840.1219855.
[51] E. Riloff, J. Wiebe, Learning extraction patterns for subjective expressions, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2003), Association for Computational Linguistics, Sapporo, Japan, 2003, pp. 105–112, http://dx.doi.org/10.3115/1119355.1119369.
[52] H. Yu, V. Hatzivassiloglou, Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2003), Association for Computational Linguistics, Sapporo, Japan, 2003, pp. 129–136, http://dx.doi.org/10.3115/1119355.1119372.
[53] A.-M. Popescu, O. Etzioni, Extracting product features and opinions from reviews, in: A. Kao, S.R. Poteet (Eds.), Natural Language Processing and Text Mining, Springer, London, 2007, pp. 9–28, http://dx.doi.org/10.1007/978-1-84628-754-1_2.
[54] E. Marrese-Taylor, J.D. Velásquez, F. Bravo-Marquez, Y. Matsuo, Identifying customer preferences about tourism products using an aspect-based opinion mining approach, Proc. Comput. Sci. 22 (2013) 182–191, http://dx.doi.org/10.1016/j.procs.2013.09.094.
[55] E. Marrese-Taylor, J.D. Velásquez, F. Bravo-Marquez, Opinion zoom: a modular tool to explore tourism opinions on the Web, in: Proceedings of the the IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT 2013), IEEE, Atlanta, GA, USA, 2013, pp. 261–264, http://dx.doi.org/10.1109/WI-IAT.2013.193.
[56] Y. Wu, Q. Zhang, X. Huang, L. Wu, Phrase dependency parsing for opinion mining, Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009), vol. 3, Association for Computational Linguistics, Singapore, 2009, pp. 1533–1541. <http://dl.acm.org/citation.cfm?id=1699648.1699700>.
[57] T. Nakagawa, K. Inui, S. Kurohashi, Dependency tree-based sentiment classification using CRFs with hidden variables, in: Proceedings of The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2010), Association for Computational Linguistics, Los Angeles, CA, USA, 2010, pp. 786–794. <http://dl.acm.org/citation.cfm?id=1857999.1858119>.
[58] P.D. Turney, Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2002), Association for Computational Linguistics, Philadelphia, PA, USA, 2002, pp. 417–424, http://dx.doi.org/10.3115/1073083.1073153.
[59] G.A. Miller, WordNet: a lexical database for English, Commun. ACM 38 (11) (1995) 39–41, http://dx.doi.org/10.1145/219717.219748.
[60] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, M. Stede, Lexicon-based methods for sentiment analysis, Comput. Linguist. 37 (2) (2011) 267–307, http://dx.doi.org/10.1162/COLI_a_00049.
[61] V.L. Rebolledo, G. L'Huillier, J.D. Velásquez, Web pattern extraction and storage, in: Advanced Techniques in Web Intelligence-I, Springer, Berlin, Heidelberg, 2010, pp. 49–77.
[62] T. Mitchell, Machine Learning, McGraw-Hill, 1997.
[63] A. Pak, P. Paroubek, Twitter as a corpus for sentiment analysis and opinion mining, Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010), vol. 10, European Language Resources Association, Valletta, Malta, 2010, pp. 1320–1326. <http://www.lrec-conf.org/proceedings/lrec2010/pdf/385_Paper.pdf>.
[64] D. Davidov, O. Tsur, A. Rappoport, Enhanced sentiment learning using Twitter hashtags and smileys, in: Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), Association for Computational Linguistics, Beijing, China, 2010, pp. 241–249. <http://dl.acm.org/citation.cfm?id=1944566.1944594>.
[65] L.T. Nguyen, P. Wu, W. Chan, W. Peng, Y. Zhang, Predicting collective sentiment dynamics from time-series social media, Proceedings of the 1st International Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM 2012), vol. 6, ACM, Beijing, China, 2012, pp. 6:1–6:8, http://dx.doi.org/10.1145/2346676.2346682.
[66] B. Pang, L. Lee, A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, in: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004), Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 271–278, http://dx.doi.org/10.3115/1218955.1218990.
[67] W. Peng, D.H. Park, Generate adjective sentiment dictionary for social media sentiment analysis using constrained nonnegative matrix factorization, in: Proceedings of the 5th International Conference on Weblogs and Social Media (ICWSM 2011), AAAI Press, Barcelona, Spain, 2011, pp. 273–280. <http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2723>.
[68] L. Zhou, P. Chaovalit, Ontology-supported polarity mining, J. Am. Soc. Inf. Sci. Technol. 59 (1) (2008) 98–110, http://dx.doi.org/10.1002/asi.20735.
[69] E. Cambria, R. Speer, C. Havasi, A. Hussain, SenticNet: a publicly available semantic resource for opinion mining, in: Proceedings of the Fall Symposium on Computational Models of Narrative, AAAI, Arlington, VA, USA, 2010, pp. 14–18. <https://www.aaai.org/ocs/index.php/FSS/FSS10/paper/view/2216>.
[70] S. Poria, A. Gelbukh, A. Hussain, N. Howard, D. Das, S. Bandyopadhyay, Enhanced SenticNet with affective labels for concept-based opinion mining, IEEE Intell. Syst. 28 (2) (2013) 31–38, http://dx.doi.org/10.1109/MIS.2013.4.
[71] C. Strapparava, A. Valitutti, et al., WordNet affect: an affective extension of WordNet, Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004), vol. 4, European Language Resources Association, Lisbon, Portugal, 2004, pp. 1083–1086. <http://www.lrec-conf.org/proceedings/lrec2004/pdf/369.pdf>.
[72] Q. Miao, Q. Li, D. Zeng, Mining fine grained opinions by using probabilistic models and domain knowledge, Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technologies (WI-IAT 2010), vol. 1, IEEE, Toronto, Canada, 2010, pp. 358–365, http://dx.doi.org/10.1109/WI-IAT.2010.193.
[73] M. Grassi, E. Cambria, A. Hussain, F. Piazza, Sentic Web: a new paradigm for managing social media affective information, Cogn. Comput. 3 (3) (2011) 480–489, http://dx.doi.org/10.1007/s12559-011-9101-8.
[74] A. Aue, M. Gamon, Customizing sentiment classifiers to new domains: a case study, in: Proceedings of the International Conference on Recent Advances in

Natural Language Processing (RANLP 2005), Borovets, Bulgaria, 2005. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.90.3612>.
[75] J. Blitzer, M. Dredze, F. Pereira, Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification, in: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007), Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 440–447. <http://www.aclweb.org/anthology/P07-1056>.
[76] P. Nakov, S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter, T. Wilson, SemEval-2013 Task 2: sentiment analysis in Twitter, in: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Atlanta, GA, USA, 2013, pp. 312–320. <http://www.aclweb.org/anthology/S13-2052>.
[77] S. Rosenthal, A. Ritter, P. Nakov, V. Stoyanov, SemEval-2014 Task 9: sentiment analysis in Twitter, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Association for Computational Linguistics and Dublin City University, Dublin, Ireland, 2014, pp. 73–80. <https://www.aclweb.org/anthology/S/S14/S14-2009.pdf>.
[78] S. Rosenthal, P. Nakov, S. Kiritchenko, S. Mohammad, A. Ritter, V. Stoyanov, SemEval-2015 Task 10: sentiment analysis in Twitter, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado, 2015, pp. 451–463. <http://alt.qcri.org/semeval2015/cdrom/pdf/SemEval078.pdf>.
[79] S. Mohammad, S. Kiritchenko, X. Zhu, NRC-Canada: building the state-of-the-art in sentiment analysis of tweets, in: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Atlanta, GA, USA, 2013, pp. 321–327. <http://www.aclweb.org/anthology/S13-2053>.
[80] X. Zhu, S. Kiritchenko, S.M. Mohammad, NRC-Canada-2014: recent improvements in the sentiment analysis of tweets, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Association for Computational Linguistics and Dublin City University, Dublin, Ireland, 2014, pp. 443–447. <http://www.aclweb.org/anthology/S14-2077>.
[81] A. Severyn, A. Moschitti, UNITN: training deep convolutional neural network for Twitter sentiment classification, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado, 2015, pp. 464–469. <http://alt.qcri.org/semeval2015/cdrom/pdf/SemEval079.pdf>.
[82] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, Semeval-2014 Task 4: aspect based sentiment analysis, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Association for Computational Linguistics and Dublin City University, Dublin, Ireland, 2014, pp. 27–35. <http://www.aclweb.org/anthology/S14-2004>.
[83] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, I. Androutsopoulos, SemEval-2015 Task 12: aspect based sentiment analysis, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado, 2015, pp. 486–495. <http://alt.qcri.org/semeval2015/cdrom/pdf/SemEval082.pdf>.
[84] S. Kiritchenko, X. Zhu, C. Cherry, S. Mohammad, NRC-Canada-2014: detecting aspects and sentiment in customer reviews, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Association for Computational Linguistics and Dublin City University, Dublin, Ireland, 2014, pp. 437–442. <http://www.aclweb.org/anthology/S14-2076>.
[85] Z. Toh, W. Wang, DLIREC: aspect term extraction and term polarity classification system, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Association for Computational Linguistics and Dublin City University, Dublin, Ireland, 2014, pp. 235–240. <http://www.aclweb.org/anthology/S14-2038>.
[86] J. Saias, Sentiue: target and aspect based sentiment analysis in SemEval-2015 Task 12, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado, 2015, pp. 767–771. <http://alt.qcri.org/semeval2015/cdrom/pdf/SemEval130.pdf>.
[87] D. Ikeda, H. Takamura, L.-A. Ratinov, M. Okumura, Learning to shift the polarity of words for sentiment classification, in: Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP 2008), vol. 1, Association for Computational Linguistics, Hyderabad, India, 2008, pp. 296–303. <https://www.aclweb.org/anthology/I/I08/I08-1039.pdf>.
[88] K. Ganesan, C. Zhai, Opinion-based entity ranking, Inf. Retrieval 15 (2) (2011) 116–150, http://dx.doi.org/10.1007/s10791-011-9174-8.
[89] M. Cataldi, A. Ballatore, I. Tiddi, M.-A. Aufaure, Good location, terrible food: detecting feature sentiment in user-generated reviews, Soc. Netw. Anal. Min. 3 (4) (2013) 1149–1163, http://dx.doi.org/10.1007/s13278-013-0119-7.
[90] J. Choi, D. Kim, S. Kim, J. Lee, S. Lim, S. Lee, J. Kang, Consento: a new framework for opinion based entity search and summarization, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM 2012), ACM, Maui, Hawaii, USA, 2012, pp. 1935–1939, http://dx.doi.org/10.1145/2396761.2398547.
[91] X. Ding, B. Liu, P.S. Yu, A holistic lexicon-based approach to opinion mining, in: Proceedings of the 1st International Conference on Web Search and Data Mining (WSDM 2008), ACM, Stanford, CA, USA, 2008, pp. 231–240, http://dx.doi.org/10.1145/1341531.1341561.
[92] L. Zhuang, F. Jing, X.-Y. Zhu, Movie review mining and summarization, in: Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM 2006), ACM, Arlington, Virginia, USA, 2006, pp. 43–50, http://dx.doi.org/10.1145/1183614.1183625.
[93] T. Scholz, S. Conrad, L. Hillekamps, Opinion mining on a german corpus of a media response analysis, in: P. Sojka, A. Horák, I. Kopeček, K. Pala (Eds.), Text, Speech and Dialogue, Lect. Notes Comput. Sci., vol. 7499, Springer, Berlin, Heidelberg, 2012, pp. 39–46, http://dx.doi.org/10.1007/978-3-642-32790-2_4.
[94] T. Scholz, S. Conrad, Integrating viewpoints into newspaper opinion mining for a media response analysis, in: Proceedings of the 11th Conference on Natural Language Processing (KONVENS 2012), Vienna, Austria, 2012, pp. 30–38. <http://www.oegai.at/konvens2012/proceedings/08_scholz12o/08_scholz12o.pdf>.
[95] T. Zagibalov, C. John, Automatic seed word selection for unsupervised sentiment classification of Chinese text, in: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Manchester, UK, 2008, pp. 1073–1080. <http://www.aclweb.org/anthology/C/C08/C08-1135.pdf>.
[96] Y. He, Incorporating sentiment prior knowledge for weakly supervised sentiment analysis, ACM Trans. Asian Lang. Inf. Process. (TALIP) 11 (2). doi:http://dx.doi.org/10.1145/2184436.2184437.
[97] E. Amigó, J. Carrillo de Albornoz, I. Chugur, A. Corujo, J. Gonzalo, T. Martín, E. Meij, M. de Rijke, D. Spina, Overview of RepLab 2013: evaluating online reputation monitoring systems, in: Proceedings of the 4th International Conference of the CLEF Initiative (RepLab 2013), Valencia, Spain, 2013. <http://ceur-ws.org/Vol-1179/CLEF2013wn-RepLab-AmigoEt2013.pdf>.
[98] E. Amigó, J. Carrillo-de Albornoz, I. Chugur, A. Corujo, J. Gonzalo, E. Meij, M. de Rijke, D. Spina, Overview of RepLab 2014: author profiling and reputation dimensions for online reputation management, in: Proceedings of the 5th International Conference of the CLEF Initiative (RepLab 2014), Sheffield, UK, 2014, pp. 1438–1457. <http://ceur-ws.org/Vol-1180/CLEF2014wn-Rep-AmigoEt2014.pdf>.
[99] D. Vilares, M. Hermo, M.A. Alonso, C. Gómez-Rodríguez, J. Vilares, LyS at CLEF RepLab 2014: creating the state of the art in author influence ranking and reputation classification on Twitter, in: Proceedings of the 5th International Conference of the CLEF Initiative (RepLab 2014), Sheffield, UK, 2014, pp. 1468–1478. <http://ceur-ws.org/Vol-1180/CLEF2014wn-Rep-VilaresEt2014.pdf>.
[100] J.-V. Cossu, K. Janod, E. Ferreira, J. Gaillard, M. El-Bèze, LIA@Replab 2014: 10 methods for 3 tasks, in: Proceedings of the 5th International Conference of the CLEF Initiative (RepLab 2014), Sheffield, UK, 2014, pp. 1458–1467. <http://ceur-ws.org/Vol-1180/CLEF2014wn-Rep-CossuEt2014.pdf>.
[101] J. Villena Román, S. Lana Serrano, E. Martínez Cámara, J.C. González Cristóbal, TASS-Workshop on sentiment analysis at SEPLN, Procesamiento del Lenguaje Natural 50 (2013) 37–44. <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/4657/2759>.
[102] J. Villena Román, J. García Morera, S. Lana Serrano, J.C. González Cristóbal, TASS 2013 – a second step in reputation analysis in Spanish, Procesamiento del Lenguaje Natural 52 (2014) 37–44. <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/4901/2915>.
[103] J. Villena Román, E. Martínez Cámara, J. García Morera, S.M. Jiménez Zafra, TASS 2014 – the challenge of aspect-based sentiment analysis, Procesamiento del Lenguaje Natural 54 (2015) 61–68. <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/5095/2976>.
[104] P. Gamallo, M. García, S. Fernández-Lanza, TASS: a naive-bayes strategy for sentiment analysis on Spanish tweets, in: Proceedings of the Workshop on Sentiment Analysis at SEPLN (TASS 2013), 2013, pp. 126–132. <http://www.daedalus.es/TASS2013/papers/tass2013-submission1-CITIUS-CILENIS.pdf>.
[105] X. Saralegi Urizar, I. San Vicente Roncal, Elhuyar at TASS 2013, in: Proceedings of the Workshop on Sentiment Analysis at SEPLN (TASS 2013), 2013, pp. 143–150. <http://www.daedalus.es/TASS2013/papers/tass2013-submission3-Elhuyar.pdf>.
[106] D. Vilares, M.A. Alonso, C. Gómez-Rodrıguez, LyS at TASS 2013: analysing Spanish tweets by means of dependency parsing, semantic-oriented lexicons and psychometric word-properties, in: Proceedings of the Workshop on Sentiment Analysis at SEPLN (TASS 2013), 2013, pp. 179–186. <http://www.daedalus.es/TASS2013/papers/tass2013-submission8-LYS.pdf>.
[107] D. Vilares, Y. Doval, M.A. Alonso, C. Gómez-Rodríguez, LyS at TASS 2014: a prototype for extracting and analysing aspects from Spanish tweets, in: Proceedings of the Workshop on Sentiment Analysis at SEPLN (TASS 2014), 2013. <http://www.daedalus.es/TASS2014/papers/2.LyS.pdf>.
[108] D.L. Hall, M. McNeese, J. Llinas, T. Mullen, A framework for dynamic hard/soft fusion, in: Proceedings of the 11th International Conference on Information Fusion (FUSION 2008), IEEE, Cologne, Gemrany, 2008, pp. 1–8. <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4632196>.
[109] D.L. Hall, M.D. McNeese, D.B. Hellar, B.J. Panulla, W. Shumaker, A cyber infrastructure for evaluating the performance of human centered fusion, in: Proceedings of the 12th International Conference on Information Fusion, IEEE, Seattle, WA, USA, 2009, pp. 1257–1264. <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5203798>.
[110] M.M. Kokar, J.A. Tomasik, J. Weyman, Formalizing classes of information fusion systems, Inf. Fusion 5 (3) (2004) 189–202, http://dx.doi.org/10.1016/j.inffus.2003.11.001.
[111] S. Wu, F. Crestani, A geometric framework for data fusion in information retrieval, Inf. Syst. 50 (2015) 20–35, http://dx.doi.org/10.1016/j.is.2015.01.001.

[112] G. Shroff, P. Agarwal, L. Dey, Enterprise information fusion for real-time business intelligence, in: Proceedings of the 14th International Conference on Information Fusion (FUSION 2011), IEEE, Chicago, IL, USA, 2011, pp. 1–8. <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5977516>.
[113] C. Sutton, C.T. Morrison, P.R. Cohen, J. Moody, J. Adibi, A Bayesian blackboard for information fusion, Proceedings of the 7th International Conference on Information Fusion (FUSION 2004), vol. 2, International Society of Information Fusion, Stockholm, Sweden, 2004, pp. 1111–1116. <http://www.fusion2004.foi.se/papers/IF04-1111.pdf>.
[114] R. Dueñas-Fernández, J.D. Velásquez, G. L'Huillier, Detecting trends on the web: a multidisciplinary approach, Inf. Fusion 20 (2014) 129–135, http://dx.doi.org/10.1016/j.inffus.2014.01.006.
[115] Z. Hai, K. Chang, J.-J. Kim, C.C. Yang, Identifying features in opinion mining via intrinsic and extrinsic domain relevance, IEEE Trans. Knowl. Data Eng. 26 (3) (2014) 623–634, http://dx.doi.org/10.1109/TKDE.2013.26.
[116] X. Xueke, C. Xueqi, T. Songbo, L. Yue, S. Huawei, Aspect-level opinion mining of online customer reviews, China Commun. 10 (3) (2013) 25–41, http://dx.doi.org/10.1109/CC.2013.6488828.
[117] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022. <http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>.
[118] A.D.S. Lima, J.S. Sichman, SORM: a social opinion relevance model, in: Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technologies (WI-IAT 2014), vol. 1, IEEE, Warsaw, Poland, 2014, pp. 78–85, http://dx.doi.org/10.1109/WI-IAT.2014.19.
[119] S.-R. Yan, X.-L. Zheng, Y. Wang, W.W. Song, W.-Y. Zhang, A graph-based comprehensive reputation model: exploiting the social context of opinions to enhance trust in social commerce, Inf. Sci. (2014), http://dx.doi.org/10.1016/j.ins.2014.09.036.
[120] B. Schuller, T. Knaup, Learning and knowledge-based sentiment analysis in movie review key excerpts, in: A. Esposito, A.M. Esposito, R. Martone, V. Mller, G. Scarpetta (Eds.), Toward Autonomous, Adaptive, and Context-Aware Multimodal Interfaces. Theoretical and Practical Issues, Lect. Notes Comput. Sci., vol. 6456, Springer, 2011, pp. 448–472, http://dx.doi.org/10.1007/978-3-642-18184-9_39.
[121] P.J. Stone, D.C. Dunphy, M.S. Smith, The General Inquirer: A Computer Approach to Content Analysis, MIT press, Oxford, England, 1966.
[122] C. Havasi, R. Speer, J. Alonso, ConceptNet 3: a flexible, multilingual semantic network for common sense knowledge, in: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2007), Borovets, Bulgaria, 2007. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.115.2844>.
[123] R.M. Karampatsis, J. Pavlopoulos, P. Malakasiotis, AUEB: two stage sentiment analysis of social network messages, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Association for Computational Linguistics and Dublin City University, Dublin, Ireland, 2014, pp. 114–118. <http://www.aclweb.org/anthology/S14-2015>.
[124] K. Liu, L. Xu, J. Zhao, Co-extracting opinion targets and opinion words from online reviews based on the word alignment model, IEEE Trans. Knowl. Data Eng. 27 (3) (2015) 636–650, http://dx.doi.org/10.1109/TKDE.2014.2339850.
[125] J. Duan, J. Zeng, Mining opinion and sentiment for stock return prediction based on web-forum messages, in: Proceedings of the 10th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2013), IEEE, Shenyang, China, 2013, pp. 984–988, http://dx.doi.org/10.1109/FSKD.2013.6816338.
[126] J. Lafferty, A. McCallum, F.C. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: Proceedings of the 18th International Conference on Machine Learning (ICML 2001), Morgan Kaufmann Publishers Inc., Williamstown, MA, USA, 2001, pp. 282–289. <http://dl.acm.org/citation.cfm?id=645530.655813>.
[127] J. Wagner, P. Arora, S. Cortes, U. Barman, D. Bogdanova, J. Foster, L. Tounsi, DCU: aspect-based polarity classification for semeval task 4, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Association for Computational Linguistics and Dublin City University, Dublin, Ireland, 2014, pp. 223–229. <http://www.aclweb.org/anthology/S14-2036>.
[128] S. Baccianella, A. Esuli, F. Sebastiani, SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining, Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010), vol. 10, European Language Resources Association, Valletta, Malta, 2010, pp. 2200–2204. <http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf>.
[129] S. Rosenthal, A. Agarwal, K. McKeown, Columbia NLP: sentiment detection of sentences and subjective phrases in social media, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Association for Computational Linguistics and Dublin City University, Dublin, Ireland, 2014, pp. 198–202. <http://www.aclweb.org/anthology/S14-2031>.
[130] C.M. Whissell, The dictionary of affect in language, Emotion Theory Res. Exper. 4 (113–131) (1989) 94.
[131] C. Fellbaum (Ed.), WordNet: An Electronic Lexical Database, MIT Press, 1998.
[132] S. Rosenthal, K. McKeown, Columbia NLP: sentiment detection of subjective phrases in social media, in: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Atlanta, GA, USA, 2013, pp. 478–482. <http://www.aclweb.org/anthology/S13-2079>.
[133] A. Agarwal, F. Biadsy, K.R. Mckeown, Contextual phrase-level polarity analysis using lexical affect scoring and syntactic n-grams, in: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009), Association for Computational Linguistics, Athens, Greece, 2009, pp. 24–32. <http://dl.acm.org/citation.cfm?id=1609067.1609069>.
[134] A. Mudinas, D. Zhang, M. Levene, Combining lexicon and learning based approaches for concept-level sentiment analysis, in: Proceedings of the 1st International Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM 2012), vol. 5, ACM, Beijing, China, 2012, pp. 5:1–5:8, http://dx.doi.org/10.1145/2346676.2346681.
[135] M. Wu, L. Wang, L. Yi, A novel approach based on review mining for product usability analysis, in: Proceedings of the 4th International Conference on Software Engineering and Service Science (ICSESS 2013), IEEE, Beijing, China, 2013, pp. 942–945, http://dx.doi.org/10.1109/ICSESS.2013.6615461.
[136] F.E. White, Data fusion lexicon, Tech. rep., Joint Directors of Laboratories, Data Fusion Sub-Panel, Naval Ocean Systems Center, 1991. <http://www.dtic.mil/dtic/tr/fulltext/u2/a529661.pdf>.
[137] T. Bass, Intrusion detection systems and multisensor data fusion, Commun. ACM 43 (4) (2000) 99–105, http://dx.doi.org/10.1145/332051.332079.
[138] J. Llinas, C. Bowman, G. Rogova, A. Steinberg, E. Waltz, F. White, Revisiting the JDL data fusion model II, in: Proceedings of the 7th International Conference on Information Fusion (FUSION 2004), Stockholm, Sweden, 2004, pp. 1218–1230. <http://www.fusion2004.foi.se/papers/IF04-1218.pdf>.
[139] B.V. Dasarathy, Sensor fusion potential exploitation–Innovative architectures and illustrative applications, Proc. IEEE 85 (1) (1997) 24–38, http://dx.doi.org/10.1109/5.554206.
[140] C.D. Manning, P. Raghavan, H. Schütze, Introduction to information retrieval, vol. 1, Cambridge University Press, 2008. Ch. 6.
[141] I. Khozyainov, E. Pyshkin, V. Klyuev, Spelling out opinions: difficult cases of sentiment analysis, in: Proceedings of the International Joint Conference on Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA 2013), IEEE, Aizu-Wakamatsu, Japan, 2013, pp. 231–237, http://dx.doi.org/10.1109/ICAwST.2013.6765439.
[142] D. Maynard, K. Bontcheva, D. Rout, Challenges, in: Proceedings of the LREC Workshop @NLP can u tag #usergeneratedcontent?! (LREC 2012), Istambul, Turkey, 2012, pp. 15–22. <http://lrec-conf.org/proceedings/lrec2012/workshops/21.LREC2012%20NLP4UGC%20Proceedings.pdf#20>.
[143] H.D. Kim, K. Ganesan, P. Sondhi, C. Zhai, Comprehensive review of opinion summarization, Tech. rep., University of Illinois at Urbana-Champaign, 2011. <http://hdl.handle.net/2142/18702>.
[144] E. Martínez-Cámara, M.T. Martín-Valdivia, L.A. Urena-López, A. Montejo-Ráez, Sentiment analysis in Twitter, Nat. Lang. Eng. 20 (01) (2014) 1–28, http://dx.doi.org/10.1017/S1351324912000332.
[145] E. Marrese-Taylor, C. Rodríguez, J.D. Velásquez, G. Ghosh, S. Banerjee, Web opinion mining and sentimental analysis, in: Advanced Techniques in Web Intelligence-2, Studies in Computational Intelligence, Springer, Berlin, Heidelberg, 2013, pp. 105–126, http://dx.doi.org/10.1007/978-3-642-33326-2_5.
[146] N. Medagoda, S. Shanmuganathan, J. Whalley, A comparative analysis of opinion mining and sentiment classification in non-English languages, in: Proceedings of the International Conference on Advances in ICT for Emerging Regions (ICTer 2013), IEEE, Colombo, Sri Lanka, 2013, pp. 144–148, http://dx.doi.org/10.1109/ICTer.2013.6761169.
[147] V. Singh, S.K. Dubey, Opinion mining and analysis: a literature review, in: Proceedings of the 5th International Conference – Confluence The Next Generation Information Technology Summit (CONFLUENCE 2014), IEEE, Noida, India, 2014, pp. 232–239, http://dx.doi.org/10.1109/CONFLUENCE.2014.6949318.
[148] I. Bloch, Information combination operators for data fusion: a comparative review with classification, IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum. 26 (1) (1996) 52–67, http://dx.doi.org/10.1109/3468.477860.
[149] D.L. Hall, J. Llinas, An introduction to multisensor data fusion, Proc. IEEE 85 (1) (1997) 6–23, http://dx.doi.org/10.1109/5.554205.
[150] D. Smith, S. Singh, Approaches to multisensor data fusion in target tracking: a survey, IEEE Trans. Knowl. Data Eng. 18 (12) (2006) 1696–1710, http://dx.doi.org/10.1109/TKDE.2006.183.
[151] H. Wache, T. Voegele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, S. Hübner, Ontology-based integration of information: a survey of existing approaches, in: Proceedings of the IJCAI Workshop on Ontologies and Information Sharing (IJCAI 2001), International Joint Conferences on Artificial Intelligence, Seattle, WA, USA, 2001, pp. 108–117. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.8073>.
[152] G.L. Rogova, V. Nimier, Reliability in information fusion: literature survey, in: Proceedings of the 7th International Conference on Information Fusion (FUSION 2004), vol. 2, International Society of Information Fusion, Stockholm, Sweden, 2004, pp. 1158–1165. <http://www.fusion2004.foi.se/papers/IF04-1158.pdf>.
[153] J. Yao, V.V. Raghavan, Z. Wu, Web information fusion: a review of the state of the art, Inf. Fusion 9 (4) (2008) 446–449, http://dx.doi.org/10.1016/j.inffus.2008.05.002.
[154] A.A. Goshtasby, S. Nikolov, Image fusion: advances in the state of the art, Inf. Fusion 8 (2) (2007) 114–118, http://dx.doi.org/10.1016/j.inffus.2006.04.001.
[155] I. Corona, G. Giacinto, C. Mazzariello, F. Roli, C. Sansone, Information fusion for computer security: State of the art and open issues, Inf. Fusion 10 (4) (2009) 274–284, http://dx.doi.org/10.1016/j.inffus.2009.03.001.

[156] N.E. El Faouzi, H. Leung, A. Kurian, Data fusion in intelligent transportation systems: progress and challenges – a survey, Inf. Fusion 12 (1) (2011) 4–10, http://dx.doi.org/10.1016/j.inffus.2010.06.001.
[157] G. Navarro-Arribas, V. Torra, Information fusion in data privacy: a survey, Inf. Fusion 13 (4) (2012) 235–244, http://dx.doi.org/10.1016/j.inffus.2012.01.001.
[158] J.D. Velásquez, Web mining and privacy concerns: Some important legal issues to be consider before applying any data and information extraction technique in web-based environments, Expert Syst. Appl. 40 (13) (2013) 5228–5239.
[159] S. Sun, H. Shi, Y. Wu, A survey of multi-source domain adaptation, Inf. Fusion 24 (2014) 84–92, http://dx.doi.org/10.1016/j.inffus.2014.12.003.

B AnCora Tags for Syntactic Dependency

Tag	Gloss	Tag	Gloss
ADJUNCT	Adjoined element	CPREP	Complement of a preposition
AO	Sentence adjunct	CREG	Prepositional complement
APOS	Apposition	DETER	Head determiner
ATR	Attribute	ESPEC**	Non-head determiner
AUX	Auxiliary Verb	ET	Textual element
CADJ	Complement of an adjective	IMPERS	Impersonal mark
CADV	Complement of an adverb	INSERT	Inserted element
CAG	Agent Complement	INTJ	Interjection
CC	Adjunct	MOD	Verb modifier
CD	Direct object	MORF	Verbal morpheme
CD.Q	Quantitative direct object	NEG	Negative element
CI	Indirect object	PASS	Passive mark
CN	Complement of a noun	PUNC	Punctuation mark
CNEG	Complement of a negation	ROOT	Sentence head
CO	Coordinating element	SUJ	Subject
CONJUNCT*	Coordinated element	SUBORD	Subordinating element
CPRED	Predicative complement	VOC	Vocative
CPRED.CD	CD Predicative complement		

AnCora Tags for Syntactic Dependency.
Source: [107].

* CONJUNCT appears as CONJ in current version of the corpus.
** ESPEC Tag appears as SPEC.

C AnCora Constituents

Constituent	Gloss
sa	Adjective phrase
sadv	Adverbial phrase
S.F.A.	Finite adjetive phrase
S.F.C.	Finite complement clause
S.F.C.co	Coordinated finite completive clause
S.F.P.	Finite prepositional phrase
sn	Noun phrase
sn.e	Elliptical noun phrase
S.NF.A.	Non-finite adjective phrase
S.NF.C.	Non-finite complement clause
S.NF.P.	Non-finite prepositional phrase
sp	Prepositional phrase
relatiu	Relative clause

AnCora Constituents.
Source: [107].

D AnCora Relationship Between Dependencies and Constituents

Function Tag	Constituent Tags
SUJ	sn, sn.e, relatiu, S.F.C., S.NF.C.
CD	relatiu, S.F.C., S.F.C.co, sn, sp
CI	sn, sp
ATR	sa, sn, S.F.C., S.NF.C., S.NF.P., sp
CPRED	sa, sn, S.NF.P.,
CREG	relatiu, sadv, S.F.C., sn, sp
CAG	sp
CC, CCT, CCL	sadv, S.F.A., S.NF.A., S.F.C., sn, sp
ET	sadv, sp
MOD	sadv, sp
NEG	neg (negation)
PASS	morfema.verbal (passive verb morpheme)
IMPERS	morfema.verbal (impersonal verb morpheme)
VOC	sn

AnCora Relationship Between Dependencies and Constituents.
Source: [107].

E CoNLL File Example

"Esto permitirá al banco sanear su portafolio, que es condición básica para continuar en su privatización." (This will allow the bank to restructure its portfolio, a necessary condition for continuing with its privatization) represented in CoNLL format with Ancora Tags.

```
1 Esto este p p gen=c|num=s|postype=demonstrative 2 suj _ _
2 permitirá permitir v v num=s|postype=main|person=3|mood=indicative|tense=future 0 sentence _ _
3 al al s s gen=m|num=s|postype=preposition|contracted=yes 2 ci _ _
4 banco banco n n gen=m|num=s|postype=common 3 sn _ _
5 sanear sanear v v postype=main|mood=infinitive 2 cd _ _
6 su su d d gen=c|num=s|postype=possessive|person=3 7 spec _ _
7 portafolio portafolio n n gen=m|num=s|postype=common 5 cd _ _
8 ,,f f punct=comma 10 f _ _
9 que que p p gen=c|num=c|postype=relative 10 suj _ _
10 es ser v v num=s|postype=semiauxiliary|person=3|mood=indicative|tense=present 5 S _ _
11 condición condición n n gen=f|num=s|postype=common 10 atr _ _
12 básica básico a a gen=f|num=s|postype=qualificative 11 s.a _ _
13 para para s s postype=preposition 11 sp _ _
14 continuar continuar v v postype=main|mood=infinitive 13 S _ _
15 en en s s postype=preposition 14 cc _ _
16 su su d d gen=c|num=s|postype=possessive|person=3 17 spec _ _
17 privatización privatización n n gen=f|num=s|postype=common 15 sn _ _
18 . . f f punct=period 2 f _ _
```

Source: Ancora Corpus [107].

F CoNLL Columns

Column Number	Field Name	Description
1	ID	Token counter, starting at 1 for each new sentence.
2	FORM	Word form or punctuation symbol.
3	LEMMA	Lemma or stem (depending on particular data set) of word form, or an underscore if not available.
4	CPOSTAG	Coarse-grained part-of-speech tag, where tagset depends on the language.
5	POSTAG	Fine-grained part-of-speech tag, where the tagset depends on the language, or identical to the coarse-grained part-of-speech tag if not available.
6	FEATS	Unordered set of syntactic and/or morphological features(depending on the particular language), separated by a vertical bar (|), or an underscore if not available.
7	HEAD	Head of the current token, which is either a value of ID or zero ('0'). Note that depending on the original treebank annotation, there may be multiple tokens with an ID of zero.
8	DEPREL	Dependency relation to the HEAD. The set of dependency relations depends on the particular language. Note that depending on the original treebank annotation, the dependency relation may be meaningful or simply 'ROOT'.
9	PHEAD	Projective head of current token, which is either a value of ID or zero ('0'), or an underscore if not available. Note that depending on the original treebank annotation, there may be multiple tokens with an ID of zero. The dependency structure resulting from the PHEAD column is guaranteed to be projective (but is not available for all languages), whereas the structures resulting from the HEAD column will be non-projective for some sentences of some languages (but is always available).
10	PDEPREL	Dependency relation to the PHEAD, or an underscore if not available. The set of dependency relations depends on the particular language. Note that depending on the original treebank annotation, the dependency relation may be meaningful or simply 'ROOT'.

CoNLL Columns' description
Source: `http://ilk.uvt.nl/conll/` Accessed August 03, 2015

G Some Abbreviations Obtained from 50000 tweets

Abbreviation	Meaning	Abbreviation	Meaning
aki	*aquí*	fb	Facebook
aprox	*aproximadamente*	FB	Facebook
aunq	*aunque*	finde	*fin de semana*
aver	*a ver*	gim	*gimnasio*
bueh	*bueno*	grax	*gracias*
cel	*celular*	grx	*gracias*
celu	*celular*	info	*información*
cm	*como*	k	*que*
cn	*con*	pls	*por favor*
d	*de*	porfa	*por favor*

20 Abbreviations obtained from 50000 tweets.

H Example of a Sentence Represented as a NLTK Dependency Graph

Below, the sentence "Devuelvan el dinero por plazo de entrega sin cumplir" (I want my money back for the unfulfilled delivery deadline) is presented as a NLTK dependency graph (both the "tag" and the "feats" fields have been omitted for presentation purposes).

```
[
{u'address': 0, u'ctag': u'TOP', u'deps': [1], u'lemma': None, u'rel': u'TOP', u'word': None},
{u'address': 1, u'ctag': u'v', u'deps': [3, 8], u'head': 0, u'lemma': u'devolver', u'rel': u'sentence', u'word': u'devuelvan'},
{u'address': 2, u'ctag': u'd', u'deps': [], u'head': 3, u'lemma': u'el', u'rel': u'spec', u'word': u'el'},
{u'address': 3, u'ctag': u'n', u'deps': [2, 4], u'head': 1, u'lemma': u'dinero', u'rel': u'cd', u'word': u'dinero'},
{u'address': 4, u'ctag': u's', u'deps': [5], u'head': 3, u'lemma': u'por', u'rel': u'sp',u'word': u'por'},
{u'address': 5, u'ctag': u'n', u'deps': [6], u'head': 4, u'lemma': u'plazo', u'rel': u'sn', u'word': u'plazo'},
{u'address': 6, u'ctag': u's', u'deps': [7], u'head': 5, u'lemma': u'de', u'rel': u'sp', u'word': u'de'},
{u'address': 7, u'ctag': u'n', u'deps': [], u'head': 6, u'lemma': u'entrega', u'rel': u'sn', u'word': u'entrega'},
{u'address': 8, u'ctag': u's', u'deps': [9], u'head': 1, u'lemma': u'sin', u'rel': u'cc', u'word': u'sin'},
{u'address': 9, u'ctag': u'v', u'deps': [], u'head': 8, u'lemma': u'cumplir', u'rel': u'S', u'word': u'cumplir'}
]
```

I Full Algorithm for Applying the Heuristics

```
Data: dependency_graph: The sentence represented as a dependency graph
Result: dependency_graph: The dependency graph with the propagated polarity
levels ← obtain levels from dependency_graph;
foreach level in levels do
    foreach node in level do
        /* Intensification Rule                                              */
        if node_tag is adverb and node_rel is {spec or espec or cc or sadv} then
            head_intensified += intensification_strength(node_word)
        /* Negation Rules                                                    */
        else if node_word is {no or nunca or sin} then
            /* Subjective Parent Rule                                        */
            if head_sent_orig > 0 then
                head_sent −= negation_strength
            else if head_sent_orig < 0 then
                head_sent += negation_strength
            /* Subject Complement -- Direct Object Rule                      */
            else if head_sent_orig == 0 then
                visited_siblings = [ ];
                foreach sibling in siblings do
                    if sibling_rel is {atr or cd} then
                        if sibling_sent > 0 then
                            sibling_sent−= negation_strength
                        else if sibling_sent < 0 then
                            sibling_sent+= negation_strength
                        end
                        append sibling to visited_siblings;
                    /* Adjunct Rule                                          */
                    else if sibling_rel is cc then
                        if cc not in visited_siblings then
                            if sibling_sent > 0 then
                                sibling_sent−= negation_strength
                            else if sibling_sent < 0 then
                                sibling_sent+= negation_strength
                            end
                        end
                        append sibling to visited_siblings;
                    end
                end
                /* Default Rule                                              */
                if visited_siblings == [ ] then
                    foreach sibling in siblings do
                        if sibling_sent > 0 then
                            sibling_sent−= negation_strength
                        else if sibling_sent < 0 then
                            sibling_sent+= negation_strength
                        end
                    end
                end
            end
```

```
    else if node_rel is art_rel_adversative then
        get adversation_type from node_tag;
        get conjunction_address from node_tag;
        define weight_main_clause depending on adversation_type;
        define weight_adversative_clause depending on adversation_type;
        main_clause_polarity ← 0;
        adversative_clause_polarity ← 0;
        foreach child in node_deps do
            if child_address < conjunction_address then
                main_clause_polarity += child_sent;
            else if child_address > conjunction_address then
                adversative_clause_polarity += child_sent;
            end
        end
        node_sent ← (weight_main_clause * main_clause_polarity) +
        (weight_adversative_clause * adversative_clause_polarity);
    end
    if node_intensified > 0 and node_sent_orig == 0 then
        node_sent* = (1 + node_intensified);
    else if node_intensified > 0 and node_sent_orig != 0 then
        node_sent+ = (node_sent_orig * node_intensified);
    end
    head_sent ← node_sent
  end
end
return dependency graph
```

Algorithm 6.0: Heuristic Application Algorithm

J Interpretation of Kappa

Kappa	Agreement
$\kappa < 0$	Less than chance agreement
$\kappa = 0$	Pure chance agreement
$0.01 \leq \kappa \leq 0.20$	Slight agreement
$0.21 \leq \kappa \leq 0.40$	Fair agreement
$0.41 \leq \kappa \leq 0.60$	Moderate agreement
$0.61 \leq \kappa \leq 0.80$	Substantial agreement
$0.81 \leq \kappa \leq 0.99$	Almost perfect agreement
$\kappa = 1$	Perfect agreement

Interpretation of Kappa
Source: [143].